Can Open Source projects succeed when the producers are not the users? Lessons from the data processing field.

Nicolas Jullien, Karine Roudaut

Can Open Source projects succeed when the producers are not users? Lessons from the data processing field[1]
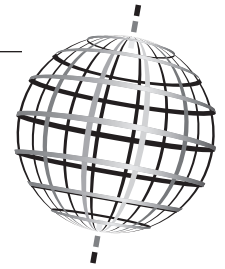
NICOLAS JULLIEN

Télécom Bretagne, Institut Mines-Telecom, EA ICI

KARINE ROUDAUT

Télécom Bretagne

RÉSUMÉ

Le logiciel libre, ou open source propose une façon originale de résoudre le dilemme de l'incitation à la production de biens informationnels, basé sur le principe proposé par von Hippel (1988) de l'utilisateur comme innovateur : parce que les utilisateurs bénéficient de l'innovation, ils ont intérêt à la produire, et comme ils peuvent s'attendre à de l'innovation cumulative sur leur propre proposition, ils ont intérêt à la partager. Mais que se passe-t-il dans ces communautés si les producteurs ne sont pas les utilisateurs ? Nous discutons cette question grâce à une étude qualitative de projets de open-source dans le secteur des « algorithmes ». Nous constatons que dans ce cas les producteurs ne participent guère à ces projets.

Mots clés : Économie de la connaissance, sociologie, open source, science, standardisation.

ABSTRACT

Free/Libre Open Source Software (FLOSS) proposes an original way to solve the incentive dilemma for the production of information goods, based on von Hippel (1988)'s user-as-innovator principle: as users benefit from innovation, they have incentive to produce it, and as they can expect cumulative innovation on their own proposition, they have incentive to share it. But what is the incentive for producers when they are not users? We discuss this question via a qualitative study of FLOSS projects in "algorithm-based industries". We find that in this case producers hardly participate in such projects.

Keywords: Knowledge economics, sociology, open source, science, standardization.

RESUMEN

El software libre (en inglés FLOSS, Free, Libre, Open Source Software), propone una forma única para resolver el dilema de los incentivos para producir bienes de información, basándose en el principio del usuario como innovador de Von Hippel (1998): conforme los usuarios se benefician de la innovación, tienen incentivos para producirla, y puesto que pueden esperar una acumulación de innovación sobre su propia aportación, también tienen incentivo para compartirla. Pero, ¿cuál es el incentivo para los productores cuando no son ellos los usuarios? Discutimos esta pregunta a través de un estudio cualitativo de proyectos FLOSS en "industrias basadas en algoritmos". Descubrimos que, en este* caso, los productores rara vez participan en dichos proyectos. Economía del conocimiento, sociología, software libre, código abierto, ciencia, estandardización.

Palabras claves: Economía del conocimiento, sociología, software libre, código abierto, ciencia, estandardización.

Free/ Libre Open Source Software (FLOSS)[2] is said to propose an original solution to the segmentation of knowledge production and is apparently more efficient than traditional Intellectual Property (IP) systems in cases where knowledge is modular and cumulative, as it facilitates and decreases the cost of producing new knowledge (Bessen, 2005). This is because "entry competition and innovation may be easier if a competitor needs only to produce a single better component, which can then hook up the market range of complementary components, than if each innovator must develop an entire system" (Farrell, 1989).

Many scholars believe this model can and should be applied to other industries. Joly and Hervieu (2003) plead for a high degree of knowledge resource mutualization in the field of genomics in Europe, not for renouncement of intellectual property, but by organizing a collective system of management of intellectual property. Hope (2008) defends the social advantages of an "open source" biotech industry. Dang Nguyen and Genthon (2006) call for "concentrating on an ambitious program of FLOSS production in the embedded systems and domestic networks at the European level". In both cases, the authors argue that this would reinforce the competitive position of European firms facing US multinationals and, by pooling basic technologies, avoid innovation clamping. Actually, Open Source initiatives are numerous, in various industries (Balka et al., 2009), but FLOSS remains the movement which has impacted its industry the most and the exemplar for the open source Intellectual Property model for innovation production.

FLOSS appears to be a specific organization of production (O'Mahony, 2003; O'Mahony, 2007). It organizes the collaboration between actors of divergent interests (O'Mahony

---

2. In this article, we will use FLOSS or Open Source Software indifferently. We define a "free, libre, open source software" as software distributed (made available), for free or not, with its source code and the right to modify the program and to redistribute these modifications. See Clément-Fontaine (2002, 2009) for a juridical analysis.

and Bechky, 2008), creating both "coat-tailing systems" to integrate heterogeneity in terms of contributions and goals (Hemetsberger and Reinhardt, 2009), and a model for virtual teams close to Cohen et al.'s (1972) garbage can model (Lacolley et al., 2007; Li et al., 2008).

But this does not mean that producing FLOSS is costless, and the fact remains that producers must have incentives to participate. In classic knowledge or software production systems, there are either financial flows from the user to the producers, or collective, public support of the producers (as in open science) to do so[3]. In this regard as well, FLOSS is innovative. While the motivations for participating vary[4], the core of the incentive framework is the "private collective" innovation model (von Hippel and von Krogh, 2003), or the "user-as-innovator principle" (Lakhani and von Hippel, 2003; von Hippel and von Krogh, 2003): as users directly benefit from the piece of innovation they produce, they have incentive to produce it, and as they can expect add-on, feedback or cumulative innovation on their own proposition, they have incentive to freely share it.

Does that mean that the only industries where FLOSS-like organization may flourish are those where producers and users are the same? This is the hypothesis we explore in this paper. We discuss this point by studying domains where FLOSS organization should be, *a priori*, of maximum efficiency, because the production of knowledge is modular and incremental, but are instead systems where the users and the producers are disjointed. This requires us to first analyze the problems raised by this division, before identifying an industry where these problems may be overcome, for the conditions most favorable to finding successful projects. In this case, we are dealing with algorithm-based industries. The lack of research on the subject suggests a qualitative approach to develop initial, but detailed, exploratory results (Von Krogh et al., 2003). We develop this approach via interviews with researchers in those industries. Our study question, to quote Yin (2009), will be to understand if software producers may find interest in participating in open source projects when they are not users, and why.

The article is organized as follows. In section 2 we review the literature to define the characteristics of a FLOSS organization and the consequences of a split of the user-producer actors on it. In section 3 we present our fields of study, "algorithm-based industries", and specifically open source projects in these industries, as we want to explore the reason producers have (or do not have) to participate in such projects. This discussion is followed by a description of our methodology. In section 4, we present our findings on the knowledge creation process, and on the use of FLOSS, before discussing them with regard to the incentives for proposing FLOSS solutions.

Research questions: incentive and investment for software producers in a FLOSS project

FLOSS is a kind of knowledge good (Foray, 2004), produced by a group of people making it freely available to users. Crowston et al. (2006), followed by Lee et al. (2009), looked at FLOSS organization as information system organization. They show that, as a software production organization, FLOSS can be studied using classic information system models (DeLone and McLean, 1992, 2002, 2003), stressing the importance of the production process to explain the quality of the production and thus of the success of the product. It is, more generally, an example of a "knowledge commons" (Hess and Ostrom, 2006a), i.e. an organization dedicated to the production of, to quote Lundvall and Johnson (1994), new algorithms to meet "practical" needs that deal with "know-why" production, "scientific knowledge of the principles", and "laws of nature".

Considering Hess and Ostrom's (2006a) framework, shown in figure 1, splitting the user-as-innovator means to restudy the action arena, or how people interact to define and produce what is needed (what we call the "FLOSS factory"), and the cost-benefit for the providers to participate in this project, to go to the action arena.

