RISE AND FALL OF PLATFORMS: SYSTEMATIC ANALYSIS OF PLATFORM DYNAMICS THANKS TO AXIOMATIC DESIGN

Julien Legrand, Pascal Le Masson, Maxime Thomas, Benoît Weil

RISE AND FALL OF PLATFORMS: SYSTEMATIC ANALYSIS OF PLATFORM DYNAMICS THANKS TO AXIOMATIC DESIGN

Legrand Julien
Center for management science
Mines Paristech
legrand.julien@mines-paristech.fr

Le Masson Pascal
Center for management science
Mines Paristech
le_masson.pascal@mines-paristech.fr

Thomas Maxime
Center for management science
Mines Paristech
thomas.maxime@mines-paristech.fr

Weil Benoît
Center for management science
Mines Paristech
weil.benoit@mines-paristech.fr

ABSTRACT

While platforms are multiplying across industries, the laws governing their dynamic are still poorly understood. The high diversity of disciplines covering the topic, spanning from strategy management to engineering design, made it difficult for any new model to integrate the numerous phenomena at stakes. In a new effort to bring them together, we exhibit Suh's Axiomatic Design as an ideal framework to systematically analyse platforms dynamics when market and technology forces meet and interact. Exporting the current description of platforms from Design Structure Matrices to Design Matrices, our research enables us to systematically explore platforms potential evolutions. While the model leads us to rediscover classical behaviours, it also uncovers new results, such as situations of split leadership and platform overthrow, in which complementors challenges the platform leader. Both can be linked to two necessary conditions: functional generativity and technical genericity. We then identify those behaviours in several industrial cases.

Key words: Platform strategies, design engineering, design theory, innovation, market implications,

1. INTRODUCTION

Platforms have been widely studied over the past three decades, for they seem to be a very efficient way of organizing product lines (Meyer, 1997), distributing innovation efforts among partners (Gawer & Cusumano, 2002) or dividing tasks while mitigating risks associated to their development (Baldwin & Clark, 1994).

While the description of a given platform at a given time now seems to be well established since (Baldwin & Woodard, 2008) work, several scholars have stressed that today's challenge is now to understand the laws governing their evolution in time (Gawer, 2014) (Gawer & Cusumano, 2014) (McIntyre & Srinivasan, 2016).

Bearing this goal in mind, the present paper introduces a new model bridging both market and technology constrains, based on Axiomatic Design (Suh, 1990). Thanks to this model, we shed new light on platform stakeholders' behaviour when they face the emergence of a new technology or a new market. While the model lets us rediscover behaviours scholars are already familiar with, it also leads us to identify two surprising platform configurations: situations of split leadership, in which two or more platform leaders cohabitate, and platform overthrow, in which a complementor becomes the new platform, turning the old one into its own complementor. We also find that the apparition of a new generic technique and the emergence of new functions are two prerequisites for such behaviours. Those new levers of action, at the heart of platforms' rises and falls, subsequently call for a new type of stakeholder to manage them, which we call a "hyper-hub": we conclude this paper by outlining its main characteristics.

Following this introduction, our discussion will be divided into five parts. The first will position this paper within the broader platform literature, identifying the key problems it addresses and explaining the reasons that led us to choose Axiomatic Design theory to solve those problems. The second will aim at transposing well-known properties of platforms in the context of this theory, partly thanks to the strong links between Axiomatic Design's Design Matrices (DM) and Design Structure Matrices (DSM). The third paragraph will analyse the impact of technical and functional generativity on the platform ecosystem, identifying the conditions under which split leadership and platform overthrow are possible. The fourth will then re-examine eight well-known case studies, proving that the overthrow dynamic and the split leadership are common in the industry. Finally, the fifth and last part concludes on our findings, reminds the limitations of our model, outlines the concept of "hyper-hub" and lists directions for further research.

2. LITERATURE REVIEW

Platforms' characteristics have been extensively studied and summed up in (Baldwin & Woodard, 2008), (Gawer, 2014) and (McIntyre & Srinivasan, 2016). It is widely acknowledged that the platform itself is divided into two major types of components: a very stable core and rapidly changing components. The core was identified as the source of platform leadership (Gawer & Cusumano, 2002) as it would organize the platform ecosystem – as described in (Iansiti & Levien, 2004) – along with a set of design rules (Baldwin & Clark, 1994). The other components were identified, in turns, as either complementors or modules, depending on the scholar's tradition: respectively, strategy / management (Yoffie & Kwak, 2006) or technology management / engineering design (Baldwin & Clark, 2003). The terms are covering a similar reality, at least within the platform literature: those are third-party companies developing their own technical solutions built upon the common foundations established by the platform core to address niche market needs. Beside those two types of components come two other concepts. Network effects, as studied in (Tirole & Rochet, 2003), emphasizes the self-reinforcing effect of platforms on both users and complementors. Lastly, (Tee & Gawer, 2009) proved that industrial architecture (Jacobides, et al., 2006) played an important role in the success of platforms.

