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

Legrand, Julien; Thomas, Maxime; Le Masson, Pascal; Weil, Benoît
MINES ParisTech, France

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 challenge the platform leader. Both can be linked to two necessary conditions: functional generativity and technical genericity. We then identify those behaviours in several cases.

Keywords: Platform strategies, Design engineering, Market implications, Design theory, Innovation

Contact:
Julien Legrand
MINES ParisTech
CGS
France
jrr.legrand@gmail.com

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 7: Design Theory and Research Methodology, Vancouver, Canada, 21.-25.08.2017.

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 and Cusumano 2002) or dividing tasks while mitigating risks associated to their development (Baldwin and Clark 1994).

While the description of a given platform at a given time now seems to be well established since (Baldwin and Woodard 2008), several scholars have stressed that today's challenge is now to understand the laws governing their evolution in time (Gawer 2014, Gawer and Cusumano 2014, McIntyre and 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, a new light is shed on platform stakeholders' behaviour when they face the emergence of a new technology or a new market. While the model includes behaviours, scholars are already familiar with, it also leads 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. It is also found that the apparition of a new generic technique, i.e. the capacity of a technique to fulfil several functions, 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, a "hyper-hub": its main characteristics will be outlined at the end of this paper.

Following this introduction, the discussion will be divided into five parts. The first will position the discussion within the broader platform literature, identifying the key problems it addresses and explaining the reasons why Axiomatic Design theory is especially fit 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 Design Structure Matrices (DSM) and Axiomatic Design's Design Matrices (DM). The third paragraph will analyse the impact of technical and functional generativity, i.e. the emergence of new techniques and functions, 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 phenomena in the industry. Finally, the fifth and last part concludes on this paper's findings, reminds the limitations of the chosen 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 by various scholars: Baldwin & Woodard (2008), Gawer (2014) and McIntyre and Srinivisan (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 and Cusumano 2002) as it organizes the platform ecosystem – as described by Iansiti and Levien (2004) – along with a set of design rules(Baldwin and Clark 1994). The other components were identified, in turns, as either complementors or modules, depending on the scholar's tradition: respectively, strategy / management (Yoffie and Kwak 2006) or technology management / engineering design (Baldwin and Clark 2003). The terms are covering a similar reality, at least within the platform literature: technical solutions addressing niche market needs built upon the common foundations established by the platform core. Beside those two types of components come two other concepts. First, network effects, as studied by Tirole and Rochet (2003), describe the self-reinforcing effect of platforms' sides on one another, usually users and complementors. Second, Tee and Gawer (2009) proved that industrial architecture (Jacobides *et al.* 2006) played a critical 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 and Cusumano 2014, McIntyre and Srinivasan 2016). Scholars from heterogeneous fields have contributed to describe specific platform behaviours. The construction of a platform leadership was studied by Gawer and Cusumano (2002) who identified four levers that industry players should consider when designing their platform strategy. Henderson and Clark (1990) introduced the concept of architectural innovation, which happens at the interfaces, in between platforms' core and complementors. Bresnahan and Greenstein (1999) and Armstrong (2006) studied the direct competition between platforms, while Yoffie and 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, Parker and Van Alstyne (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 by Gawer (2014) and McIntyre and 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 generated value 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 and Woodard (2008) argued for the representation of platforms using either DSM 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 first attempted to unify the literature by adopting an organisational point of view (Gawer 2014): while her framework does integrate both streams of literature together, it seems to lose predictive power to the benefit of a holistic, static, description, therefore making it unfit to study platforms dynamics. As this paper looks 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 the simple yet efficient framework of Axiomatic Design and its DM.

Axiomatic Design's DM 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 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 by Axiomatic Design's two axioms of independence and information. The first one states that every single point of the functional space should be accessible, i.e. all functions should be fulfilled by at least a technique, while the second one requires that the "path" to this point, i.e. the setting of the various techniques to fulfil a function, 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 highest number of functions, and then move towards more and more specific ones. Therefore, the most generic technique is the one coordinating the system. Lastly, it should be noted that Axiomatic Design describes how platform stakeholders interact, provided that a discussion on technique ownership structure is included. For all those reasons, Axiomatic Design seems especially fit to solve the problem at hand, 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

The Axiomatic Design Theory, however, does not provide a description of platforms. The first effort of this paper, in consequence, will be to model them with Suh's matrices. A three-steps approach is adopted. First, the DM associated to Baldwin and Clark's (1994) DSM is exhibited and simplified so that further discussions can focus on the platform leader – complementor duality. Second, an interpretation of the representation is provided. Third, a secondary model describing technique ownership structure is introduced to discuss the stability of perturbed states.

