Platform Competition and Complementor Responses: Insights from Combining Design Rules with the Comparative Adjustment, Transaction and Opportunity Cost Framework

Argyres, N.; Nickerson, J.; Özalp, H.

DOI
10.1093/icc/dtac027

Publication date
2023

Document Version
Final published version

Published in
Industrial and Corporate Change

Industrial and Corporate Change, 2023, **32**, 112–128
DOI: https://doi.org/10.1093/icc/dtac027
Advance access publication date: 31 May 2022
Original Article

OXFORD

Platform competition and complementor responses: insights from combining design rules with the comparative adjustment, transaction, and opportunity cost framework

Nicholas Argyres[1,*], Jackson Nickerson[1] and Hakan Ozalp[2]

[1]Olin Business School, Washington University in St. Louis, One Brookings Drive Campus Box 1156, St. Louis, MO 63130, USA. e-mail: argyres@wustl.edu; e-mail: nickerson@wustl.edu and [2]Amsterdam Business School, University of Amsterdam, De Boelelaan 1105, Amsterdam 1081 HV, The Netherlands. e-mail: h.ozalp@uva.nl

*Main author for correspondence.

Abstract

Platform owners regularly make decisions regarding whether, when, and how to upgrade their platforms as they compete with other platforms. These decisions must consider how platform complementors and rivals are likely to respond to various changes because a platform suffers in competition when the quantity or quality of its complementors is low. This paper analyzes such interactions in the US videogame industry. We show how a combined design rule-comparative adjustment, transaction, and opportunity cost framework can help to explain and predict these interactions in this industry.

JEL classification: L20, O31

The power of modular business and technological systems is great indeed. The system interfaces or "thin crossing points" (Baldwin, 2008) in modular systems attract and leverage the innovative local efforts of a wide range of specialists while limiting the costs of coordinating specialists' innovation choices (Baldwin and Clark, 2000). However, while modular systems facilitate innovation within modules, achieving innovation at the system level is more difficult. Challenges of system-level innovation loom particularly large in modular systems in which the specialists are independent firms over which the system owner lacks hierarchical authority (e.g., Teece, 1996; Chesbrough and Teece, 2002). Platform-based ecosystems are extreme forms of these modular systems and therefore face critical challenges of system-level innovation (Baldwin and Woodard, 2009; Gawer, 2014).

In this paper, we are concerned with challenges associated with system-level innovations in competing, established platform-based ecosystems for which the ownership of the entire ecosystem by one firm is infeasible (Jacobides *et al.*, 2018). We develop a general framework for analyzing these challenges that combine Baldwin and Clark's (2000) notions of design rules (DRs) with the comparative adjustment, transaction, and opportunity (CATO) cost framework presented in Argyres *et al.* (2019). DRs include features of a platform's technical architecture (hardware and operating system), interfaces, integration protocols, and other policies that affect the openness of the platform to complementors.[1] The CATO framework predicts when, to where, and how firms strategically reposition in response to an innovation shock. Considering both DRs and the CATO framework, we explore interactions between platform owners' and complementors' strategic choices, and the resulting competitive dynamics and market evolution. More specifically, we are concerned with the following research question: *How do platform owners' decisions about platform DRs affect strategic positioning decisions by complementors and competing platform owners, thereby shaping system-level innovation and firms' competitive performances?*

While Baldwin and colleagues' pioneering research on modularity has addressed the impacts of DRs and innovation regimes on various kinds of costs (e.g., Baldwin, 2008; Baldwin and von Hippel, 2011), it has not directly addressed the interactions between DR choices and complementor response strategies. Other literature focuses on complementor strategies during platform transitions (e.g., Venkatraman and Lee, 2004; Kapoor and Agarwal, 2017) but does not address the effects of CATO costs in particular. Our emphasis on these kinds of costs contributes to the literature by illuminating the considerations that platform owners must consider when deciding on the timing and nature of system innovations.