While platforms' characteristics are well-understood, the same cannot be told about the laws governing their rises and falls (Gawer, 2014) (Gawer & Cusumano, 2014) (McIntyre & Srinivasan, 2016). Scholars from heterogeneous fields have contributed to describe specific platform behaviours. The building of a platform leadership was studied by (Gawer & Cusumano, 2002) who identified four levers that industry players should consider when designing their platform strategy. (Henderson & Clark, 1990) introduced the concept of architectural innovation – to be distinguished from incremental and radical innovations – which happens at the interfaces, in between platforms' core and complementors. (Bresnahan & Greenstein, 1999) and (Armstrong, 2006) studied the direct competition between platforms, while (Yoffie & Kwak, 2006) as well as (Gawer, 2014) noticed that competition could also happen between the platform leader and its complementors, though no specific theory of this behaviour was provided. Lastly, (Eisenmann, et al., 2011) analysed how platforms as well as complementors could leverage their user base to compete between each other through product bundles.

Strikingly, those identified platform behaviours come from heterogeneous disciplines. As described in (Gawer, 2014) and (McIntyre & Srinivasan, 2016), platform literature seems astride a market-centred literature on one side and a technology-centred one on the other. Market-centred literature emphasizes the search for new markets and the capture of the value they generate through a variety of strategies and tactics, technologies being given, while technology-centred literature looks at the impact of new technology on a given product and the design of platform products, markets being given. Both dimensions bring important insights separately but both fields are obviously interrelated: changes in the market will have repercussions on the technologies and vice versa. Both approaches are therefore insufficient per-se and a third way needs to be taken.

While (Baldwin & Woodard, 2008) argued for the representation of platforms using either Design Structure Matrices or Layer Maps, those tools prove inadequate to qualify the interrelations between markets and technologies. Symmetrically, economists' tools also prove unfit to integrate technological impact. It should be noted, however, that (Gawer, 2014) first attempted to unify the literature by adopting an organisational point of view: while the framework does integrate both streams of literature together, it seems to lose sight of both technology and markets, therefore making it unfit for our needs. As we are looking at market and technology interdependencies, the use of a matroids-based model could be justified (Le Masson, et al., 2016) but seems unnecessarily difficult to use when (Suh, 1990) provides us with the simple yet efficient framework of Axiomatic Design and its Design Matrices.

Axiomatic Design's Design Matrices are matrices whose columns represent functions and lines techniques. The theory was developed to provide designers with a framework indicating the best possible design, functional requirements being given, through the choice of various design parameters and techniques. Everything being equal from the strategy point of view, a market will be addressed by a given product if this product's techniques can fulfil the set of functions this market requires. The best possible design is then determined thanks to the Axiomatic Design's two axioms of independence and information. The axiom of independence states that every single point of the functional space should be accessible while the axiom of information requires that the access to this point, i.e. the setting of the various design parameters, should be as easy as possible. A first consequence of those axioms is that the best possible design is represented by a diagonal matrix, i.e. a fully uncoupled system. A second one is that it induces a hierarchy between techniques: so as to access a given point of the functional space, the best strategy is to first set the most generic technique, i.e. the one addressing the largest number of functions, and then move towards less and less specific ones. Therefore, the most generic technique is the one coordinating the ecosystem. Lastly, it should be noted that Axiomatic Design lets us describe platform stakeholders, through the discussion on technique ownership structure.

For all those reasons, Axiomatic Design seems especially adapted to solve our initial problem, i.e. understanding the role of new markets and technologies in platforms' rises and falls.

3. MODELLING PLATFORMS AND ITS STAKEHOLDERS IN AXIOMATIC DESIGN: FROM DESIGN STRUCTURE MATRICES TO DESIGN MATRICES

So as to explore technological and functional generativity impact on platforms in a systematic way, we first need to understand how platforms can be modelled within the framework of Axiomatic Design. We proceed in three steps. First, we exhibit a "good candidate", a Design Matrix (DM) whose main property is to lead us back to (Baldwin & Clark, 1994)'s Design Structure Matrix (DSM) through Dong & Whitney's transformation (Dong & Whitney, 2001), and prove that this candidate is the one that best meets Axiomatic Design's two axioms. This candidate matrix is further simplified so that our discussions can focus on the platform leader – complementor duality. Second, we interpret the representation. Third, we introduce a secondary model describing technique ownership to discuss the stability of perturbated states.