3.1 Modelling platforms thanks to Design Matrices

As demonstrated by Baldwin and 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). The following proves that one DM representing platforms in Axiomatic Design is the DSM's transpose in which columns are not techniques anymore but functions. Indeed, applying Dong & Whitney's (2001) transformation to this DM leads back to Baldwin & Clark's DSM. Noting T_i the techniques, F_i the functions and f the applications linking them, the transpose of the DSM corresponds to the initial system of equations below. After isolating the techniques and re-injecting them into the first system of equations, the third is obtained in which functions have disappeared. The resulting system of equations corresponds exactly to Figure 1 (left) matrix, i.e. the DSM of a platform.

$$\begin{cases} F_n = f((T_i)_{1\leq i\leq n}) \\ F_j = f(\,T_j,\,T_1)\ \forall j < n \end{cases} \Rightarrow \begin{cases} T_n = f(F_n, (T_i)_{1\leq i\leq n-1}) \\ T_j = f(\,F_j,\,T_1)\ \forall\ 1 < j < n \\ T_1 = f(F_1) \end{cases} \Rightarrow \begin{cases} T = f\big((T_j)_{1\leq j\leq n}\big) \\ T_j = f(\,T_j,\,T_1)\ \forall j < n \end{cases} \quad (1)$$

What would other alternatives be? 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.

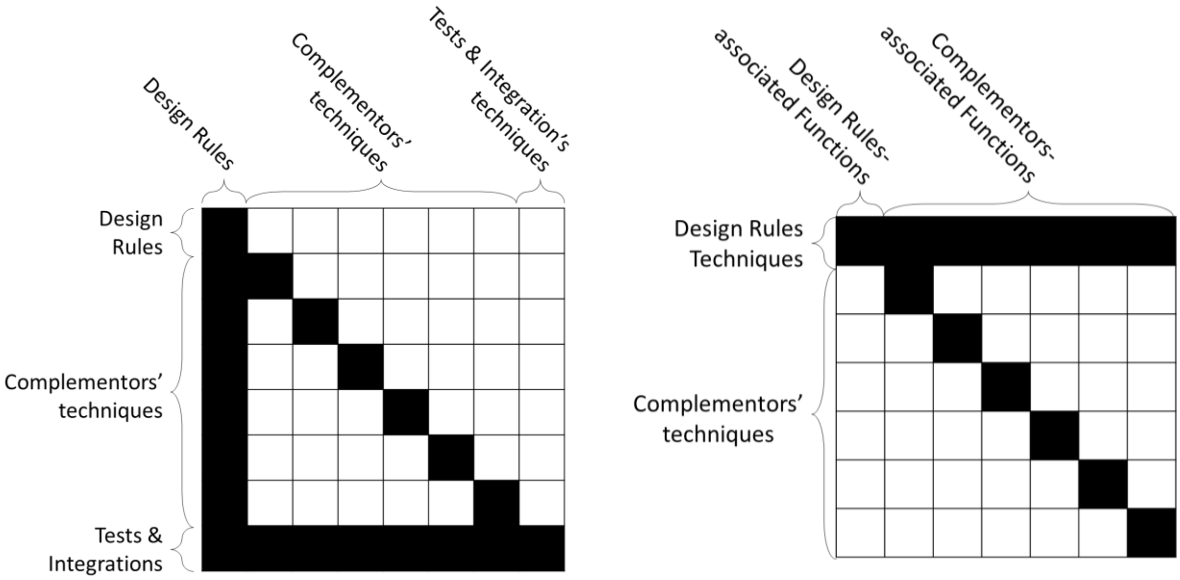

Figure 1. Design Structure Matrix (left) and simplified Design Matrix (right) of a Platform. Blackened boxes correspond to the existence of a relationship between the line and the column. White box, on the opposite, shows the absence of such relationship.

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 further discussions, for both are usually directly controlled by the platform leader (Baldwin and Clark 1994, Gawer and Cusumano 2002). Tests-and-integration techniques and functions can therefore be put aside to focus the 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) below.