To ground our analysis, we consider system-level innovation challenges in the console video game market, which involves competing platform (i.e., console) owners and the developers that create and publish games for the consoles.[2] Console providers choose a strategy for launching a new console that aims to attract both sides of the market: consumers and developers. Console providers also determine the timing of a console's launch. A console's DR and launch timing affect choices by incumbent and entering developers regarding the console for which to program their games (often referred to as "joining" or "supporting" the console). Console owners desire to attract developers in order to increase the library of games available for their consoles and thereby increase the attractiveness of the console to consumers.[3] Developers are thus (potential) "complementors" (Brandenburger and Stuart, 1996) to console owners. In addition to choosing which console(s) they will support, developers choose the genres and designs of their games, and how to market and promote them in competition with developers programming for the same and different consoles.

Our analysis shows how the DRs for new and upgraded consoles significantly affected game developers' and competing console providers' CATO costs. These costs in turn played a large role in determining how developers and rivals responded to these DR changes, which affected their competitive performances. We maintain that the CATO framework is useful for understanding and predicting these responses. The implication is that all players should take CATO costs into account when deciding on, or responding to, new DRs and that such costs will determine a variety of firm- and industry-level outcomes from such rules. Indeed, taking these kinds of costs into account may be an important aspect of what Baldwin (2015) terms "dynamic architectural capability," which she defines as the ability to "change the system's structure to manage bottlenecks and modules in conjunction with the firm's organizational boundaries and property rights."

1 Baldwin and Clark (2000: 77) define technical architecture as the set of modules that are part of a system and their roles; interfaces as "detailed descriptions of how the different modules will interact, including how they fit together, connect, communicate, and so forth"; and integration protocols and testing standards as "procedures that allow designers to assemble the system and determine how well it works, whether a particular module conforms to the design rules and how one version of a module performs relative to another."

2 The broader video game industry includes the markets for personal computers, arcades, and mobile phone games. For ease of exposition, we use the term "developers" to refer to both video game developers and video game publishers. Developers (game development studios) do the creative and programming work, whereas publishers provide financing, distribution, marketing, and often resources for development, such as development tools (Tschang, 2007). Publishers also manage licensing agreements with platform owners. Developers can be independent firms that work under contract with publishers, or they may be owned by publishers or platform owners. In this article, we only consider developers that are not owned by platform owners because developers owned by platform owners cannot reposition to other platforms.

3 Video game consoles are subject to both direct and indirect network externalities (Corts and Lederman, 2009). Many gamers enjoy playing with each other. Therefore, gamers often favor consoles their friends own, driving direct network effects. Gamers also prefer the console for which a wide selection of games is available, driving indirect network effects. Both effects imply that demand for a given console depends in part on the number of users of that console.

1. Literature on platform transitions

Research in strategy and technology management has emphasized the importance of complementor responses during transitions to new platforms and studied various aspects of those transitions (Adner, 2017). Scholars have shown how the newness of a platform and the network structure of a complementor affect platform choice (Venkatraman and Lee, 2004); how system maturity leads to poor performance by late-adopting complementors and complementor free-riding (Rietveld and Eggers, 2018; Cennamo and Santalo, 2019); how a system's technical architecture affects complementor engagement and performance during transitions (Kapoor and Agarwal, 2017; Saadatmand *et al.*, 2019); and how a complementor's experience affects its performance during transitions (Ozalp *et al.*, 2021). The literature has also examined how knowledge-sharing by platform owners can ease complementors' transitions (Ozalp *et al.*, 2018; Foerderer, 2020; Fang *et al.*, 2021).

An important theme in this literature, therefore, is how platform transitions affect heterogeneous complementors differently. The literature has been largely empirically driven, however, and while a few efforts have attempted to develop broader frameworks (e.g., Gawer, 2014; Adner, 2017; Jacobides *et al.*, 2018), those efforts have not contemplated the interactions between DRs and CATO costs that, we maintain, play important roles in determining strategic repositioning choices and hence firm performances during platform transitions.[4]

2. Console generations

Operationalizing our joint DR-CATO framework requires detailed information about a specific market setting to which it is applied. Therefore, we begin with a brief history of the many platform transitions in the video game market, with a special emphasis on console generations 5–7, covering 1993–2012. Following this history, we illustrate how linking DR logic to the CATO framework informs repositioning choices in the console video game market as a way to illuminate how the combined framework can be applied to understanding the strategy and evolution of this and other platform-based industries. We posit that the joint DR-CATO framework can provide useful guidance for managers to make these kinds of decisions.