3.1. Modelling platforms thanks to Design Matrices

As described in (Baldwin & Clark, 1994), a platform DSM can be divided into three main components: design rules, complementors and test and integration techniques. It can be modelled as in Figure 1 below (left). We now argue that the DM that best represents platforms in Axiomatic Design is the DSM's transpose in which columns are not techniques anymore but functions. Applying Dong & Whitney's transformation to this DM leads us back to Baldwin & Clark's DSM, confirming that it is a good candidate to represent platforms in Axiomatic Design. The transpose of the DSM corresponds to the initial system of equations below, from which we isolate techniques. Reinjecting the second system into the first, we obtain the third in which functions have disappeared. The resulting system of equations corresponds exactly to Figure 1 (left) matrix, i.e. the DSM of a platform.

$$\begin{cases} FR_n = f((T_i)_{1\leq i\leq n}) \\ FR_j = f(T_j,\, T_1)\ \forall j < n \end{cases} \Rightarrow \begin{cases} T_n = f(FR_n, (T_i)_{1\leq i\leq n-1}) \\ T_j = f(FR_j,\, T_1)\ \forall\, 1 < j < n \\ T_1 = f(FR_1) \end{cases} \Rightarrow \begin{cases} T = f((T_j)_{1\leq j\leq n}) \\ T_j = f(T_j,\, T_1)\ \forall j < n \end{cases} \quad (1)$$

What would be other alternatives? Three options are available to modify this DM: (1) adding or (2) deleting a relation or (3) adding more functions. (1) increases coupling, therefore leading to a worse design per the axiom of information. (2) leads to the wrong DSM by Dong & Whitney's transformation. As for (3), it leads to a worse design by the axiom of independence. Hence this matrix is the best DM to represent platforms within Axiomatic Design. At this point, the matrix distinguishes between three types of techniques and functions: design rules-related, complementors-related and tests-and-integration-related. The distinction between design rules on one side and tests-and-integration on the other feels unnecessary to our further discussions, for both are usually directly controlled by the platform leader (Gawer & Cusumano, 2002) (Baldwin & Clark, 1994). We therefore put aside both tests-and-integration techniques and functions to focus our discussions on the more meaningful distinction between the platform leader on one side and the complementors on the other. The final DM that will be used for further discussions is therefore presented in Figure 1 (right) above.

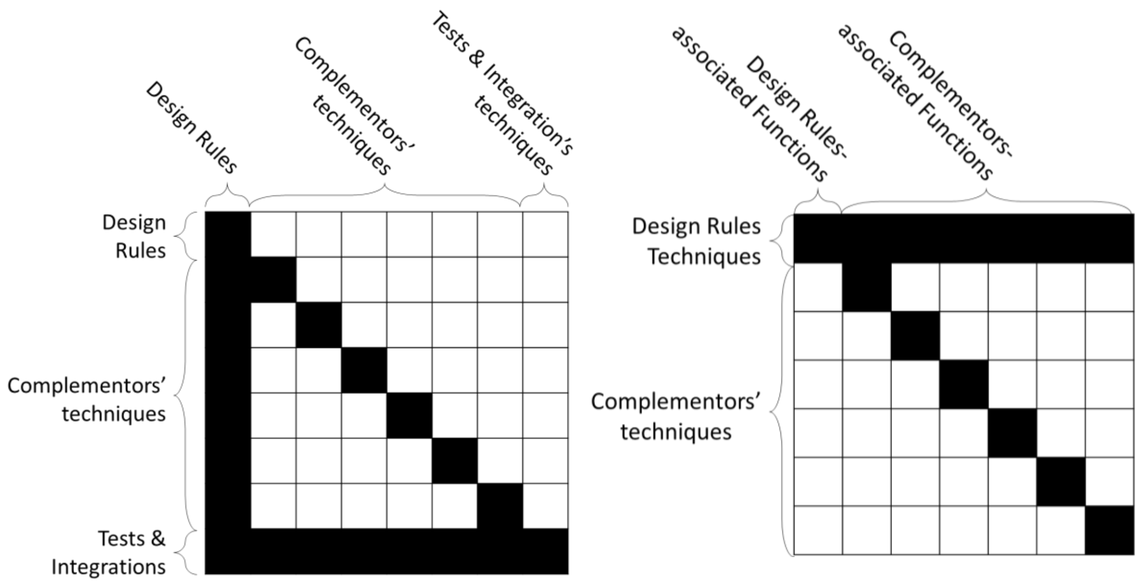

Figure 1. Design Structure Matrix (left) and simplified Design Matrix (right) of a Platform. Blackened boxes correspond to the existence of a relationship between the line and the column. White box, on the opposite, shows the absence of such relationship.

3.2. Interpreting the platform Design Matrix model: old and new

The transposition of platforms from a DSM-based description into a DM-based introduces new elements while conserving others. Concepts such as design rules and test & integration techniques, both supposed to be controlled by the platform leader, remain. Likewise, complementors' techniques are still represented in the DM.

