Proceedings Paper:

Ozalp, H orcid.org/0000-0003-4811-6072 and Cennamo, C (2017) Platform Architecture, Multihoming and Complement Quality. In: Academy of Management Proceedings. Academy of Management, 04-08 Aug 2017, Atlanta, Georgia, USA. Academy of Management , p. 12276.

https://doi.org/10.5465/AMBPP.2017.228

PLATFORM ARCHITECTURE, MULTIHOMING AND COMPLEMENT QUALITY: EVIDENCE FROM THE U.S. VIDEO GAME INDUSTRY

HAKAN OZALP
Leeds University Business School
University of Leeds, Leeds LS6 1AN

CARMELO CENNAMO
Bocconi University

INTRODUCTION

The early literature on platforms has emphasized the role of indirect network effects in adoption decisions of independent complement providers and users (Hagiu and Wright 2015, Parker and Van Alstyne 2005, Rochet and Tirole 2006, Rysman 2009). The defining feature of a platform in this literature is that it facilitates transactions between complement providers on one side (sellers) and consumers on the other side (buyers), with each side of the platform benefiting more from the platform the higher the number of users on the other side. Thus, a complement provider wants to reach as many consumers as possible, on the same platform or even across multiple platforms. Indeed, several studies highlight a recent trend of complement providers to develop their products for multiple competing platforms, to "multihome" (Bresnahan et al. 2015, Corts and Lederman 2009, Landsman and Stremersch 2011). Most studies on multihoming make the point that the cost of multihoming (developing the product for one additional platform) declined over the past years, not least due to the emergence of "middleware" application tools that make the writing of software code less specific to a given platform technology (Corts and Lederman 2009).[1] With the cost of multihoming supposedly minimal compared to the high fixed development costs of the product, and with greater expected benefits from targeting users on multiple platforms instead of just one, multihoming has become the preferred strategy for the majority of complement providers in many markets. Consequently, platform specific factors such as the design and core technology are expected to play a negligible role for complementary product innovation as most complements are developed for multiple platforms anyway. Has technology in platform systems been modularized "to oblivion" so that technological quality and architecture are not decisive anymore in attracting high-quality innovation in complements? To tackle this general question, we focus on whether and how platform architecture impacts the innovation performance of multihoming complements. In essence, we ask: Do complements multihome better onto some platforms than others? If so, how does platform architecture affect this?

HYPOTHESES

Platform owners face a tradeoff between modular platform architecture, which minimizes interdependencies among modules through interfaces (Simon 1962, Baldwin and Clark 2000) and greater co-specialization (by complement providers) into the platform system, which requires higher integration between complements and platform architecture (Tiwana et al. 2010). Complement providers willing to make greater investments specific to the platform system may eventually achieve greater innovation performance for their products, which benefits the whole system (Claussen et al. 2015b). This creates benefits beyond what would be possible in a more generic, modular system (Schilling 2000). However, in such a tightly integrated system, each component is highly specific to the system, and using non-specific components entails a loss of performance (Simon 1962, Schilling 2000). Multihoming complements clearly face this tradeoff as they must compromise between platform-specific investments that optimize the product for each platform but duplicate investments, and cross-platform optimization. We thus expect:

> H1: The quality of multihoming complements is lower on more complex platforms.

Simultaneous vs. Staggered Releases

A simultaneous product release strategy follows a less platform specific product development approach: the product is designed to be compatible up front with every platform it is released on. Because perfect modularity across distinct platforms is hard to achieve, the key challenge for firms with this strategy is keeping the compatibility costs of product development at a minimum while designing the product to work on multiple platforms (Lengyel 2000). Since the complement provider is focused on developing a modular product with minimal platform specific investments, the complement may perform worse on complex platforms following the logic of our first hypothesis. We thus expect the negative link between complement quality and platform complexity to be stronger for simultaneously released complements than for staggered ones, and to be weaker for staggered releases the longer the timespan between releases.

> H2: The negative relationship between platform complexity and the quality of multihoming complements weakens with the time since the first release of the complement.

Vertical Integration by Complement Providers

The need for vertical integration extends to complement providers for a complex platform. Greater platform complexity entails an increase in asset specificity (Williamson 1985) of the complement development process for the platform. This requires greater coordination between platform owner and complement provider. Greater coordination is also required between developers and publishers of the complement. Under a contracted relationship, both the developer and the publisher cannot fully anticipate the possible development issues that may unfold during the process and will organize (and contract) production and marketing of the complement according to the average time and effort required.

> H3: The negative relationship between platform complexity and the quality of multihoming complements is weaker for complements developed by a vertically integrated provider.

Development Tools in the Production of Multihoming Complements

Our logic assumes that technological complexity cannot be easily reduced by complement providers unless they make specific learning investments into it. In fact, complexity within a platform's subsystems can be shielded with a simple interface that hides the details of the interactions within the subsystems (Gawer 2014). An example is the graphical user interface (GUI) of the Windows, where every click by the user translates multiple and complex commands to the computer. Similarly, in the development of multihoming complements, development tools are available to reduce the complexities of each platform. For example, in the video game industry, middleware tools hide the inner workings of each platform and provide developers with a specific development framework that can translate the output to any platform-specific code. Therefore, we expect that complements using such development tools will be better able to integrate with complex platforms as part of the platform-specific integration task will be handled by these tools.