3.2 Interpreting the platform Design Matrix model: old and new

Modelling platforms with DM preserve classical concepts while introducing new ones. Concepts such as design rules and complementors' techniques remain. Functions are a first novelty introduced, and two types can be distinguished in this simplified model: design rules-related and complementors-related functions. This second type is the most straightforward to understand: these are the functions and, behind it, the markets complementors specifically address and derive value from. 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 structure of a platform DM has two consequences. First, the matrix is trigonal, not diagonal: the trigonal configuration is stable according to Axiomatic Design Theory but 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 disintegrated system owned by many companies. It seems intuitive that platforms are indeed systems whose design could be simplified, even though their complexity is probably one of the sources of their robustness to change. Second, as detailed in part 2, Axiomatic Design provides rules to know, in such a configuration, which technique should be first to speak, and how others should react. In the case at hand, 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 DM-based representation of platforms therefore provides a useful tool to monitor the evolution of platforms according to two dimensions: their functional space, i.e. the number of function the system addresses, and the coordination capacity of stakeholders, i.e. who is leading and who is following in the process of fulfilling those functions.

3.3 Towards technical and functional generativity analysis: configurations stability

If so much energy got 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. The previous paragraphs explained how the DM describes the functional space addressed by the system as well as the hierarchy that exists between the platform techniques. Monitoring the evolution of those two dimensions provides insights on how both kinds of generativity affect the platform. Furthermore, Axiomatic Design also provides a model to understand the impact of new technologies and markets on platforms in a systematic way. Introducing a new technology indeed equals to simply adding new lines – i.e. a set of techniques – to the matrix, while addressing a new market equals to adding new columns – i.e. a set of functions – (cf upper table of Figure 2 below). 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.

Such a discussion, however, implies combinatorics that can quickly grow out of hand without adding much value: for such reasons, the following discussions will be based on a 2x2 matrix in which there is only a platform leader and a single complementor. When analysing outcomes of the model, one should therefore keep in mind that, in all generality, there are several complementors all of which controlling several techniques fulfilling several functions.

In this simplifying case, combinatorics already leads to 25 configurations. Four categories of perturbed states can be distinguished: pure technical generativity (3 cases), pure functional generativity (3 cases), cases of joint technical and functional generativity in which the new technique does not address the new function – i.e. a superposition of the two previous categories – (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 to fix it: assuming in this paper that it has full authority over the system's complementors, it will partly integrate their techniques, i.e. update its design rules, in order to get rid of the loops. For the matrix itself, this is equivalent to a Gaussian elimination taking place in the DM. Furthermore, complementors are incited to comply, given that those feedback loops make it much more difficult for them to develop themselves in the platform. It should be noted here that this assumption is a strong one – the platform leader is assumed to be almighty – and enables to set aside discussions on complementors' willingness to comply with the platform leader. The effects highlighted in this paper are therefore tightly linked to the way sets of techniques fulfil sets of functions, and not related to relative power of stakeholders.

After taking into account the effects of 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).

The 9 remaining configurations can similarly be processed and lead to situations of platform extension or split leadership.