Functions are a first novelty introduced, and three types can be distinguished: design rule-related, test & integration-related and complementors-related functions. This last type is the most straightforward

to understand: these are the functions and, behind it, the markets complementors specifically address and derive value from. The test and integration-related functions are those specifying how the whole system should be integrated and tested to make sure the platform core and its complementors work well together. Lastly, design rules-related functions are those specifically performed by the platform leader: they can be linked to the specific markets it addresses alone or to the functions which, together, enable the platform leader to indeed be a platform upon which complementors will be able to build their own solutions.
Furthermore, the trigonal configuration of a platform DM has two consequences. First, the matrix is trigonal, not diagonal: a better configuration exists, the diagonal matrix, corresponding, depending on the techniques' ownership structure, to a fully integrated system owned by one company or a fully disintegrated system owned by many companies. Second, as detailed before, Axiomatic Design provides us with rules to know, in such a configuration, which technique should be first to speak, and how others should react. In our case, the platform leader should be the first to set its technique and every complementor will then individually adapt their own technique to address their specific functions.
The modelisation of platforms with Design Matrices therefore lets us observe two main characteristics: their functional space, i.e. the system performance, and the coordination capacity of stakeholders, i.e. who is leading and who is following.

3.3. Towards technical and functional generativity analysis: configurations stability

If so much energy was deployed to obtain the platform DM, it is because the Axiomatic Design theory offers an ideal frame to discuss the impact of new functions and techniques on the system. In the previous paragraphs, we saw how the Design Matrix let us observe the functional space addressed by the system as well as the hierarchy that exists between the platform techniques. Observing changes of those two dimensions will let us know how both kinds of generativity affect the platform. But not only does Axiomatic Design give us good observables, it also provides us with a tool to systematically explore the effects of technical and functional generativity on platforms. Introducing a new technique indeed equals to simply adding a new line to the matrix, while adding a new function equals to adding a new column. What is left to discuss then is the impact of the new interrelations between the new technique, the new function and the pre-existing system.
Obviously, such combinatorics can quickly grow out of hand without adding much value to the discussion: for such reasons, we will focus on a 2x2 matrix in which there is only a platform leader and a complementor. When analysing outcomes of the model, one should therefore keep in mind that, in all generality, more than one complementor, and therefore more than one complementor's technique and function, exists.
In this simplifying case, combinatorics lead to 25 distinct configurations to be discussed. We distinguish four types: technical generativity only (3 cases), functional generativity only (3 cases), cases of joint technical and functional generativity in which the new technique does not address the new function – those cases are combinations of the two precedent types of generativity – (3 cases) and joint technical and functional generativity in which the new technique addresses the new function (16 cases).
Among those 16 cases, most are represented by non-trigonal matrices, indicating that those configurations are not stable: couplings are inducing feedback loops that endanger the platform integrity (Suh, 1990). In such situations, it is the role of the platform leader (the former or the new one) to fix it: assuming in this paper that it has full authority over the complementors of its ecosystem, it will partly integrate their techniques, i.e. update its design rules, in order to get rid of those dependencies again. In algebraic language, this is equivalent to a Gaussian elimination. Furthermore, complementors are incited to comply, given that if they do not, those feedback loops make it much more difficult for them to develop themselves in the platform. After this last transformation, the 16 cases can be reduced to 4 final configurations: rise of a new, distinct, market, platform extension, split leadership and platform overthrow (see Figure 2 below). New functions without new techniques similarly lead to either a platform extension or a situation of split leadership. Lastly, new techniques without new functions lead to traditional forms of competition between complementors (Baldwin & Clark, 1994), platform leader and complementors, and between platform leaders and wannabes (Gawer & Cusumano, 2002), which we will not further discuss here.