Throughout the history of video game consoles, consoles have tended to be introduced in clusters referred to as "generations." Each generation offers technological leaps in hardware that increase the potential game speed and therefore the graphical capability of games. In addition to providing faster consoles in each generation, console owners often change the DRs for new and upgraded consoles. These new DRs affect the strategic decisions made by complementors and by rival console owners. We briefly summarize the first seven console generations and then describe the strategic interactions surrounding generations 5–7 in more detail.

Launched in 1972 with the Magnavox Odyssey, the first-generation consoles were comprised of single-game devices such as Pong, a game introduced by Atari in 1975. As a result, the first-generation consoles could only play the game(s) that came installed on the console. The second-generation console owners, of which Atari is the most prominent, decomposed the design into two components: a console and an interchangeable game cartridge that enabled the user to switch games through a standard interface. This modularization invited a flood of developers to enter the market. Many of these developers produced error-filled or low-quality games, which, together with an economic recession, caused demand to drop precipitously (Boudreau and Hagiu, 2008). Atari failed to recover its market-leading position after the 1983 market crash.

Following Atari's collapse, Nintendo launched its NES (Nintendo Entertainment System) as the first third-generation console. The NES succeeded in re-establishing the console gaming market in the United States and elsewhere through a modular operator (Baldwin and Clark, 2000: 12 and 13) of exclusion. In particular, Nintendo employed a security chip that excluded unlicensed games from operating on its consoles. This security chip thus allowed Nintendo to exercise strict

4 Additional contributions include de Vaan (2015), who studied the effects of developers' ties on their survival during transitions, and Cennamo (2018), who studied how the value created by a console depends on the number and variety of games developed by developers and the console owner.

quality control over games written for the NES.[5] Performance quality in the third-generation consoles also benefited from 8-bit microprocessors that greatly expanded color availability, audio channels, and graphics capabilities.

The fourth-generation consoles introduced a substantial advancement in technology by using 16-bit microprocessors. The consoles featured a greatly increased range of colors, more pixels, and better audio capabilities for two-dimensional (2D) games. Sega launched its Mega Drive system in Japan in October 1989 and in the United States (as Genesis) in November 1990. Nintendo, its primary competitor, initially enjoyed a commanding market share with its third-generation NES and delayed launching a fourth-generation console. Steady market share erosion, however, eventually caused Nintendo to respond by introducing its 16-bit Super NES, but it reached the market 2 years later than Sega's console. Nintendo's delay enabled Sega to build a substantial market share lead.

Launched in 1993, the fifth-generation consoles were a watershed because the technology offered 32-bit microprocessors, 3D graphics, and optical media formats compact disc read-only memory (CD-ROMs). These improvements helped Sony sell over 100 million consoles. Some consoles featured expandable memory and multiple processors that greatly increased the complexity of programming needed to take full advantage of the console's performance capabilities, especially for three-dimensional (3D) graphics.

The DRs for the sixth generation changed in the direction of greater modularity. In addition to enhanced 3D graphics, several consoles offered modules that enabled online connectivity. These features made consoles much more cost-efficient compared to a personal computer in terms of gaming performance. Microsoft entered the console market with the Xbox, and the market consolidated to the three console owners that dominate as of this writing: Sony, Microsoft, and Nintendo. Sony chose the DVD as the media format to increase its user base by allowing users to enjoy a single piece of entertainment hardware for both movies and games. Sony's sixth-generation console, the PlayStation 2, became the highest-selling and longest-lived console ever. Over 155 million units of this console were sold until its discontinuation 12 years later, in 2013.