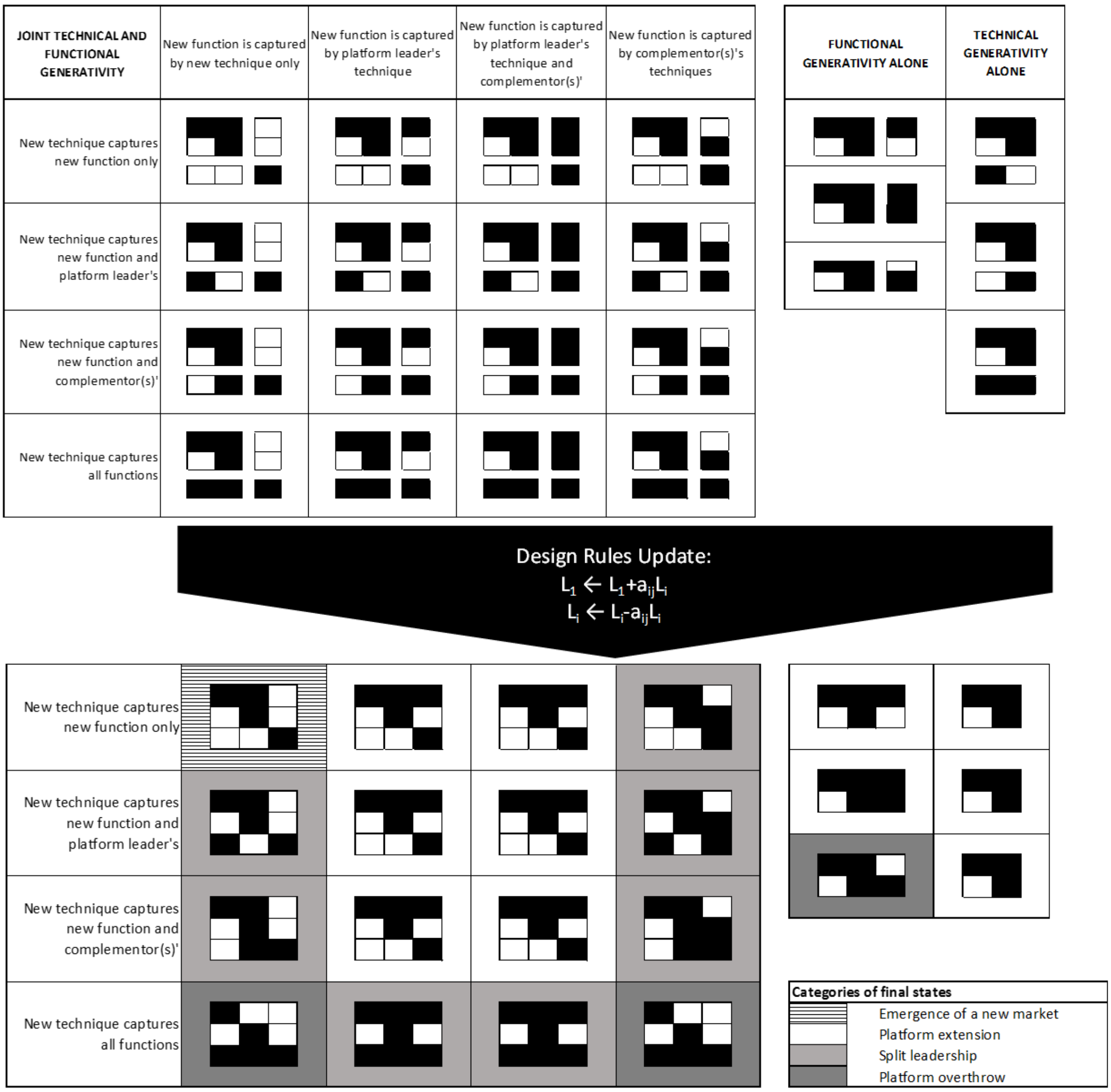

Figure 2. Perturbed states resulting from technical and functional generativity (up) and the associated final state once the platform leader's design rule has been updated (below). Four categories of outcomes are distinguished. The mathematical operation corresponds to a Gaussian elimination: L_i stands for lines and a_{ij} stands for coefficients of the matrix.

4 TECHNICAL AND FUNCTIONAL GENERATIVITY IMPACT ON THE PLATFORM EQUILIBRIUM

At this point of the discussion, four main types of evolutions for platforms have been identified: the emergence of a new distinct market, platform extension, split leadership and platform overthrow. Here, those configurations are analysed by the two above-mentioned observables, addressable functional space and coordination capacity, in order to single out the conditions under which they were made possible.

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. This configuration, therefore, represents indeed the emergence of a new and distinct market. Further evolutions from that point are already widely addressed in the literature and this case will therefore not be further discussed here.

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 described 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 adapt 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 Conclusions on the model predictions

The model, introduced to study in a systematic way how platforms could evolve in time, therefore only permits four types of outcomes: the emergence of a new, distinct, market, platform extension, split leadership and platform overthrow. If the two first were expected, the two other are much more surprising. It should be noted that the dynamics behind split leadership and platform overthrow do not involve direct competition between two products but rather competition 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. Both evolutions only need two conditions to be triggered: functional generativity, i.e. the emergence of a new function, and technique genericity, i.e. the capacity of a technique to fulfil several functions.

5 EMPIRICAL TEST 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 examples in which those surprising behaviours have indeed taken place. Those are well-documented, almost "classical" ones. The authors here focus on the hardware & software industries as (1) they believe its architecture enables a higher pace of change and (2) they were familiar with the field.