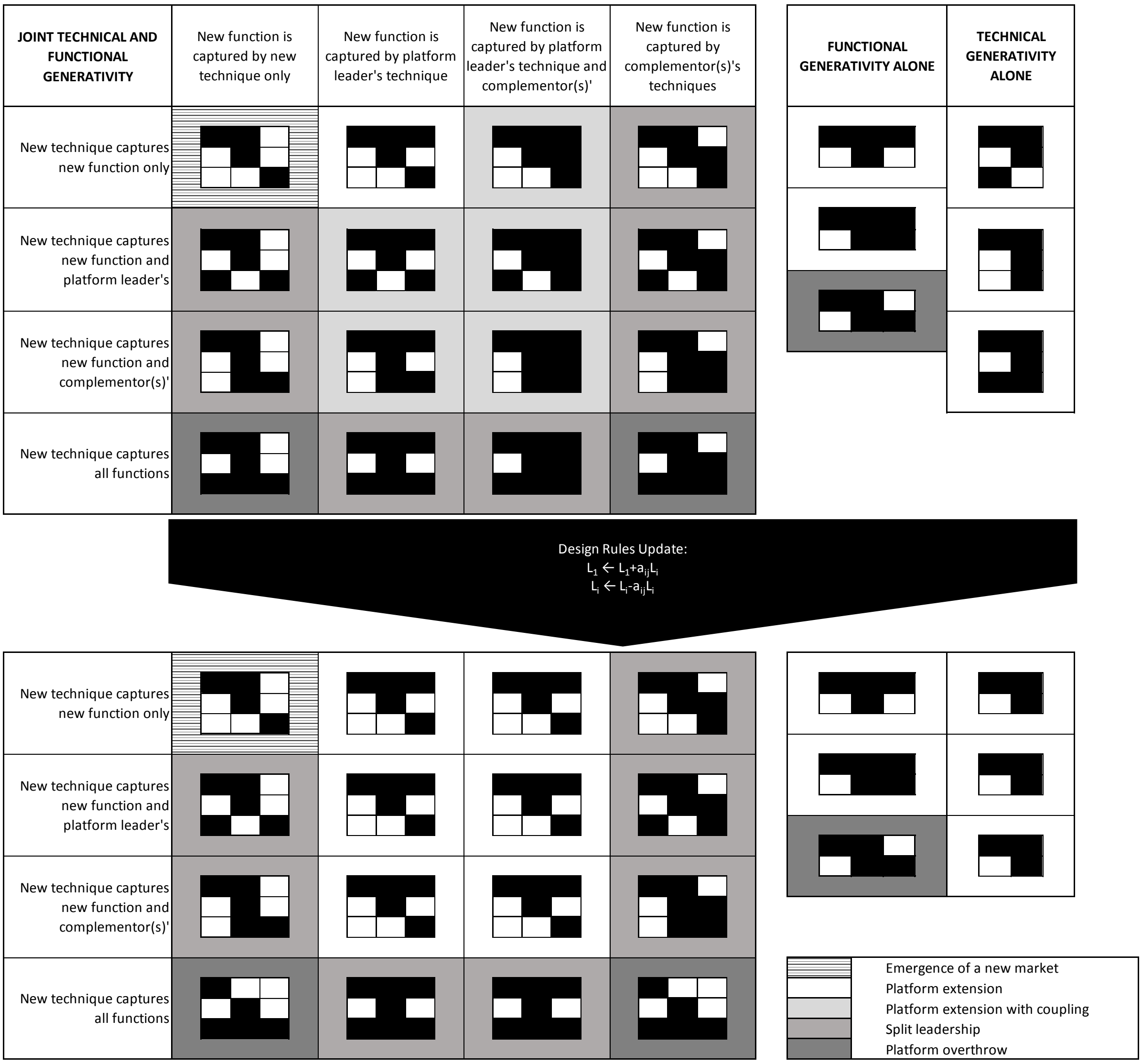

Figure 2. Potential outcomes of technical and functional generativity (up) and their further simplification once the platform leader's design rule is updated (below). Four categories of outcomes are distinguished: new, distinct, market, platform extension, split leadership and platform overthrow.

4. TECHNICAL AND FUNCTIONAL GENERATIVITY IMPACT ANALYSIS

At this point of our discussion, we have identified four main types of evolutions for platforms: the emergence of a new distinct market, platform extension, split leadership and platform overthrow. Here, we analyse those configurations by the two sus-mentioned observables, addressable functional space and coordination capacity, and single out the conditions that led to them in this fourth part.

4.1. The emergence of a new, distinct, market

In this configuration, the new technique only addresses the new function and the new function is only addressed by the new technique. While the overall system does cover more functions, there is no coordination between the pre-existing platform and the new component: it does not need to adapt to the platform leader's generic technique settings. There is therefore no connection between the pre-existing platform ecosystem and this new component, which leads us to conclude that this configuration represents the emergence of a new and distinct market. Further evolutions from that point are already widely addressed in the literature and we will not further discuss this case.

4.2. Platform extension

In this configuration, the system functional space dimension increases while its coordination remains ensured by the platform leader's generic technique. It proves generic enough to address the new function together with a new complementor. On the other hand, any other complementor's technique that would prove generic and therefore could also participate in the new function is stopped from doing so by the platform leader's design rules update mechanism that was describe in 3.3. This evolution is the most expected one: the platform evolves perfectly with its environment, ensuring that the new function is realized by both the platform leader and a complementor.

4.3. Split leadership

The situation gets a little trickier when the platform leader's generic technique is not generic enough while a complementor's technique, on the contrary, proves generic. The system functional space dimension increases but coordination capacity becomes more evenly distributed. On one hand, the platform leader's technique lack of genericity implies that it will less be able to control complementors addressing the new function: complementors may choose to set their techniques so as to address the new function rather than those within the platform. On the other hand, the complementor's technique genericity provides it with some coordination capacity. While this capacity generates troublesome feedback loops endangering the system integrity, the platform leader cannot solve it: the integration mechanism is partly altered, as complementors start to lose incentives to cede their technique to the integrator, which now means giving away a market they are in a good position to capture.