The fact that some use what others produce is not uncommon in knowledge commons production. Crowston and Fagnot (2008), quoting Mockus et al.'s (2000) study on the open source project Apache, and Zachte's (2007) work on Wikipedia, noted that the distribution of contributions is biased, with a few people doing most of the work and most doing little or nothing, as is the case in most volunteer organizations (Marwell and Oliver, 1993; Crowston, 2011). What is specific here is those who do are not users, and thus may not perceive the platform of the FLOSS project as directly useful for them. And this "perceived usefulness" (Hackman, 1987; Seddon, 1997) has proved to be an important factor in explaining participation (Crowston and Fagnot, 2008). We will thus study, for the producers, the interest of participation, looking at their perceived costs and benefits, considering the fact that we tried to focus on projects where these perceived costs should be minimal and these perceived benefits maximal. We do not mean only financial consequences, but more generally what the perceived involvement and interests are. But as explanatory variables, we will also have to look at users' adoption and feedback, and the projects' environments (who initiated the project, or the policy makers, institutional support, the level of openness, or the rules-in-use) as both matter for producers' valuations, as explained by Crowston and Fagnot (2008), we will have to look at them also, as an explanatory background. The "biophysical characteristics"[5]

---

3. Foray and Cassier (2001b,a) propose a synthesis of the discussion of the ins and outs of the economics of knowledge creation and insist on the need for the creation of incentives for the producers of knowledge to produce and to diffuse.

4. The study of FLOSS participants' motivations has already generated a vast literature, reviewed by Lakhani and Wolf (2005) and Shah (2006), Scacchi (2007) among others.

5. Here, we mean the scientific and industrial field the platforms address, their technical characteristics if they matters.

FIGURE 1

Institutional Analysis and Development framework
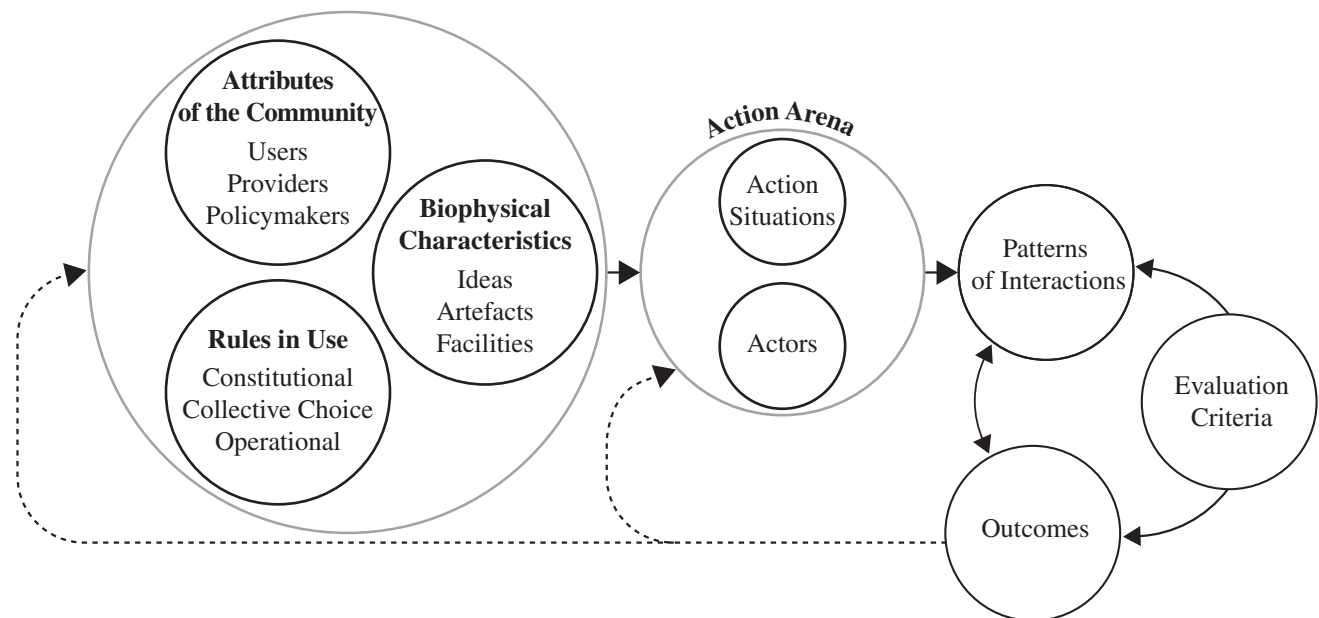
(Hess and Ostrom, 2006a, p. 44)

will be described in the methodology, in the presentation of the field we decided to look at and of the Floss factories existing in this field.

The producers

Not being users, producers may not benefit from innovation, which is the core of von Hippel's classic FLOSS incentive regime, but not the only incentive.

Theoretical analyses of incentives, in software projects (Foray and Zimmermann, 2001; Lerner and Tirole, 2002) or in wikis (Forte and Bruckman (2005), using Latour and Woolgar's (1979) analysis of science "cycles of credit"), estimate that the other main vector for participation is the quest for reputation. Applied works on Wikipedia (Nov, 2007; Yang and Lai, 2010; Zhang and Zhu, 2011), professional electronic networks (Wasko and Faraj, 2005; Jullien et al., 2011), and open source software (Shah, 2006; Scacchi, 2007), confirm that peer recognition, whether it be professional or community recognition, is a main motive for participation, in addition to intrinsic factors (personal enjoyment and satisfaction from helping by sharing their knowledge). This argument that on-line volunteer participation can be explained by the same incentives found in science has also been used in reverse. Schweik (2006) uses the same framework by Hess and Ostrom (2006a) to describe open source organization in order to extract its main mechanisms to construct an open science project.

Considering this, the chance to find a successful FLOSS production project where providers are not users should be enhanced if the producers belong to the scientific community, where reputation is one of the main rewards (Foray and Cassier, 2001b,a). This should be the case, even if the sociology of sciences (Lamy and Shinn, 2006) shows that the commitment to non-academic projects (in that case entrepreneurship) may be weighted by potential concerns about the "scientific value" of such non-traditional academic production, the "attachment" to the cycles of academic valorization.

The first question to answer are thus the following: What is the interest for producers to publish open source material? Does this provides an improvement in their reputation? How do they consider this reputation in comparison to classic academic reward?

The investment to participate in a FLOSS project may seem obvious, as it deals with the production of a piece of software. Once this software is produced, making it open source is just a matter of license and of uploading the product on a platform. Looking at case studies where producers have to produce software anyway may thus lead to a close-to-zero extra investment for FLOSS. However, there are specific costs attached to the participation in a FLOSS community (Von Krogh et al., 2003): one has to understand how the community works, what the global structure of the project is, what new features are expected, what programing language(s) is/are in use, etc. FLOSS organization may decrease the cost

in comparison to a classic closed platform (Kogut and Metiu, 2001), but it does not make them disappear.

Thus our second major question is what specific investments are involved in contributing to an open-source project, and how they are evaluated by the producers themselves. Actually, these investments may vary as the producers are not the users, because they have to understand what the users need, and may have to explain to these users how their production meets these needs. And the benefits may also vary according to the incentives the project initiators have settled on. This leads to two preliminary questions we will discuss in the following two sub-sections.

Methodology

Our case study is based on three scientific fields: bio-informatics, remote sensing and digital communication. In this section, after presenting a short definition of these scientific fields, we explain why they offer fertile grounds for our discussion. We explain the qualitative research method (Miles and Huberman, 1994), based on an interpretative approach. The empirical data produced by this method are the foundation of our argument.