> H4: The negative relationship between platform complexity and the quality of multihoming complements is lower for complements using development tools.

DATA AND METHODS

We study multihoming games in the U.S. Video Game Industry from 1999 to 2010. The industry is a quintessential example of a platform technology, with strong indirect network effects and the key importance of complementary products (Clements and Ohashi 2005, Cennamo and Santalo 2013, Kretschmer and Claussen 2016). Our empirical setting provides a natural way of identifying our effect by looking at multihoming games. By using game fixed-effects, we can avoid any omitted variable bias arising from between-game differences and isolate game-level variance in quality from variance in quality performance across consoles due to game-console integration. We build our dataset from multiple sources. Our primary data source is the MobyGames website, the world's largest online video game archive on the Internet (Corts and Lederman 2009, Mollick 2012, Claussen et al. 2015a, Kretschmer and Claussen 2016). Our second data source is the GameRankings website. The site has review information for over 14,000 games, with over 300,000 individual game reviews. Our final dataset includes 1,483 sixth-generation console title-platform observations and 806 seventh-generation console title-platform observations, a total sample of 2,289 title-platform observations. We run our regressions for each console generation separately.

Platform owners carefully design their hardware as it affects a whole ecosystem of complement providers who need to work with it for the next six to seven years. There are three main components of platform design: CPU (including any co-processors), Graphics Processor (again, including co-processors), and RAM. Although other dimensions of performance matter (e.g., bus speed, cache memory, optical media, and so on), these three are by far the most important ones (and largely determine other architectural elements as well). When releasing new consoles, platform owners aim at high hardware power as this is the way to push cutting edge graphics games and the way to lure both users and developers alike (Kretschmer and Claussen 2016, Claussen et al. 2015, Zhu and Iansiti 2012). There is emerging theory (Anderson et al. 2014) and cases from the industry indicating that the advanced, but more complex, hardware creates difficulties in the production of complements.

Two related issues are emphasized for "complex" platforms in the industry. First, the number of processors in a console was highlighted as a main source of complexity (Kent 2001, Pettus 2013, Parish 2014).[2] The reason of this complexity is borne out of the need to manage an increasing number of interdependencies to utilize the architecture in an optimal way.[3] Second, the complexity is also attributed to the need to use platform-specific language to optimize (i.e., to integrate) the game code for the platform (Pettus 2013, Roth 2013), which is also known as the “assembly language”: An example is helpful to illustrate the sources of variation in console complexity. Playstation 2 and Playstation 3 are the most complex consoles for the sixth and seventh generation, respectively.

We measure the quality of games by using the average aggregate GameRankings score with the variable Title-Platform Quality. The quality score varies between 0 and 100, with the average quality score being close to 70. Games released on different platforms often receive different reviews, reflecting distinct game graphics or smoothness of gameplay between different platforms. Therefore, the quality of a specific version of a game on a particular platform gives information on how well a game has been implemented on a platform.

Complex Platform. We examined the effect of the complex platform architecture on the quality of complements (games) by using a dummy for those platforms. As summarized above, Playstation 2 and Playstation 3 are the complex platforms in the sixth and seventh generation, respectively.

ln(Delay in Days). This variable measures the logged number of days of delay for the focal platform release since the first release of the game for another platform. It is zero for simultaneous releases and the first platform release of a game.

Inhouse Development. We capture vertical integration by creating a dummy that takes value 1 if a game’s developer and publisher belongs to the same parent company.

Licensed Middleware. We identified games using development tools for cross-platform development with this dummy, which takes value 1 if the game uses a licensed graphics engine (including well-known graphics engines categorized as “middleware” in our dataset), game engine, or 3D engine.

RESULTS

Hypothesis 1 states that multihoming complements have lower quality on complex platforms compared to the other platforms they are released to. Columns 1 and 2 show that Complex Platform indeed has a negative and statistically significant coefficient, providing support for our first hypothesis for consoles of sixth and seventh generation, respectively. Hypotheses 2, 3, and 4 investigate moderators of the relationship between platform complexity and multihoming complement quality. Hypothesis 2 predicts that the drop in quality on complex platforms decreases in the existence and the extent of a delay between initial release and subsequent release on a complex platform. We test this by interacting Complex Platform and ln(Delay in Days). Results report supporting evidence for the Hypothesis; the interaction is positive and significant so that the longer the time between the first release of the game and the release to the complex platform, the smaller the difference in game quality between Complex Platform and other platforms. We find support for Hypothesis 3 that vertically integrated firms have lower drops in quality on a complex platform for Generation 6, but not for Generation 7. We do not find support for our Hypothesis 4 that licensed middleware reduces the quality drops on complex platform. In fact, the interaction between Complex Platform and Licensed Middleware in Column 1 shows a negative and marginally significant coefficient.