The more generic the complementor's technique, the more challenged the platform leader. The extreme case is when the new technique captures all the functions addressed by the platform leader: both actors then have a symmetrical position within the ecosystem and none really has leadership over the other. The leadership is split between the two actors who can, in turns, assume it.

4.4. Platform overthrow

Going one step further, the new technique can prove more generic than the platform leader's, i.e. it captures the new function as well as all the existing ones while the platform leader's does not capture the new function. In this case, the system functional space dimension has increased while the ecosystem coordination potential has changed hands: the new technique owner is now the one in charge. The platform leader got overthrown.

4.5. Remarks on the novelty of split leadership and platform overthrow

Let us insist on the novelty of the split leadership and platform overthrow outcomes: while most scholars were studying the direct competition between two platform leaders for a given market, or between a platform leader and a wannabee, the dynamics described here is subtler. There is no direct competition between two products involved but rather from within a single product, within the same platform. In one case, two players can end up sharing the platform leadership while, in the other, the leadership changes hand, turning the once platform leader into a complementor of the new one. And both evolutions only need two conditions to be triggered: functional generativity and technique genericity.

5. CASE STUDIES OF SPLIT LEADERSHIP AND PLATFORM OVERTHROW

Previous paragraphs identified four potential platform evolutions, among which two had already been noticed by scholars, two had not. This fifth part provides concrete cases in which those surprising behaviours have indeed taken place.

5.1. Cases of platform overthrow in the industry

Here focusing on the most surprising one, the platform overthrow, we analysed well-known cases in which platforms got challenged, looking for the emergence of a new function and a new technique and searching whether it could be said that the once complementor had become the new platform and the once platform had now become a complementor. 8 cases are listed below, half of which are successful platform overthrow, the other half corresponding to overthrow attempts that got stopped.
All those cases are following the same pattern. The complementor identifies new market needs, develops a technology to address it and this technology proves to be generic, leading to a power shift to its advantage. Other complementors then start to align themselves on the new emerging platform while abandoning the old one.

Table 1. Cases of platform overthrow

	IBM loss of leadership to "Wintel"		Nokia & the rise of smartphones	Yahoo! vs Google	JAVA & Microsoft	Netscape & Microsoft	Intel & Microsoft	Apple & Adobe
Initial Platform	IBM PC	IBM PC	Nokia	Yahoo!	Microsoft Windows	Microsoft Windows	Microsoft Windows	Apple iOS
Initial Complementor	Intel CPU	Microsoft OS	Symbian OS / Windows OS	Google	JAVA	Netscape Navigator	Intel	Adobe
Generic technique introduced	Bus PCI	MS/DOS	OS for mobiles	PageRank	Virtual Machine	Over-the-internet OS	NPS	Flash
Function the initial platform did not address	Let the CPU evolve at its own pace	Let software and hardware be independant	Let 3rd party applications blossom	Automatically indexing the Internet	Let developpers produce OS-independant software	Enable over-the-internet services to be developped	Let software developpers talk directly to the hardware	Let 3rd party developpers to distribute their apps without Apple control
Final Platform	Intel CPU's PCI bus	Microsoft MS/DOS	Windows OS	Google	Microsoft Windows	Microsoft Windows	Microsoft Windows	Apple iOS
Final Complementor	IBM PC	IBM PC	Nokia	Yahoo!	JAVA	Internet Explorer	Intel	HTML5 + CSS + JavaScript
Outcome Category	Platform overthrow	Platform overthrow	Platform overthrow	Platform overthrow	Countered platform overthrow (license purchased)	Countered platform overthrow (envelopment)	Countered platform overthrow (use of bargaining power)	Countered platform overthrow (censorship)
Sources	Gawer & Cusumano, 2002	Ferguson & Morris, 2003 ; Gawer & Cusumano, 2002	Gawer & Cusumano, 2014 ; Kenney & Pon, 2011	Rindova et al., 2012	Auletta, 2001	Auletta, 2001 ; Gawer & Cusumano, 2002 ; Yoffie & Cusumano, 1999	Gawer & Cusumano, 2002	Jobs, 2010