5.1 Cases of platform overthrow in the industry

Here focusing on the most surprising one, the platform overthrow, well-known cases in which platforms got challenged are analysed, looking for the emergence of a new function and a new technique and searching whether it can be said that the once complementor has become the new platform and the once platform has now become a complementor. 8 cases are listed below in Table 1, half of which are successful platform overthrow, the other half corresponding to countered overthrows.

All those cases, be they successful or not, follow 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 potential power shift to its advantage. Other complementors then start to align themselves on the new emerging platform while abandoning the old one, if this one does not counter-attack.

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 by Gawer and Cusumano (2002), Baldwin and Clark (2003) and Ferguson and Morris (1993). Intel developed a new bus technology as it needed to free itself from the PC architecture that was to slow to evolve. 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 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 identical to IBM's. As described by Gawer and Cusumano (2014) and Kenney and Pon (2011), Nokia did not manage to address the set of 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 becoming 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 to their overthrow attempts. Both companies 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 (Yoffie and Cusumano 1999, Auletta 2001, Gawer and Cusumano 2002). Both attempts failed for different reasons: Windows forced JAVA to become a complementor through licensing and squeezed out Nestcape by bundling Internet Explorer to Windows.

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 and Cusumano 2002).

Adobe Flash also had the potential to challenge if not overthrow Apple iOS leadership in the app distribution: if its technology were to be accepted by Apple, developers would not need to get Apple approval anymore to distribute their apps. Apple consequently blocked Flash from working on its iPhone.

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 the model. To the best of the authors' knowledge, 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 functions, but not the same ones: none of the two can take the lead over the other.

Such cases of split leadership are especially common in products that combine both hardware and software, in which a hardware platform and a software platform coexist 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 several other functions and turn by turn assumed leadership on certain decisions while simply aligning on others (Casadesus-Masanell and Yoffie 2007).

6 CONCLUSION AND DISCUSSION

This paper introduced a new model based on Axiomatic Design Theory with the aim of describing platforms' evolution over time while bridging the literature on technology management / engineering design and economy / strategy. It led to systematically analyse potential outcomes of such evolution, some of them turning out to be well-known evolutions – classical competition, apparition of new markets or platform extensions –, some unknown – platform overthrow and split leadership –. Concrete examples in which platforms evolved to see their leader overthrown or ended up in a split leadership configuration were then provided.

Our model also proves that two factors, functional generativity and technique genericity, are the sources of challenges met by both platform leaders and their complementors. Can those two factors be managed? The authors of this paper 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. This new role should be called 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 main new avenues for further research. The first one is to continue exploring what the DM model tells on platforms' dynamics. The variety of outcomes is high and they could not all be described in details, 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 also deserve deeper study. The second one would be to further detail the concept of hyper-hub, analysing cases of platforms and identifying the stakeholders filling this role and the levers they use to perform it.

REFERENCES

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

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

Baldwin, C. and Clark, K. (1994), “Modularity-in-Design: An Analysis Based on the Theory of Real Options”.

Baldwin, C. and Woodard, J. (2008), *The Architecture of Platforms: A Unified View*Harvard Business School.

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

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

Casadesus-Masanell, R. and Yoffie, D. B. (2007), “Wintel: Cooperation and Conflict”, *Management Science*, 584-598.

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

Eisenmann, T., Parker, G. and Van Alstyne, M. (2011), “Platform Envelopment”, *Strategic Management Journal*, 32, 1270-1285.

Ferguson, C. and 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), 1239-1249.

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

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

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

Iansiti, M. and Levien, R. (2004), “Strategy as Ecology”, *Harvard Business Review*.

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

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

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

McIntyre, D. P. and 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), 17-28.

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

Tee, R. and 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*, 6, 217-232.

Tirole, J. and Rochet, J.-C. (2003), “Platform competition in two-sided markets”, *Journal of the European Economic Association*, 1(4), 990-1029.

Yoffie, D. B. and Cusumano, M. A. (1999), “Building a Company on Internet Time: Lessons from Netscape”, *California Management Review*, 41(3), 8-28.

Yoffie, D. B. and Kwak, M. (2006), “With Friends Like These: The Art of Managing Complementors”, *Harvard Business Review*.