The seventh generation introduced yet another set of DRs. The changing of several modules made consoles function even more like personal computers, although the microprocessor designs were different. Consoles featured modules for interfacing with the internet, networking for wired and wireless connections, and online multiplayer games. Consoles came with HDMI ports, motion controllers, and high-definition optical discs. Competition between 2005 and 2017 was mainly between Xbox 360 (Microsoft), PlayStation 3 (Sony), and Wii (Nintendo). Each of these firms sold over 80 million consoles during this period.

3. Console owner and developer strategies during the fifth, sixth, and seventh generations

While each console generation introduced substantial system-level innovation that increased the complexity and realism of video games, these improvements came at a high cost to game developers. For example, the cost to develop the blockbuster game *Wing Commander III* was $5 million in 1993 (Kotaku, 2014), while the cost to develop *Grand Theft Auto V* in 2013 was $265 million (*The Economist*, 2014). Therefore, in deciding whether to program for a new or upgraded console, and if so which one(s), developers faced difficult trade-offs between game cost and quality. Developers could choose whether their games would use the full or less-than-full functionality of the new or upgraded console's hardware. They also faced the choice of whether to develop games that could operate on a single console only or on multiple consoles (so-called multihoming). If a developer decided on multihoming, it faced a further decision regarding whether to do so sequentially (one console at a time) or simultaneously. The varying DRs in each generation of consoles, as well as across different console owners within the same generation, greatly influenced these strategic decisions. We therefore summarize critical DR alternatives for each generation in more

5 Nintendo was so dominant in the 1980s that it was able to restrict developers from developing games for competing consoles. In 1991, however, the Federal Trade Commission ruled that this behavior violated anti-trust law and punished Nintendo.

detail before describing our framework for understanding console owner and developer strategic choices across three console generations.

3.1 Fifth generation

A key aspect of the DRs for the fifth-generation consoles is that they varied significantly across different console owners. Nintendo designed its generation 5 console with highly complex DRs to attract more technically capable developers who could provide faster, more graphically intense games. This strategy was successful in that a group of highly capable developers (an example is the developer Factor 5) chose to write for the Nintendo console, even though Nintendo continued to use relatively expensive cartridges that imposed extra costs on developers. In contrast, Sony chose comparatively simple DRs for its PlayStation 1 console that featured much less expensive CD-ROMs. Sony also provided developers with software libraries together with its software development toolkit to ease programming for the console. Sony's strategy was to attract a wider range of (perhaps less capable) developers to stimulate the production of a large library of games for its console. Sony's strategy was successful; many developers wrote popular games for the platform, and developer support for Sega's competing Saturn console plummeted. Ultimately, PlayStation 1's user base grew to represent 60% of the fifth-generation market (Corts and Lederman, 2009), which contributed to Sega's exit from the console market following the release of its sixth-generation console, Dreamcast.[6] Thus, console owners' design and business strategies took heterogeneity in developer technical capabilities into account.

3.2 Sixth generation

Sony made a major change in its DRs for its generation 6 console, PlayStation 2. The new console introduced highly complex DRs that required developers to make significant investments to build expertise in the console's hardware so as to take full advantage of its functionality. Analysts assessed that Sony sought to leverage its dominant position with developers to lock them in by inducing them to specialize in writing games for Sony's hardware (Reimer, 2005). However, many developers complained that building the expertise required to write games for PlayStation 2 was excessively expensive. Sony's extensive libraries provided with the development toolkit for PlayStation 1 were not initially provided for PlayStation 2 (Evans *et al.*, 2006).