An emblematic case is IBM loss of leadership over the PC industry to the benefit of the "Wintel" alliance as described in (Gawer & Cusumano, 2002), (Baldwin & Clark, 2003) and (Ferguson & Morris, 1993). Intel developed a new bus technology as it needed to free itself from the PC architecture that was evolving too slowly. Intel felt that enabling independence between hardware components would be key to let everyone evolve at their own pace, and the PCI bus was instrumental in doing so. Soon enough, the whole industry would follow Intel's leadership in orchestrating the hardware-side of computers. Microsoft Windows had a similar history on the software side. It identified that software needed less dependence on hardware and therefore developed its operating system to meet this new need. Software developers would quickly switch from developing PC-compatible to Windows-compatible solutions. Nokia's fate facing the rise of smartphones and their operating systems was similar to IBM's. As (Gawer & Cusumano, 2014) and (Kenney & Pon, 2011) describe, Nokia did not manage to address the set of new functions emerging with smartphone-related technology, and did not foresee the central role of the OS to attract third-party software developers. Therefore, OS providers such as Microsoft and, more recently, Android managed to establish themselves as the new platform, Nokia being only one of their many complementors. Lastly, Yahoo! failed to recognize the importance of keyword-based search engines and the functions they enabled, letting Google, once a small part of Yahoo!'s portal, become the platform Yahoo! is now a component of.

On the other hand, JAVA and Netscape are well-known cases of new technologies that were perceived as threatening by Microsoft and therefore led it to design a specific counter-attack. We argue that one of the reasons why Microsoft was right to think so is because they were answering emerging needs, cross-OS software development for JAVA and software-over-the-Internet for Netscape, thanks to new technologies, JAVA's virtual machine and Netscape's Navigator. Those had the potential to turn Microsoft Windows into a complementor of their own platform (Auletta, 2001) (Gawer & Cusumano, 2002) (Yoffie & Cusumano, 1999). Similarly, Intel tried to push the NPS technology to bypass Microsoft, letting software developers to work directly on hardware pieces, therefore threatening Microsoft Windows leadership. Microsoft forced PC manufacturers not to use Intel technology, consequently avoiding the platform overthrow (Gawer & Cusumano, 2002). Last but not least, Adobe Flash had the potential to challenge if not overthrow Apple iOS leadership position in the distribution of apps: if the Flash technology were to be accepted in the iOS platform, app developers would not need to get Apple approval anymore to distribute their products. Apple consequently blocked Flash from working on its iPhone (Jobs, 2010).

These few examples prove that a similar pattern is at stake behind platform overthrows, and that our model enables us to better understand it.

5.2. Cases of split leadership

Situations of split leadership, in which two platform leaders coexist at the same time, are another singular prediction of our model. We believe they were never identified before, yet they encompass a very interesting power equilibrium between multiple leaders of a single platform. They happen when the initial platform leader's technique and the new one control the same number of function, while not the same ones: none of the two can take the lead over the other, none of the two can integrate the other. We believe that cases of split leadership are especially common in products that combine both hardware and software, in which a hardware platform and a software platform coexists together without being fully able to establish their leadership upon one another. This is best observable in the case of the "Wintel" alliance. While Microsoft and Intel both overthrew IBM's leadership in their respective functional spaces – software for Microsoft and hardware for Intel – as described above, they still had to cooperate on a number of other functions and turn by turn assumed leadership on certain decisions while simply aligning on others (Casadesus-Masanell & Yoffie, 2007).

6. CONCLUSION AND DISCUSSION

This paper introduced a new model based on Axiomatic Design with the aim of describing platforms' evolution over time by bridging the literature on technology management / engineering design and economy / strategy. It led us to systematically analyse potential outcomes of such evolution, some of them turning out to be well-known evolutions (e.g. competition between platform leader and complementors, competition between complementors), some unknown (e.g. platform overthrow and split leadership). We then provided concrete examples in which platforms evolved to see their leader overthrown or to end up in a split leadership configuration.

Our model proves that two factors, technique genericity and functional generativity, are the source of challenges met by both platform leaders and their complementors. Can those two factors be managed? We believe so, and this would lead to identify a third role to be filled in platform ecosystems: while complementors address niche markets based on the platform leader's core technology, this third stakeholder would be the one setting functional generativity intensity – from none, so as to fully preserve the current ecosystem, to high, letting the ecosystem reorganize itself – and technique genericity – forcing other stakeholders to only introduce low-genericity techniques to preserve the ecosystem or letting highly generic techniques enter the ecosystem to let it reorganize. We call this new role a hyper-hub, for it is an underlying architectural component of platforms ("hub") and has the capacity of deeply disrupt the way platforms based on its foundations work ("hyper").

This paper therefore opens two new avenues for further research. The first one is to continue exploring what the model tells us on platforms' dynamics. The variety of outcomes is high and we did not detail all of them, so some may hide more unexpected behaviours. The integration mechanism described in 3.3 could also use further investigation. The two concepts of platform overthrow and split leadership deserve deeper study. The second one would be to further detail the concept of hyper-hub, analysing cases of platforms and identifying who is filling this role and how it is done.