FIELD OF INVESTIGATION: DATA PROCESSING

Information and communication technologies have changed the way some knowledge-based sciences work: "ICT allow the exploration and analysis of the contents of gigantic databases" (Foray, 2004, p. 29). Genetics, but also earth observation, or ICT industries like digital communication, have seen major breakthroughs in the past decade. All these disciplines take a huge amount of data, provided by physical "pipes" (sensors, genes, antennas and coders), apply algorithms[6] to process the data and provide a result (a piece of information), and then compare this result with reality (e.g. original signal, comparable genes, in-the-field measures). This processing activity involves, to a large extent, filtering and cleaning the data of errors of measure and comparing various sources of data, usually with statistical methods. One consequence of the appeal of statistical skills for the traditional sciences is the emergence of new scientific fields for these specific classical domains: biocomputing or bioinformatics to support the work of the biologist, remote sensing to help with earth observation, and digital transmission for communications.

Bioinformatics

"Derives knowledge from computer analysis of biological data. These can consist of the information stored in the genetic code, but also experimental results from various sources, patient statistics, and scientific literature. Research in bioinformatics includes method development for storage, retrieval, and analysis of the data. Bioinformatics is a rapidly developing branch of biology and is highly interdisciplinary, using techniques and concepts from informatics, statistics, mathematics, chemistry, biochemistry, physics, and linguistics. It has many practical applications in different areas of biology and medicine." (Nilges and Linge, 2002)

Remote Sensing

"In the most generally accepted meaning refers to *instrument-based techniques employed in the acquisition and measurement of spatially organized* (most commonly, geographically distributed) *data/information on some property(ies)* (spectral; spatial; physical) of an array of target points (pixels) within the sensed scene that correspond to features, objects, and materials" (Short, 2009). Once the data have been collected, they may be used to recognize patterns. This is done by pattern recognition, i.e. "techniques for classifying a set of objects into a number of distinct classes by considering similarities of objects belonging to the same class and the dissimilarities of objects belonging to different classes [....]" (Short, 2009, part 2.6).

Digital transmission or data communication

Is *"the electronic transmission of information that has been encoded digitally (as for storage and processing by computers)"*[7]. The problem of digital transmission is that errors may occur during the transmission and thus information may be lost. To cope with this problem, some techniques add extra information (error-correcting codes) which, once decoded helps to correct the errors (see Anderson, 2005, for a complete presentation). Of course, the aim of the algorithms is to correct as many errors as possible with a minimum of extra information added and/or in a minimum of data processing time. To do so, statistical techniques for digital data have had to be developed (referred to as digital communication theory, covered by a dedicated IEEE group[8]).

What is common to these fields is that designing a "better" algorithm requires competencies in statistics and classification, a form of applied mathematics. These are not integrated into biology, earth observation or communication, but rather are autonomous disciplines referenced by IEEE. This separation was the insight which made us think about

---

6. An algorithm is a "procedure that produces the answer to a question or the solution to a problem in a finite number of steps" (Britannica Concise Encyclopedia). "In writing a computer program to solve a problem, a programmer expresses in a computer language an algorithm that solves the problem, thereby turning the algorithm into a computer program". (Sci-Tech Encyclopedia)

7. http://wordnetweb.princeton.edu/perl/webwn?s=digitalcommunication

8. http://committees.comsoc.org/comt/

a distinction between algorithm producers and users. And this distinction has been confirmed by the people we interviewed, as the interaction between them (discussed further in Section).

These definitions highlight at least three successive steps to providing a statistical solution to a problem, leading to the production of software:

- first, the "translation" of a practical problem into "data", which means the identification of the physical patterns which must be captured, and their definition as digital data which can be computerized,

- second, the "expression of the algorithms", i.e. the different mathematical steps in the treatment of the data, and the results expected from the calculation,

- third, the "implementation" of the algorithms, or the production of compatible formats understandable by the chip or computer which will do the job[9].

This is the classic process of production of an application in software programming: a functional description of a problem, software development to implement it and the integration of this development into working processes and products. The result of such a process in the fields we study is a piece of software, using more or less dedicated "languages", such as VHDL for chip design. In addition to the production of software, these three domains contain FLOSS factories, as presented in the next subsection.

FLOSS FACTORY IN DATA PROCESSING

The second reason for studying data processing for knowledge-based sciences is that there are FLOSS initiatives, more or less visible at an institutional level. An exhaustive presentation of all of them is outside the scope of this paper, but some are quite visible on the Web, at algorithm level and at platform or chain-of-treatment level, and others are more local, as Table 2 shows. Here we selected, as our "unit of analysis" (Miles and Huberman (1994, p. 25), Yin's (2009) third component of research design), platform projects aiming to aggregate knowledge based on the accessibility of the participants; we wanted to have access to the initiator(s) of the project and we wanted them to try to be open to users and to contributions, as our goal was to study the participation of people to these projects. According to Yin (2009) a case study approach could be considered here, because the focus of the study is first to answer to the analysis of the process, and second to understand how and why the

participants in these platforms have been involved in these processes of knowledge production and feed these tools. And third, we want to cover contextual conditions, because they are relevant to the phenomenon under study. These platform projects offer us the opportunity to explore this phenomenon in context, using a variety of data, both an individual level, organizational, community, or research programs supported.

In remote sensing, we chose a library of open source programs, "Orfeo tool box", developed by the French space agency (CNES[10]). The aim of this library is to provide basic tools and algorithms to program the chains of treatment needed to use satellite data provided by the Pleiade satellite program[11].

In bio-computing, we identified two platform initiatives: the "Biogenouest" platform (the life science core facility network in western France) called "Bioside" (developed by Télécom Bretagne and the CNRS)[12], and that of the scientific information system service at Institut Pasteur, called Mobyle[13].

As far as digital communication is concerned, we were not able to identify any initiatives other than Open core[14].

Data collection and analysis

Data collection was performed between the end of 2008 and the end of 2009 and consisted of interviews (21 semi-guided interviews of more than 90 minutes each). All the interviewees were scientific professionals (researchers and research engineers), 18 belonging to public institutions (institutes of technology, CNRS, Institut Pasteur, CNES) and three to firms or private institutions. As the FLOSS movement has its roots "in the university and research environment" (Bonaccorsi and Rossi Lamastra, 2002, p. 6), one can argue that these producers should be more open to such arrangements[15]. We had to collect two types of points of view regarding the production of algorithms and their diffusion in the chosen disciplines: the algorithm producers and the producers of the platforms.

Regarding the platform side, we interviewed the initiators of the projects, at CNES (Orfeo Toolbox), at Télécom Bretagne and CNRS (Bioside) and at Institut Pasteur (Mobyle). Regarding the algorithm producers' side, we interviewed researchers from a CNRS laboratory called LabSTICC (Information and Communication Science and Technology Laboratory), which is run by Télécom Bretagne, a French institute of technology, in partnership with two universities in Brittany, Université de Bretagne Occidentale and Université

---

9. In the first case, this translation means the design of a chip (hardware translation via a dedicated software language VHDL, VHDL standing for "VHSIC hardware description language"), in the second the development of a program (software translation).

10. "Founded in 1961, the Centre National d'Etudes Spatiales (CNES) is the [French] government agency responsible for shaping and implementing France's space policy in Europe". http://www.cnes.fr/web/CNES-en/3773-about-cnes.php

11. http://www.orfeo-toolbox.org/packages/OTBSoftwareGuide.pdf

12. http://www.sb-roscoff.fr/SIG/spip.php?rubrique7

13. http://mobyle.pasteur.fr/cgi-bin/portal.py

14. http://www.opencores.org/

15. See, for instance, the open source science initiative, http://www.opensourcescience.net/, or the online initiative to cure tropical deseases, in biotech, http://pubs.acs.org/cen/science/84/8430sci1.html.

de Bretagne Sud. It brings together three poles working within one central theme: "From sensors to knowledge". The three poles are the following: MOM (microwaves and materials); CAC (communications, architecture and circuits); and CID (knowledge, information, decision)[16]. The fields covered by this laboratory include digital communication, remote sensing and tools for biologists. These initial points of view led us to interview other actors in the process of production-diffusion-use of an algorithm, in firms (chip design and remote sensing) or in public institutions (bio-computing, users of the platforms), because they were cited in the interviews and thus could provide us with a more global vision of the field than the people interviewed for a specific project, who were more project-centered, but also could confirm and specify how joint work is organized.