Sony responded, however, by offering a "Tools & Middleware" program in which third-party companies were granted licenses to Sony's proprietary hardware and application programming interface (APIs) to create value-added "middleware." This middleware could, in turn, be licensed to developers (by third-party middleware developers) to accelerate game programming. In the console gaming context, middleware is a multi-module software development environment or a single software module that can be used to develop games. The former includes game engines such as Unreal Engine and Unity, and the latter includes individual modules for real-world physics such as Havok and procedural tree and terrain generation such as SpeedTree. While console providers had always provided game development toolkits to developers based on assembly language, and later on higher-level languages such as C, ready-to-use middleware module(s) and development frameworks did not arrive until the sixth generation. In addition, some developers leveraged "higher-level" middleware that facilitated platform-agnostic development of games with cross-platform APIs.[7] Middleware thus allowed the writing of video games by using ready-made modules and software libraries that are abstracted at the "higher level," thereby lowering the cost of "porting" across consoles supported by the middleware.

6 Sega's exit derived from several causes. Perhaps the most important causes were (i) frustration by developers (especially Electronic Arts) with Sega's non-standard chipset and hard bargaining over licenses and (ii) intense competition from PlayStation 2.

7 The term "middleware" has several definitions depending on the domain in which it is used. In the video game console market, middleware functions between a console's operating system and game programming, acting as an abstraction layer. Middleware facilitates game development by eliminating the need to develop programming modules and tasks from scratch and by minimizing platform-specific coding in assembly language. It also reduces the need for idiosyncratic coding methods and structure required to program for a console. The trade-off is that middleware usually cannot deliver the highest levels of software performance, which limits processing speed.

Sony collaborated with middleware providers to sponsor expositions in Japan, the United States, and Europe. It also published newsletters that shared stories of successful game development using the middleware. Many developers eventually adopted these tools and middleware, with Renderware being a prevalent one at the time. PlayStation 2 became highly successful based on its extensive library of games, made possible in part by middleware.

Observing developers' positive responses to Sony's "Tools & Middleware" program, Nintendo and Microsoft followed with their own similar programs. In a perhaps unintended consequence for Sony, however, the use of third parties for the provision of tools and middleware made it easier to develop games for PlayStation and to multi-home them across competing consoles. Third-party-developed tools and middleware featuring high-level programming languages (and cross-platform APIs) thus created greater programming modularity by facilitating platform-agnostic game development. Multi-platform middleware enabled developers to capture economies of scale and scope in game development (Corts and Lederman, 2009). As a result, sequential and simultaneous multihoming grew dramatically during the sixth generation.

3.3 Seventh generation

While Sony maintained its strategy with another complex set of DRs for its seventh-generation PlayStation 3, Microsoft adopted simpler DRs for its competing console, Xbox 360, while using the same IBM processor as Sony. Microsoft's choices made it relatively easy and cheap to write games for Xbox 360, which led most developers to multi-home their games across Xbox 360 and PlayStation 3. Middleware therefore became even more widely used in the seventh generation. Microsoft's successful strategy catapulted Xbox 360 to a near equal market share with PlayStation 3.[8]

Meanwhile, Nintendo made an even more radical change to the DRs for its new console, the Wii. Wii was based on older and simpler technology, making it the least expensive console for which to develop games. Wii quickly attracted a large number of users because it appealed to customers beyond the usual segment of teenage and young adult male gamers. Yet Nintendo's strategy was only partly successful because many of its casual users soon lost interest in Wii, turning instead to fast-growing mobile phone-based games. In addition, building Wii on an older, simpler technology meant that Wii's hardware was not powerful enough to run the more graphics-intensive, "bleeding-edge" games (e.g., the latest action or first-person shooter games). Developers such as THQ, which made large bets on Wii and had outperformed other developers in the first few years after Wii's launch, went bankrupt as many consumers moved to either low-end mobile phone games or high-end games on Xbox 360 and PlayStation 3. Consumers who especially enjoyed Nintendo-produced titles remained loyal to Wii, but this, of course, did not help independent Wii developers.