REFERENCES

Armstrong, M., 2006. Competition in two-sided markets. *RAND Journal of Economics,* 37(3), pp. 668-691.

Auletta, K., 2001. *World War 3.0: Microsoft and its enemies.* New York: Random House.

Baldwin, C. & Clark, K., 1994. Modularity-in-Design: An Analysis Based on the Theory of Real Options.

Baldwin, C. & Woodard, J., 2008. *The Architecture of Platforms: A Unified View,* s.l.: Harvard Business School.

Baldwin, C. Y. & Clark, K. B., 2003. Managing in an age of modularity. In: R. Garud, A. Kumaraswamy & R. N. Langlois, eds. *Managing In The Modular Age: Architectures, Networks, and Organizations.* s.l.:Blackwell Publishing, pp. 150-172.

Bresnahan, T. F. & Greenstein, S., 1999. Technological Competition and the Structure of the Computer Industry. *The Journal of Industrial Economics,* XLVII(1).

Casadesus-Masanell, R. & Yoffie, D. B., 2007. Wintel: Cooperation and Conflict. *Management Science,* pp. 584-598.

Dong, Q. & Daniel, W., 2001. *Designing a Requirement Driven Product Development Process,* Pittsburgh, PA: Massachussetts Institute of Technology Engineering Systems Division.

Dong, Q. & Whitney, D. E., 2001. *Designing a Requirement Driven Product Development Process,* Pittsburgh, PA: Massachussetts Institute of Technology Engineering Systems Division.

Eisenmann, T., Parker, G. & Van Alstyne, M., 2011. Platform Envelopment. *Strategic Management Journal,* Volume 32, pp. 1270-1285.

Eppinger, S. D., 2001. Innovation at the Speed of Information. *Harvard Business Review.*

Ferguson, C. & Morris, C., 1993. *Computer Wars: The Fall of IBM and the Future of Global Technology.* Washington, D.C.: Beard Books.

Gawer, A., 2014. Bridging differing perspectives on technological platforms: Towards an integrative framework. *Research Policy,* 43(7), pp. 1239-1249.

Gawer, A. & Cusumano, M. A., 2002. *Platform leadership.* Boston: Harvard Business School Press.

Gawer, A. & Cusumano, M. A., 2014. Industry Platforms and Ecosystem Innovation. *Journal of Product Development & Management Association,* 31(3), pp. 417-433.

Hatchuel, A., Le Masson, P. & Weil, B., 2009. Platforms for the design of platforms: collaborating in the unknown. In: *Platforms, Markets and Innovation.* s.l.:s.n.

Henderson & Clark, 1990. Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established firms.

Henderson, R. M. & Clark, K. B., 1990. Architectural Innovation: The Reconfiguration of the Existing. *Administrative Science Qarterly,* 35(1), pp. 9-30.

Iansiti, M. & Levien, R., 2004. Strategy as Ecology. *Harvard Business Review.*

Jacobides, M. G., Knudsen, T. & Augier, M., 2006. Benefiting from innovation: Value creation, value appropriation and the role of industry architectures. *Research Policy,* 35(8), pp. 1200-1221.

Jobs, S., 2010. *Apple.com.* [Online]
Available at: http://www.apple.com/hotnews/thoughts-on-flash/
[Accessed 02 12 2016].

Kenney, M. & Pon, B., 2011. Structuring the Smartphone Industry: Is the Mobile Internet OS Platform the Key?. *Journal of Industry, Competition and Trade,* 11(3), pp. 239-261.

Le Masson, P., Hatchuel, A., Kokshagina, O. & Weil, B., 2016. Designing techniques for systemic impact: lessons from C-K theory and matroid structures. *Research in Engineering Design,* pp. 1-24.

McIntyre, D. P. & Srinivasan, A., 2016. Netwoks, platforms, and strategy: emerging views and next steps. *Strategic Management Journal.*

Meyer, M., 1997. Revitalize your product lines through continuous platform renewal. *Research Technology Management,* 40(2), pp. 17-28.

Rochet, J.-C. & Tirole, J., 2003. Platform Competition in Two-Sided Markets.

Suh, N. P., 1990. *The Principles of Design.* New York: Oxford University Press.

Tee, R. & Gawer, A., 2009. Industry architecture as a determinant of successful platform strategies: a case study of the i-mode mobile Internet service. *European Management Review,* Volume 6, pp. 217-232.

Tirole, J. & Rochet, J.-C., 2003. Platform competition in two-sided markets. *Journal of the European Economic Association*, 1(4), pp. 990-1029.
Yoffie, D. B. & Cusumano, M. A., 1999. Building a Company on Internet Time: Lessons from Netscape. *California Management Review*, 41(3), pp. 8-28.
Yoffie, D. B. & Kwak, M., 2006. With Friends Like These: The Art of Managing Complementors. *Harvard Business Review*.