What we want to empirically appreciate in this paper is the participation of the researchers in FLOSS production and how this production is organized in a FLOSS factory. For this purpose, we collected information on the activities and production of the researchers and research engineers in the selected disciplines (bio-computing, remote sensing and digital transmission). We interviewed people about their representations of their production in their scientific environment and about the definition of what a "good algorithm" is. We also interviewed them on the existence of joint work, namely collaboration practices between the different actors contributing to and using the chains of treatment. Amongst the interviewees, some participate in the cooperative development of platforms of knowledge production (some open source, others not), while others do not. This allowed us to identify why they do or do not participate. We stopped the collection of new interviews when the exploration of the content of each new interview did not bring additional significant meaning (a summary of the methodology is available in the Appendix, table 1).

While the total number of interviews may appear low, their length allowed us to collect a fair amount of rich material (more detail and variance), and to identify some "coherence in attitude and social behavior" embedded in "a historical path, both personal and collective", to quote and translate Beaud (1996), the result expected from this kind of qualitative analysis. "This path is personal, as each interview describes the trajectory of a scientific actor, but also collective because it describes the specific scientific field it is embedded in" (ibid). According to Flyvbjerg (2011, pp. 301-316), referring to the definition of Merriam-Webster's Online Dictionary (Merriam-Webster, 2009), "case study focus on an "individual unit", what Stake (2008) calls a "functioning specific" or "bounded system".[...] Finally, case studies focus on "relation to environment"[...]". In our case, the emphasis is on analyzing the actors' relationships with their scientific knowledge production, and their environment.

After a brief introduction of the goal of the interview, the guide looked at the following dimensions: the career, the origin of the participation in the project, their definition of their contribution to the project is (based on interviees' own activity in the project), the significance of the project for the person's career, and the definition they give to their work[17].

Results

All the actors we interviewed agreed on the aim of the collaboration: proposing the best methods to extract "pertinent" information from physical data, using algorithms. However, this "best" does not mean the same thing to everyone: the algorithm producer would look at mathematics lock-ins to be solved, while the user would look at the quickest way to solve a problem (not always the most efficient), either looking at an already implemented algorithm to do the job or, when this is not possible, taking it to an algorithm producer who can understand them. This means that they have to invest time and money to understand each other, to develop "patterns of interaction". Actually this investment seems to be made more by specific actors (the "boundary spanners") than by the use of tools such as the open source platforms developed, in the project we studied. Non-use is not due to an a priori discrimination against FLOSS, but rather because these projects seem to provide less help to boundary spanners and few incentives to algorithm producers (in terms of reputation or of institutional incentive), whereas they have important extra investments to make the programs they developed open source.

The contextual factors: absorptive capacity and environment

A costly construction of the absorptive capacity and of the research agenda

On the application or process side, the algorithm is not integrated into the final product the same way.

- In biotechnologies the chain of algorithms (bio-computing) is only *a tool,* and the user (biologist) has little interest in it and *does not look at it*. Statistics are only here to "clean the data" (expression we heard)[18], to extract pertinent information from the data, to be able to perform biological tests on a small number of items (e.g. to select a small sample of genes able to produce a specific protein). Data analysis is part of the biological research process and of the article published, but is not really the core of the production.

- In signal processing there is a strong link between algorithms, and chips, and *the users* of the algorithms are chip designers and chip integrators (mobile phone manufacturers, for instance), so they *have good knowledge of algorithms,* and ask for algorithms that are not too complex to be implemented on chips. Consequently, there are two criteria for evaluation, namely the *efficiency* of the algorithm and its *practical usability,* that is to say its practical implementation in a circuit.

- Remote sensing falls in between: if *the design of the algorithm is driven by the application,* by the problem the users (called "thematicians") want to solve, then the algorithm producers are more involved in the definition of the knowledge which fits with the needs. In publication, there is an algorithm and description of the way it has been used. The difference between remote sensing and biology is that, in the latter, *data processing is part of the final result,* and not a pre-requisite before other biological tests.

  > "So, this interaction algo-circuit, it is... It is not that simple to make it live, because, you always need to maintain this double competency, because, if not, you are losing… well, at some point, if you are going too much on the circuit side, you are losing all the algo part, thus you cannot really be innovative; on the other hand, if you are going too much on the algo side, well, you do what the other algos do, thus…" (Joseph, 50, Professor in Digital Communications)

  > "We take others' data, we aggregate them and analyse them with statistical and bio-computing methods and we extract value added in terms of knowledge" (Mathieu, 30, engineer in Bio-computing).

While perception varies from one domain to another, all the interviews shed light on three common key structuring notions: those of *efficiency, productivity and usability.* The algorithm is developed with the scientific goal of making real data processing more efficient, i.e. to accelerate the work of the users, to clear their way, extracting pertinent information, allowing a decrease in the error rate, faster. In other words, it is taken as an external input by the users to make it possible to push back the boundaries.

Because of this search for efficiency, algorithms are very specialized and are, as we expected, a scientific specialization in themselves, far from the specialization of the disciplines of application (biology, geography, etc.). The consequence is a distinction between algorithm producers (called *algorithmists* in the rest of the article) and knowledge-based scientists, such as biologists or geographers (called *end-users* in the rest of the article). At the same time, the interaction described here - whether structured around an infrastructure such as a platform or centered on individuals - stresses the importance, and the personal "costs" for the researcher, of building a collective at work. One respondent used the eloquent expression of "bundle of skills" ("faisceau de compétences", Cyril - Ph. D. (Biology), Computer Service), when he described what a bioinformatics platform is. The collective building dimension, more than the individual, institutional or organizational ones, seems to be the key for producing results.

All the actors we interviewed agreed on the aim of the collaboration: proposing the best methods to extract "pertinent" information from physical data, using algorithms. This means, from the *algorithmist*'s side, identifying applied mathematics lock-ins and solving them, and from the *end-user*'s side, identifying either an algorithmist able to understand their problem and solve it, or some available algorithm to do the job that is already implemented. Otherwise, there is a human go-between, with knowledge from both worlds:

> "I am at the interface"
>
> [...]
>
> "A kind of interpreter role between the two communities, trying to reformulate the programmer's question to the biologist and then, to reformulate the biologist's answer to the programmer." (Cyril, 40, Ph. D. (Biology), Computer service)

Typically, although not always, these *frontier actors,* or *"marginal men"*[19] have a PhD in the algorithmic part but work in end-user labs, as research engineers, or graduated in biology but have turn to computer (because of the job market) and work for biologists in a computer service. For instance, in the field of biology, these new actors are named bio-statisticians, or bio-informaticians. They are able to choose amongst the great deal of knowledge proposed (being algorithms or applied maths researchers to work with), and to construct of the chains of production (aggregation of the pieces), that end-users use.

> "There is monitoring work to do, [which is] very important, and [which we] do not always do very well, there are lots of novelties, it is a fast-evolving domain, there are lots of, most of the time, we use programs which, eventually, are obsolete, there are others [that are] really better to do the same thing and we do not know them" [...] (Mathieu, 30, Computer engineer in biocomputing)

In any case, constructing these interactions means investing. Either people develop the standard pieces of software, institutions hire dedicated people as bridges, or researchers have to cross the border themselves. And it seems that the more specific the problem raised, the closer the interaction between algorithmists and end-users must be.

---

19. For a study of that role in the case of engineering, see Evan (1964)

This has consequences for the way researchers on the algorithm side choose the scientific question they want to tackle. We found three strategies, based on three research agendas, leading to three different conceptions of what an algorithm is.