This brief history of video game console generations until the seventh generation illustrates how platform owners' design choices affected consumer choices by affecting complementors' strategic decisions. Complementors' decisions were impacted by the ways in which the choice of console DRs shaped their production costs and time to market. Moreover, because the market encompassed multiple console owners, hundreds of developers with heterogeneous capabilities, substantial leaps in technology, and variations across each console generation, the strategic decisions for each console owner were complex and ill-structured for each generation. Similarly, complex decisions arise in other platform-based industries in which complementors' technological choices are sensitive to platform owner decisions, including enterprise software, social media, smartphones, and the like. How can managers assess and process all of this information to guide strategic decision-making? We suggest that the CATO framework (Argyres *et al.*, 2019), which has shown empirical purchase in the early auto (Argyres *et al.*, 2015) and bicycle gear (Bigelow *et al.*, 2019) industries, is helpful in this regard. We describe this framework in the next section.

8 Xbox 360's simpler technology also enabled it to be released a year earlier and at a lower price than PlayStation 3.

4. The CATO framework

The CATO framework offers a methodology that can be used to predict when, to where, and how firms strategically reposition in response to an exogenous shift in supply or demand, an innovation, or a new regulation. If that change leads to a demand surge that is remarkably rapid, significant, and unanticipated, we refer to it as a "shock." The CATO framework uses comparative assessment to predict when, to where, and how incumbents will strategically reposition following an exogenous shock.

Adjustment costs refer to the costs incurred from changing a strategic position in the market and include costs from developing the organizational capabilities, assets, and marketing efforts needed to serve customers in the new position.[9] Transaction costs stem from the contractual hazards associated with outsourcing instead of integrating co-specialized capabilities needed for repositioning. While outsourcing can speed repositioning, it may result in less valuable co-specialization or generate maladaptation costs from inappropriate governance choices. Therefore, incumbents must assess whether faster repositioning is worth the potential efficiency loss from lower product values or higher transaction costs, at least until the incumbent arrives at the new strategic position and can re-invest in needed co-specialization under the safeguards provided under hierarchical control. Opportunity costs arise when incumbents move to a new strategic position early or late. A decision to move early involves reconfiguring assets and capabilities that causes an incumbent to forgo profits that would have been earned if repositioning was delayed or avoided entirely. Alternatively, moving late can result in ceding first-mover advantage, and the benefits that derived from it, to rivals who move to the same position earlier. Among other factors, an incumbent's opportunity costs of moving early or late are determined by the growth in market demand in the original segment in which an incumbent was positioned as compared to the new segment, as well as by the competitive advantage of the incumbent's position in each segment vis-à-vis rivals in those segments.[10]

The term "comparative" has three implications. First, the strategist compares adjustment, transaction, and opportunity costs to reposition from a firm's existing position to each plausible new position. Second, the strategist compares these costs among all firms for each plausible new position, which informs whether outsourcing key transactions in the short term is, on net, beneficial given plausible repositioning by rivals. All firms must be considered because even if a focal firm has lower adjustment costs to a given position than some of its rivals, other rivals may enjoy even lower adjustment costs. Third, these comparisons are relative in the sense that only ordinal (rank order) instead of cardinal (quantitative) measures are generally needed to make an assessment, which (greatly) diminishes information needs. By highlighting the roles of comparative adjustment costs, transaction costs, and opportunity costs, the CATO framework facilitates context- and firm-specific predictions about when, to where, and how firms will reposition in response to an exogenous change.

5. Console design and developer decisions

Introducing a new generation console in the video game console market can be seen as a "shock" in that it represents an important change that rivals and developers take as exogenous, and to which they must respond in some way.[11] After the introduction of a new console, developers and rivals may wish to strategically reposition because new DRs and technology are introduced for that console. This section analyzes how developer CATO costs are influenced by console

9 Adjustment costs of this kind were not discussed by Williamson (1991), who argued against an important role for strategic positioning.

10 Because such relative demand growth is uncertain, an incumbent's expectations about it are important in the repositioning decision (Schilling, 2003). This expectation is especially relevant when, as in the case of platforms, demand is subject to network externalities and therefore can increase or decrease quickly (Hagiu and Spulber, 2013).