INSERT TABLE 1 ABOUT HERE

DISCUSSION AND CONCLUSION

We study the impact of platform architecture on multihoming complement quality and consider the complexity of the underlying platform core technology as a key platform specific factor in determining the quality of multihoming complements (after controlling for game-specific factors). We look at multihoming video games in two generations of video game consoles from 1999 to 2010 and find support for our main prediction that games receive lower quality scores on complex platform architectures. Put simply, it is more difficult to port a game onto a complex platform and accordingly, quality performance of the complement on that platform is lower. This confirms that platform characteristics, and in particular their technological complexity, matter for the decisions and outcomes of complement providers to develop and port their complements to specific platforms. Multihoming therefore depends on which platform a complement is multihomed to. Moreover, as complex platforms are typically the most advanced of their generation, we add another dimension to the discussion on why superior technological performance does not always yield a competitive edge (Anderson et al. 2014, Kretschmer and Claussen 2016, Claussen et al. 2014).

Our study of the video game consoles also sheds light on the different product and firm strategies that are more or less suited to manage complements for a complex platform. We find that simultaneous game releases perform worse on complex platforms than staggered releases, as under time pressures the design of these complements tends to be more modular and hence less integrated for a complex platform. We also found that firms vertically integrating the development and publishing of the game experienced lower drops in quality when porting their products to a complex platform. Finally, we did not find robust evidence that the use of licensed middleware facilitated multihoming on complex platforms.

Overall, our findings show the value of considering platform architecture in explaining the evolution of platform markets and systems. Hence, we answer recent calls to bring the IT artifact to the core of theory development (Tiwana et al. 2010, Orlikowski and Iacono 2001). We offer a first answer to the question of "how platform architecture influences the evolutionary dynamics of ecosystems and modules [complements] in platform settings" (Tiwana et al. 2010) by focusing on a key dimension of platform architecture: complexity. Interestingly, a seemingly beneficial design choice by the platform owner (higher performance that brings about complexity) results in lower quality multihoming complements. This has multiple implications. First, complex platforms require higher co-specialization by third-party developers to achieve high-quality of software performance. Using cross-platform development technology such as middleware tools does not help in avoiding these platform-specific investments; exclusive complements are better suited. The owner of a complex platform might need to encourage more exclusive products (or develop them itself) to take advantage of its architecture, and affect the evolution of the platform rather than sticking to lower-quality multihoming complements. Our results also let us make some predictions about platform evolution. For example, we expect that complex platforms attract more games from vertically integrated developers of complementary products as they have the organizational means to overcome the corresponding integration problems. Conversely, if the industry starts to rely more on simultaneous releases, complex architectures may turn out to be a burden for the platform owner, especially in an environment where exclusivity is the exception.

1. Middleware in game development generally refers to “software that adds services, features, and functionality to improve the game as well as make game development easier.” https://software.intel.com/en-us/articles/middleware-in-game-development, accessed 10 September 2016.
2. “Saturn had eight processors… to get that platform to do what it was designed to do was a very complex and painful learning process for developers, including the best and sharpest minds that Sega had to bring to bear on it, in both Japan and the U.S.”, Interview: Joe Miller (Sega of America Senior VP of Product Development), http://www.sega-16.com/2013/02/interview-joe-miller/, accessed 07 September 2016.
3. “Trying to program for two CPUs has its problems. … The two CPUs start at the same time but there's a delay when one has to wait for the other to catch up... I think that only one out of 100 programmers is good enough to get that kind of speed out of the Saturn.” (Yuji Naka, lead programmer and creator of the “Sonic the Hedgehog”; Pettus, 2013; p.193).

REFERENCES AVAILABLE FROM THE AUTHOR

Table 1. Quality Differences across Complex vs. Non-Complex Platforms. DV = Review Score of the Game on the Focal Platform.

	1	2
ln(Delay in Days)	-0.747***	-0.989***
	(0.070)	(0.172)
Average Quality of 1st Party Exclusives	0.027	0.066
	(0.028)	(0.060)
Average Quality of 3rd Party Exclusives	-0.026	0.206**
	(0.036)	(0.072)
Complex Platform	-1.029***	-2.642**
	(0.275)	(0.878)
Complex Platform x ln(Delay in Days)	0.314*	0.685**
	(0.145)	(0.228)
Complex Platform x Inhouse Development	0.829*	0.338
	(0.323)	(0.497)
Complex Platform x Licensed Middleware	-0.925^	-0.450
	(0.478)	(0.645)
Constant	71.791***	53.406***
	(2.885)	(7.419)
Observations	1,413	806
R-squared (within)	0.144	0.104
Number of Titles	612	361
Game FE	Yes	Yes
Generation	Sixth	Seventh

Notes. Standard errors in parentheses. *** p<0.001, ** p<0.01, * p<0.05, ^ p<0.10