The first strategy we identified is driven by the research agenda in mathematics, for which the topics of investigation are more theoretical. The aim is to produce the "best algorithm possible" in terms of mathematical performance. This means "working alone, without taking the implementation into account" (Anne, 30, associate professor in applied mathematics for Digital Communications). Regarding usability, it is quite like working on "a huge labyrinthine system" which "will never be implemented" («*une usine à gaz, qui ne verra jamais le jour*», Anne). However, it is useful as a "benchmark, a theoretical frontier in terms of performance" for the second and third strategies. The second strategy is more driven by practical implementation and involves looking for "a less efficient algorithm, but an implemented one" (ibid). In the third strategy, the algorithm is developed with the scientific goal of making real data processing more efficient, i.e. to accelerate and facilitate the job of end-users: to clear the way, extracting relevant information that allows a decrease in the error rate, and makes it faster. In other words, it makes it possible to push back the boundaries. This third strategy is driven by practical implementation and requires a "dialogue" (Gurvan, 50, professor in remote sensing) between the designer of the algorithm and the person or people who implement(s)/use(s) it. This dialogue may be directly between *algorithmists* and *end-users* to define the problem to be solved, in other words the information to be extracted from the physical data, or it might be between algorithmists and data providers, to understand what these data contain. According to our interviewees, the second or third strategies of research are not necessarily easier to solve than the first. But in that case, efficiency matters more than novelty or theoretical strength, relying on a formal mathematical demonstration, for instance. But for these two last strategies, the implementation, i.e. the software development of the algorithm matters, so they should be more willing to participate in FLOSS projects, whose goal is facilitating the transfer to users.

The environment

This selection of programs and the verification that they work together appears costly, and in order to increase end-users' productivity, efficiency, and usability (terms present in interviewees' discourse), software may play this go-between role, when the process of treatment is "standard" and "common knowledge" amongst end-users. The actors have developed platforms with the goal of collecting this common knowledge to propose standard chains of treatment.

In the case of remote sensing, for example, the Matlab platform (proprietary software), and its open source competitor, Scilab, were cited by two researchers we interviewed, and these standard developments are sold as components of the platform, or as extra libraries.

The reason for these dedicated platform development initiatives, is that their initiators bet they will help them to better serve these demands, because they have new material (for example CNES with the Pleiade satellite series) or recurrent demands for data analysis (as with the computer support for scientific analysis service at Pasteur), or because of the increasing coordination costs due to dealing with compatibility issues between modules whose source and behavior are closed (chips, see http://opencores.org/opencores,mission).

The reasons for open-sourcing these platforms are linked to standards competition (as explained by Muselli, 2004): free provision facilitates the evaluation of this experience-good[20] and thus its adoption, but also the development of complementary technologies (the programs working on it). The reason for this is that it is a platform whose value lies as much in its facility of use as in the library of programs working on it, and it eventually acts as a tool to federate and coordinate the community.

For the time being, as far as bio-computing platforms are concerned (Mobyle and Bioside), the designers have implemented open-source programs in their platforms by themselves. In remote sensing (Orfeo), the institution (CNES) has paid researchers to design and format programs, according to our interviews, in order to increase the coordination and the reuse of algorithms, and to produce, or be able to use new algorithmic knowledge.

> "What I put in this... this particular library, are things asked for by CNES and, actually, it is CNES which is encouraging the development of this particular library." (Maurice, 30, associate professor in Remote Sensing)

Question: "And these programs, they were developed here, at Pasteur?"

> "Very few of them. I think twenty out of the 230, 10% are developed either by our division, or, sometimes, by researchers who have taken a keen interest in programming, and we have installed their programs because they work well." (Cédric, 50, computer engineer in a Biocomputing support service)

This could be explained by the novelty of the platforms and the fact they have not yet reached a critical mass allowing the automatic functioning of an increasing returns process[21], as in a classical network effect (Katz and Shapiro, 1985). In that respect, platform sponsors face the classic

---

20. Defined as a good a user needs to use to be able to value, such as information. See Shapiro and Varian (1999, p. 5).

21. Due to technical interrelation (Arthur, 1989): the more programs there are available, the more users these platforms may have, and the more users they have, the more interest the algorithm producers may have in contributing.

difficulties for initiating what would be a virtuous cycle for them; it ends up being a waiting-game from both users and producers:

- first of all, if these platforms are open (and free) in access for users, they are not (yet?) for the developers, as traditional FLOSS projects are. The designers do not really believe outside people can collaborate on the development of the platform by itself, and there is neither injunction from the employer, nor strong support;

Question: "And, you said it was not yet on the agenda to open... well, to have a collaboration... How would you imagine this organization?"

> "Well, this precisely, is difficult, because it depends on what we would like to get from it, because, either we kept on developing, but we would delegate some parts to people willing to involve themselves, or if we were not able to develop any more, we would not have the point of view on the project, people would take the lead [....] But, actually, I do not know exactly, it is just... actually, it is the people who asked "do you want to do it?" (Nestor, 40, computer engineer in Biocomputing, PhD in Biology).

> "The scientific direction is happy we do that, because they understand we are one of the departments open to the outside" (Cédric, 50, computer engineer in a Biocomputing support service).

> "In a way, the network [RENABI] aims to respond to the lack of will from the institutes and from the State to federate the bio-computing facilities. So all of the interested actors are networking and collaborating. But it is not really supported." (Mathieu, 30, bio-computing Engineer)

- in addition to this, the algorithms did not mention injunction or specific reward from their employer to contribute to these platforms, a point we will come back to later,

- on the other hand, the interest in new algorithms is far from being obvious for end-users, as they do not understand them. But these users have to cope with the increasing amount of data and the competition between laboratories, making it necessary to develop these new algorithms, and make them faster / more efficient in terms of error rate.

> "Everything is open, but it is not enough. Most of the time, the biologist uses the same things each time, he already knows, he comes back to the same web sites, etc. and... most of the time, he is... He doesn't know the best tools, the researcher in bio-computing, in his sub-field of competency, he brings a real added value." (Mathieu, 30, computer engineer in Biocomputing)

> "[Programs] are free (open source), but one needs to install them in an appropriate environment, and thus, according to the type of data and to the constraints of the algorithm, regarding the memory, the speed of execution, or I do not know what, means that it is not obvious a biologist can use it. You need an expert both to install it and..." (Cédric, 50, computer engineer in a Biocomputing support service)

So the fact remains that end-users still need an expert, a frontier user, or institutional support to detect, finance and process these new algorithms. As far as institutional FLOSS platforms are concerned, frontier users say that the workflow may be useful for standard chains of production, but also that they may have already programmed these chains, and their program is as easy to use for them as the workflow. Thus, the added value of a FLOSS platform for the end-users of algorithms is weak.

> "I start from a set of tools that I know [...] it is when a new issue arises that I will look elsewhere, another tool [eg a different algorithm] [...] There is no method that consensus from everyone." [...] I find information on new technologies and algorithmic tools in publications, the 'R' package provides development, and I am able to fend for myself to implement it." (Diane, 43, research engineer, Bio-computing)

Given the background of these FLOSS platforms, it seems that there is a recognized need for such a tool, to improve producers and users' efficiency, providing standard chains of treatment. On the other hand, these projects came more from computer services than from a global demand from the whole community or as a strategic goal for institutions. This point leads to the question of why the producers have not invested more in the platform to diffuse their production.

FLOSS, A WORTHWHILE INVESTMENT FOR PRODUCERS?

All the researchers we met are open-minded regarding FLOSS. They may use FLOSS-based computers (with GNU/ Linux operating system, or/and Firefox browser) and some disciplinary FLOSS programs, which can "facilitate their work" because they do not have to "redevelop standard applications". Most of them (both algorithmists and end-users) think that, at a global level, having access to FLOSS programs implementing the algorithm would facilitate access to "knowledge" (of the existence of a new algorithm, of its performance, of how to use it, etc.)