11 Argyres *et al.* (2015) define an innovation shock as a new-to-the-world product whose launch was not anticipated by rivals or others, and for which demand increases significantly in an unanticipated way. While developers and console rivals can anticipate that eventually a new generation and its consoles will be introduced, the precise timing and DRs are unknown until a final decision is revealed by the platform owner. Because console producers have a strong competitive incentive to keep DR decisions private as long as possible, we treat a new generation console as an exogenous innovation shock.

owners' DR decisions regarding potential repositioning. We consider two major categories of DR decisions emphasized in the literature: backward compatibility and "platform openness."[12]

5.1 Backward compatibility

When deciding on the DRs for a new platform, the platform owner must decide whether to incorporate backward compatibility with the prior platform. Backward compatibility means that complementary products for the prior generation of platform can operate on the new platform, implying a greater degree of modularity. A classic example of backward compatibility is the ability of Microsoft's old computer file formats (and software) to run on Microsoft's productivity software (and modern operating systems) even after 30 years.

In the console video game market, backward compatibility creates a demand-side incentive for users to upgrade to the next-generation console so that they can continue to use their existing games while potentially enjoying the new platform's improved performance or features (Kretschmer and Claussen, 2015). For example, at its launch, PlayStation 2 was backward compatible with almost all PlayStation 1 games, which encouraged users to upgrade to Sony's new console.

The backward compatibility decision also shapes comparative adjustment costs for developers, with these effects differing depending on developer technological capability. For instance, a next-generation console that is backward compatible generates low adjustment costs only for developers whose games do not use a new console's advanced features. This is because these developers may simply release new versions of their games that were developed for the old console. Alternatively, they may develop versions for a new console that require few new generation-specific features. For example, they may create sequels that rely more on story or theme innovation rather than heavy graphical programming.

The more that a developer positions itself with bleeding-edge games that rely on advanced console DRs and features, the greater will be its adjustment costs to a next-generation console if it is to replicate its position. Nonetheless, adjustment costs to the next-generation console could be lower than the costs needed to reposition to a different and incompatible console.[13] Therefore, a new console featuring backward compatibility is likely to stimulate a greater increase in the supply of games.

DR decisions about backward compatibility also affect developer opportunity costs. For example, delaying the development of a next-generation game may entail low opportunity costs if consumers are not demanding leading-edge graphics and audio for the game. A new generation console with DRs that involves low developer adjustment costs may increase entry and competitive intensity and therefore increase the opportunity cost of repositioning late to the new console. Thus, the choice of backward compatibility can impact the timing and variety of game availability from incumbent developers through its effects on developer CATO costs.

Owners sometimes choose DRs so that the platform is initially backward compatible, but this backward compatibility can be removed later. The DRs for these kinds of decisions open up another dimension along which the platform is modular, albeit usually at a cost. For example, some owners included older processors alongside their new console's new processor to provide backward compatibility for the consoles but stopped this practice a few years later.[14] This approach to backward compatibility aimed to boost early game supply and later to spur demand

12 There are of course other decisions that console owners face when launching a new console and attempting to attract developers for it that we do not consider here. For example, a console owner may wish to lower comparative adjustment costs for particular developers whose games were highly demanded in the previous generation. This can be achieved by licensing with preferential terms and early access, providing early information on the timing of new generation introduction, disclosing the plans of rival developers, contractually committing not to introduce or license others to introduce directly competing games, offering to promote their games to consumers, and bestowing awards on them (Rietveld *et al.*, 2019).

13 If hardware modules in the older and newer generation console are related architecturally, then the adjustment costs to the newer generation are likely to be less than the cost of moving from a console provided by a given owner (e.g., Sony) to a next-generation console provided by a different owner (e.g., Microsoft), all else equal.

14 After generation 6, consoles favored software emulation-based backward compatibility over hardware-based ones, which created difficulties for architecturally complex consoles. For example, Sony was able to offer backward compatibility from PlayStation 4 to PlayStation 3 only through a streaming service. Providing software emulation in PlayStation 4 proved technologically difficult because PlayStation 3's complex DRs were significantly different.

for the new console. Sony employed this strategy when it introduced its PlayStation 2 Slim and PlayStation 3 Slim, neither of which offered the older processors. Removing backward compatibility raises developers' opportunity costs if they do not develop new versions of their games for the current generation console. Yet, because they are repositioning late, adjustment costs are typically low at that point.