But, as far as algorithm producers are concerned, even if they have a positive opinion of open source, open-sourcing the program they have developed when conceiving a new algorithm (to test it) is not obvious.

> "That is, I do plenty of things otherwise, that I do not systematically put. [...] Because... well, for

the time being, I judge it is not mature enough, to propose it as a FLOSS, because it needs to be quite under control. The development is, all the same, quite... quite important, to do open source, you need the open source program working, you need comments everywhere, you need guarantees of... of quality which are rather more high than if you simply make the program for your own machine, with data you control quite well. So, this is why, you have to do the ratio between the time spent to make a good program to provide the FLOSS and the time spent only to develop the method, validate it to, finally publish it directly." (Maurice, 30, associate professor in Remote Sensing)

Question. "Do you open source your programs?"

"No, clearly, because, most of the time, we make our stuff as it is, and them, when speaking about handing over [the piece of software] it requires re-writing..."

Question. "This is it. So, you give up"

"Well, it takes time, so, of course". (Delphine, 40, Professor in Digital Communications)

It is not just a question of publishing the program they have "cobbled together" for their own need, under a free, open source license. It is a question of making a "clean", "stable", "robust" program, with documentation, thus of doing extra work[22]. A part of the work is "do-it-yourself", especially regarding the development of software to test hypotheses. But the investments needed to make the experience replicable, for the industrialization of software, is too demanding.

This weak appetite for publishing FLOSS programs should also be partially explained by the fact that the gains expected do not cover this extra investment.

The reason for researchers to participate in platforms or standards development is that they have been asked to do so and supported, which is interesting for funding new research, but also to make their work better known. In the remote sensing and bio-computing cases we studied, researchers were funded to contribute, developing "clean" software, i.e. well-developed, documented programs, and in these cases, developed under rules that make them work with other programs. In electronics, if a patent is published on a technology (an algorithm and a circuit, most of the time) and integrated into a norm, this generates financial flow for the lab (and for the researcher).

They may also have institutional incentives to publish "clean" software, if it is requested within a bilateral cooperation (research contract), but this remains marginal in the activity of the researcher (i.e. the algorithm producer). If the

notion of scientific engineering is relevant when looking at scientific production, the reward for researchers is the same for all, and based on classical science behavior: the publication of articles in recognized scientific journals within the discipline. The researchers interviewed confirmed the attachment to the academic scientific values that Lamy and Shinn (2006) found in other fields. But the process is quite long (time for an article to be accepted, time for establishing one's reputation), and uncertain (will the article be accepted?)

Interviewees know and practice other channels of diffusion: bilateral agreement (secret), or IPR system (patents and their exploitation in bilateral or standard setting organizations, software diffusion and the choice of the license), but before being selected by users (especially for contractual relationships), one has to have solid scientific recognition, and thus to have published.

"Actually, publication is also a showcase of what we can do. And the contract we won with [....], was won precisely because we show our competencies, acknowledged by publications, in that domain." (Maurice, 30, associate professor in Remote Sensing).

Thus, the user system, and the specific demands set by the researchers' employer (the organization) to be part of this system, may be, in the short run, more interesting. This is why researchers agree to participate, in a contractual basis, in applied projects aiming to develop FLOSS platforms, providing FLOSS programs. But in their everyday work, there is no such incentive, as the publication of the source code of the program used to process the simulations joined to the presentation of the method is neither compulsory nor common practice in the disciplines studied. Some researchers also pointed to the fact that publishing software may even be counter-productive in the scientific perspective, as providing the software program to competing labs helps them to close the gap more quickly, thus diminishing the original inventor's room for publication, access to collaboration with the industry and therefore to financing, etc.

In conclusion, the costs for open sourcing work are perceived as high and the gain for the better only slightly positive.

On the other hand, access to new algorithms is facilitated by their open-sourcing, as it accelerates the appropriation of new algorithms. This point was stressed by all the people we met. It is not a necessity as these users understand the methods published in scientific journals and are able to re-develop the implementation of the method, whether it is software (bio-computing, remote sensing) or hardware (digital communications), but it is a real added value.

To sum up, we are facing a classic collective action paradox: even if agents agree on the fact that a FLOSS organization will improve the efficiency of the domain, on both the

---

22. Stallman and al. (First publication 1992, this version, 2009) explain what Free Software is for the creator of the idea.

algorithmist side and the end-user side, no one seems to be ready to invest the extra cost for producing the IP.

> "Why have we lost this leadership? Because nobody could dedicate himself 100% to the project, for many people it was something in addition [to the day-to-day work], there is something like activism in that, and eventually, if you cannot direct enough human resources to the project there are risks of loss of impetus." (Cyril, Computer Service, 40)

Does this disqualify the idea of having a FLOSS factory without a FLOSS IP regime? We address this question in the conclusion.

Conclusion: FLOSS without user as innovator?

This study of "applied" science knowledge production has shown that today, FLOSS is not a solution on its own to knowledge diffusion, because there are not enough incentives for researchers to publish their software, and there is a need for extra investments to integrate the software produced into a chain of production. The design of standard platforms may help for that, but for the time being, knowledge producers still don't have incentives to contribute to these platforms. When users are not producers, in contrast to the traditional FLOSS incentive regime supporting a FLOSS factory, the other two traditional regimes do not seem able to take over in the long run. This situation may change with the evolution of the system of evaluation in science, which has been initiated in bio-computing.

Actually, though the impact of publication of a piece of software seems low (but not null) in remote sensing or in digital communication, according to our interviewees, it is quite important in biology (new algorithms for biology). The availability of a program (usable, thus under an open source license) is said to increase the citation of the original article. This is because, when an algorithm is used in a biological experiment, the article is cited in the biological article by the frontier actor, namely the bio-statistician who co-authors it and "has to put a paragraph in the article about the statistical techniques used". For some scientific publications of new algorithms, the program must even be open sourced[23]. And because some frontier users can understand it and must cite the original publication when using an algorithm, there is a direct classical science reward.

This could be combined by an increase in the interest for some platforms from the algorithmist side[24]. In those sciences, there are problems in evaluating the efficiency of the algorithm, and comparing it to an already existing one, replicating the simulations from the original paper. Thus, at the research community level, a collective platform implementing standard chains of treatment and providing standard data could be of interest for the community. This would facilitate the replication of the tests performed on the programs and the benchmarking between two algorithms. According to some researchers we met, this evolution could be facilitated by the increase of the level in programming skill in the younger generation of researchers (they develop "better", "cleaner" programs, even for their own needs). In that scheme, at least for the algorithmist side, it would be a way to create a FLOSS IP regime, as the producers would become the users of the platform.

But to transfer the finding to users, the existence of *frontier users* is still a necessity, and these specialists do not really need the second step of codification which is the platform, the workflow: how using the new algorithm, linked with which program, for which kind of thematic problem, etc. So capitalization remains at the individual level and is not really transferred to end-users. In these communities, we have not identified actors who are both producers and users. The best proxy is the frontier user (or *boundary-actor*), but who remains a user (and not a producer of original algorithms).

References

Anderson, John B. (2005). Digital Transmission Engineering. The Institute of Electrical and Electronics Engineers. Second Edition.

Arthur, W. B. (1989). Competing technologies: an overview. In Dosi, G.; Freeman, C.; Nelson, R.; Silverberg, J.; Soete, L., editors, Technical Change and Economic Theory, pages 590–607. Frances Pinter, London.

Balka, Kerstin; Raasch, Christina; Herstatt, Cornelius (2009). Open source enters the world of atoms: A statistical analysis of open design. First Monday, 14(11).

Beaud, Stéphane (1996). L'usage de l'entretien en sciences sociales. plaidoyer pour l'"entretien ethnographique". Politix, 9(35): 226–257.

Beaud, Stéphane; Weber, Florence (2003). Guide de l'enquête de terrain: produire et analyser des données ethnographiques. La Découverte, Paris.

Bertaux, Daniel (2010). Le récit de vie. L'enquête et ses méthodes. col. 128, Armand-Colin, 3 edition.

Bessen, James (2005). Open Source Software: Free Provision of Complex Public Goods. Technical report, Research on Innovation. URL: http://www.researchoninnovation.org/opensrc.pdf.

Blanchet, Alain; Gotman, Anne (2010). L'enquête et ses méthodes. col. 128, Armand-Colin, 2 edition.

Bonaccorsi, Andrea; Rossi Lamastra, Cristina (2002). Why Open Source Software Can Succeed. LEM working paper, (2002/15).

---

23. See the "software" section in Bioinformatics journal intruction to the authors, http://www.oxfordjournals.org/our_journals/bioinformatics/for_authors/general.html

24. We thank Patrick Meyer and Patrick Hénaff (Télécom Bretagne) for having formulated this idea in an informal discussion we had with them.

Clément-Fontaine, Mélanie (2002). Les licences libres. analyse juridique. In (Jullien et al., 2002),, editor, Nouveaux modèles économiques, nouvelle économie du logiciel, pages 163–177.

Clément-Fontaine, Mélanie (2009). L'œuvre libre. Juris-classeur PLA, fasc 1975.

Cohen, M. D.; March, J. G.; Olsen, J. P. (1972). A garbage can model of organizational choice. Administrative Science Quarterly, 17(1): 1–25.

Crowston, Kevin (2011). Lessons from volunteering and free/libre open source software development for the future of work. Turku, Finland. Springer. URL: http://crowston.syr.edu/system/files/ifipwg82paper_final.pdf.

Crowston, Kevin; Fagnot, Isabelle (2008). The motivational arc of massive virtual collaboration. The IFIP WG 9.5 Working Conference on Virtuality and Society: massive Virtual Communities, Lüneberg, Germany (2008).

Crowston, Kevin; Howison, James; Annabi, Hala (2006). Success in free and open source software development: Theory and measures. Software Process Improvement and Practice, 11: 123–148.

Dang Nguyen, Godefroy; Genthon, Christian (2006). Les perspectives du secteur des Tic en Europe (Perspectives for the European ICT Industry). Marsouin Working Paper, (4-2006).

DeLone, W. H.; McLean, Ephraim R. (1992). Information systems success: The quest for the dependent variable. Information Systems Research, 3(1): 60–95.

DeLone, W. H.; McLean, Ephraim R. (2002). Information systems success revisited. In Proceedings of the 35th Hawaii International Conference on System Sciences, Waikola, Hawaii. HICSS-35. URL: http://csd12.computed.org/comp/proceedings/hicss/2002/1435/08/14350238.pdf.

DeLone, W. H.; McLean, Ephraim R. (2003). The delone and mclean model of information systems success: a ten-year update. Journal of Management Information Systems, 19(4): 9–30.

Denzin, Norman K.; Lincoln, Yvonna S., editors (2011). The Sage Handbook of Qualitative Research. Sage, Thousand Oaks, CA, 4 edition.

Evan, W. M. (1964). The engineering technician: dilemmas of a marginal occupation, In Berger, P. L., editor, Human Dimensions of Work, Studies in the Sociology of Occupations, pages 36–63. Macmillan, New York.

Farrell, J. (1989). Standardization and intellectual property. Jurimetrics Journal.

Feller, J.; Fitzgerald, R.; Hissam, S.; Lakhani, R. K., editors (2005). Perspectives on free and open source software. MIT Press, New York.

Flyvbjerg, Bent (2011). Case study. In (Denzin and Lincoln, 2011),, editor, The Sage Handbook of Qualitative Research, chapter 17, pages 301–316.

Foray, Dominique (2004). The Economics of Knowledge. MIT Press.

Foray, Dominique; Cassier, Maurice (2001a). Économie de la connaissance: le rôle des consortiums de haute technologie dans la production de bien public. Économie et Prévision, 4/5(150-151): 107–122.

Foray, Dominique; Cassier, Maurice (2001b). Public Knowledge, Private Property and the Economics of High-tech Consortia. Economics of Innovation and New Technology, 11(2): 123–132.

Foray, Dominique; Zimmermann, Jean-Benoît (2001). L'économie du logiciel libre: organisation coopérative et incitation à l'innovation. Revue économique, 52: 77–93. Special Issue, hors série : économie d'Internet, sous la direction d'É. Brousseau et N. Curien.

Forte, A.; Bruckman, A. (2005). Why do people write for Wikipedia? Incentives to contribute to open-content publishing. working paper.

Hackman, J. R. (1987). The design of work teams, In Lorsch, J. W., editor, Handbook of Organizational Behavior, pages 315–342. Prentice Hall, Englewood Cliffs, NJ.

Hemetsberger, Andrea; Reinhardt, Christian (2009). Collective development in open-source communities: An activity theoretical perspective on successful online collaboration. Organization Studies, 30: 987–1008.

Hess, Charlotte; Ostrom, Elinor (2006a). Introduction: An Overview of the Knowledge Commons. In (Hess and Ostrom, 2006b),, editor, Understanding Knowledge as a Commons. From Theory to Practice, pages 3–26.

Hess, Charlotte; Ostrom, Elinor, editors (2006b). Understanding Knowledge as a Commons. From Theory to Practice. MIT Press.

Hope, Janet (2008). Biobazaar: The Open Source Revolution and Biotechnology. Haward University Press.

Joly, P.B.; Hervieu, B. (2003). La marchandisation du vivant. Futuribles, (292).

Jullien, Nicolas; Clément-Fontaine, Mélanie; Dalle, Jean-Michel (2002). Nouveaux modèles économiques, nouvelle économie du logiciel. Technical report, Projet RNTL, 218 pages. URL: http://www.marsouin.org/article.php3?id_article=78.

Jullien, Nicolas; Roudaut, Karine; le Squin, Sandrine (2011). L'engagement dans des collectifs de production de connaissance en ligne. Le cas GeoRezo. Revue française de socio-économie, 8(2): 59–83.

Katz, M. L.; Shapiro, C. (1985). Network externalities, competition, and compatibility. American Economic Review, 75: 3: 424–440.

Kogut, B. M.; Metiu, A. (2001). Open source software development and distributed innovation. Oxford Review of Economic Policy, 17(2): 248–264.

Lacolley, Jean-Louis; Loilier, Thomas; Tellier, Albéric (2007). La prise de décision dans les équipes de la communauté des logiciels libres: Faut-il mettre le bazar dans la poubelle? In XVIe Conférence Internationale de Management Stratégique, Montréal.

Lakhani, Karim; von Hippel, Eric (2003). How open source software works: Free user to user assistance. Research Policy, 32: 923–943. URL: http://opensource.mit.edu/papers/lakhanivonhippelusersupport.pdf.

Lakhani, Karim; Wolf, R. (2005). Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects, in Feller et al. (2005) edition, pages 3–22.

Lamy, Erwan; Shinn, Terry (2006). L'autonomie scientifique face à la mercantilisation. formes d'engagement entrepreneurial des chercheurs en france. Actes de la recherche en sciences sociales, 4 (164): 23–50.

Latour, B.; Woolgar, S. (1979). Laboratory Life: The Social Construction of Scientific Facts. Sage Publications, Beverly Hills.

Lee, Sang-Yong Tom; Kim, Hee-Woong; Gupta, Sumeet (2009). Measuring open source software success. Omega, 37(2): 426–438.

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

Li, Qing; Heckman, Robert; Crowston, Kevin; Howison, James; Allen, Eileen E.; Eseryel, U. Yeliz (2008). Decision making paths in self-organizing technology-mediated distributed teams. In Proceedings of the International Conference on Open Source Systems, Paris, France, 14-17 December.

Lundvall, B.; Johnson, (1994). The learning economy. Journal of Industry Studies, 1(2): 23 – 42.

Marwell, G.; Oliver, P. (1993). The Critical Mass in Collective Action: A Micro-Social Theory. Cambridge University Press, Cambridge.

Merriam-Webster, (2009). Online Dictionary. URL: http://www.merriam-webster.com/dictionary/.

Miles, Matthew B; Huberman, A Michael (1994). Qualitative Data Analysis: An expanded Sourcebook, volume 2nd. Sage Publications.

Mockus, A.; Fielding, R.T.; Herbsleb, J. D. (2000). A case study of open source software development: The apache server. pages 263–272, Limerick, Ireland. Proceedings of The International Conference on Software Engineering (ICSE'2000).

Muselli, Laure (2004). Les licences informatiques. un instrument stratégique des éditeurs de logiciels. Réseaux, 3(125): 143–174.

Nilges, Michael; Linge, P. Jens (2002). A definition of bioinformatics. In Offermanns, Stefan; Rosenthal, Walter, editors, Encyclopedia of molecular pharmacology. Springer Verlag. URL: http://www.pasteur.fr/recherche/unites/Binfs/definition/bioinformatics_definition.html.

Nov, Oded (2007). What motivates wikipedians? Communications of the ACM, 50: 60–64.

O'Mahony, Siobhán (2003). Guarding the commons: how community managed software projects protect their work. Research Policy, 32(7): 1179–1198. Open Source Software Development.

O'Mahony, Siobhán (2007). The governance of open source initiatives: what does it mean to be community managed? Journal of Management and Governance, 11(2): 139–150.

O'Mahony, Siobhán; Bechky, Beth A. (2008). Boundary organizations: Enabling collaboration among unexpected allies. Administrative Science Quarterly, 53: 422–459.

Scacchi, Walt (2007). Free/open source software development: recent research results and emerging opportunities. In ESEC-FSE companion 07: The 6th Joint Meeting on European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering, pages 459–468, New York, NY, USA. ACM.

Schweik, Charles M. (2006). Free/Open-Source Software as a Framework for Establishing Commons in Science. In (Hess and Ostrom, 2006b), editor, Understanding Knowledge as a Commons. From Theory to Practice.

Seddon, P. B. (1997). A respecification and extension of the delone and mclean model of is success. Information Systems Research, 8(3): 240–253.

Shah, Sonali K. (2006). Motivation, governance, and the viability of hybrid forms in open source software development. Management Science, 52(2): 1000–1014.

Shapiro, C.; Varian, H. (1999). Information Rules: A Strategic Guide to the Network Economy. Harvard Business School Press.

Short, Nicholas M. (2009). Remote sensing tutorial. Technical report. URL: http://rst.gsfc.nasa.gov/.

Stake, Robert E. (2008). Case Study, (?) edition, pages 119–150.

Stallman, Richard; al., (First publication 1992, this version, 2009). GNU Coding Standards. Free Software Foundation. URL: http://www.gnu.org/prep/standards/standards.pdf.

von Hippel, Eric (1988). The Sources of Innovation. Oxford University Press, New York.

von Hippel, Eric; von Krogh, George (2003). Open source software and the "private-collective" innovation model: Issues for organization science. Organization Science, 14(2): 209–223.

Von Krogh, George; Spaeth, Sebastian; Lakhani, Karim R. (2003). Community, joining, and specialization in open source software innovation: A case study. Research Policy, 32(7): 1217–1241.

Wasko, Molly McLure; Faraj, Samer (2005). Why should i share? examining social capital and knowledge contribution in electronic networks of practice. MIS Quarterly, 29(1): 35–57.

Yang, Heng-Li; Lai, Cheng-Yu (2010). Motivations of wikipedia content contributors. Computers in Human Behavior, 26(6): 1377–1383. Online Interactivity: Role of Technology in Behavior Change.

Yin, Robert K. (2009). Case study research: design and methods. SAGE, 4th edition.

Zachte, E. (2007). Wikipedia statistics-tables-english. Technical report. URL: http://stats.wikimedia.org/EN/TablesWikipediaEN.htm.

Zhang, Xiaoquan; Zhu, Feng (2011). Group size and incentives to contribute: A natural experiment at chinese wikipedia. The American Economic Review, 101(4): 1601–1615.

APPENDIX

TABLE 1

Research methodology and interview guide

Choice of the methodology	Qualitative research approach by means of semi-structured interviews with individual actors participating (or not) in collaborative platform projects (FLOSS factories). Methodology based on Beaud and Weber (2003), Blanchet and Gotman (2010), Bertaux (2010). We identified the players interviewed given a) their visibility in the selected collaborative platform projects, or b) their visibility in a disciplinary field in which the selected projects are developed. In our investigation, we have always given priority to the individual (personal and professional) dimension of the contribution to these projects against the institutional dimension.
Data collection	Development of an interview guide based on the one used in a research project on on-line communities of practice (ANR CCCP-Prosodie). Semi-structured interviews (21) with researchers and research engineers in the disciplines selected, contributing (or not) to these platform projects. Recording and full transcript of the interviews.
Interview guide. Adapted from http://www. cccp-prosodie.org/spip. php?article40	From interviews, we wanted to include the participation, or lack of participation, in this type of project, the terms of participation (eg. if it was part time or not), and a commitment (or not) to the the idea defended by these platforms in the scientific production. The main themes developed in the guide were: • Description of who the actor is, his job, his activities. • Description of what he does in the selected project, the way he contributes to the project (production, how he works and produces knowledge, role, relation to the production, relation to the project). • Description of the project's activities (relations, actors, activities). • Commitment to the project: description of the process steps of the actor's commitment to the project, from the first contribution to greater involvement. The goal is to identify the various registers of the commitment. Evolution of the actor in the career of the project. • Evaluation by the interviewee of his project participation (relative to the project itself, the job, and the scientific field). • Professional career.
Data analysis	Classical sociological manual procedure to analyze thematic content of the interviews: a horizontal analysis, i. e., for each interview the characterization of the main themes emerging from the discourses, then a transverse or vertical analysis for all interviews, i. e. the identification of the common themes amongst the interviews, and the confrontation of the actors' positions on each theme. Content analysis on each selected disciplines, then on all the software platform projects.

TABLE 2

A summary of different open source initiatives by fields.

Open source initiatives	Bio-computing	Remote sensing	Signal processing
Piece of knowledge (algorithms)	Open source programs. http://madswichmann.dk/wordpress/?page_id=15for some examples of bio-computing programs, more or less specific.	See Open Source Geospacial web site (OSGeo) http://www.osgeo.org/	Digital signal processing open core. Ex: Philips' subsidiary (NXP) initiative: http://www.nxp.com/news/backgrounders/bg0016/
platforms (set of algorithms, chains of treatment)	Workflow platforms: Mobyle project (Pasteur Institute), Bioside project (Ouest Genopole)... The aim is to create a "distribution" of programs, with a process to package the programs and making them work together. It is comparable to Linux "distributions" such as Debian or RedHat.	Library of open source programs: Orfeo tool box (CNES). The aim of this library is to provide basic tools and algorithms to program the chains of treatment needed. http://www.orfeo-toolbox.org/packages/OTBSoftwareGuide.pdf	Proposition of open source designs of chips implementing algorithms and thus which can be used to develop a complete system http://www.systemc.org/ Open core initiative: http://www.opencores.org/ The difference with the two other cases is there are hardware designs rather than software design to simulate/test a chain of treatment (open source software simulating a communication channel for instance).