5.2 Platform openness

One of the most fundamental choices platform owners make is how open or "accessible" they make their platforms for complementors. Apple Computer, for example, famously insisted on an almost completely closed operating system for its computers in the 1980s, while the competing IBM PC/MS-DOS architecture was almost completely open. We conceptualize openness as a matter of degree, depending on the platform's DRs. We distinguish between two types of openness decisions, where openness refers to the accessibility of the platform to complementors: (i) those that affect openness *between* one platform generation and the next, and (ii) those that affect openness *within* a single platform generation. We maintain that taking platform complementors' CATO costs into account informs decisions with regard to each type of openness.

5.3 Platform openness between generations

A console's openness is determined by its DRs, especially their degree of complexity (Cennamo *et al.*, 2018). More complex rules require greater programming capability and associated costs to fully access console processing and graphics capabilities (Ozalp *et al.*, 2018). Console owners choose a microprocessor, which affects potential computational speed, graphics, and audio performance; memory, which impacts several aspects of game performance; various hardware features such as input/output devices; and the APIs. Recall, for example, that Nintendo's third-generation NES was partially open because it limited access to licensed developers. Sony's PlayStation 1 was more open because its licensing terms were less strict, and its simpler DRs eased game development. These same console owners later offered consoles that were less open because their DRs were more complex (e.g., PlayStation 2 and Nintendo 64).

A new console featuring comparatively simple and easy-to-program DRs implies low adjustment costs for developers who wish to reposition to support the new console, as well as for those who wish to reposition to develop new kinds of games for that console. Because simpler DRs enhance openness, they are likely to attract a larger number of developers, which simultaneously increases the opportunity costs for developers who reposition late to the console. On the other hand, simple DRs also can make it difficult for some developers to reposition to the new console because such consoles often lack the functionality (e.g., advanced graphics) needed by a developer's games. For example, *Grand Theft Auto*, the blockbuster game series by Rockstar Games (formerly DMA Design), was never developed for Wii because the game's advanced graphics could not be programmed for Wii's simple architecture. Thus, simpler DRs based on older technology make a console less attractive for developers who wish to differentiate based on bleeding-edge features such as advanced graphics and audio.

On the other hand, more complex DRs based on the newest technology greatly raise adjustment costs because even sophisticated developers must invest in co-specialized capabilities for programing a new console if they are to reposition early with bleeding-edge games. Therefore, a CATO analysis would predict that bleeding-edge game developers that reposition early to a new console will tend to vertically integrate because such firms incur lower transaction costs.[15]

Data from Mobygames.com are consistent with this expectation. Recall that publishers finance, market, and distribute video games, with some also being backward integrated into game production. The data indicate that vertically integrated publishers produced 56% of the games of the (complex) PlayStation 2 during its first 2 years, but only 50% for the (simpler) Xbox during its first 2 years. However, for bleeding-edge games, the differences are greater, with PlayStation 2 at 67% and Xbox at 56%. Similarly, vertically integrated publishers produced

15 Because the investments are console-specific, transaction costs increase even for those less capable developers whose games are not bleeding-edge and who subcontract for game development as part of their repositioning strategy.

79% of the games available for the (complex) PlayStation 3 during its first 2 years, but only 71% of the games available for the (simpler) Xbox360 during its first 2 years, and only 54% of games for the (simplest) Wii during its first 2 years. The comparable figures for bleeding-edge games were 70–80% for PlayStation 3 and Xbox360 and 60% for Wii.

As these data suggest, DR complexity and the newest technology can help to attract technologically capable developers who want to reposition early by making co-specialized investments to take full advantage of the console's performance capability. Developers without the requisite capabilities and resources, on the other hand, might avoid repositioning to bleeding-edge games or attempt to outsource game development, thereby incurring higher transaction costs. Alternatively, they may wait for knowledge dissemination through developer-community learning to take place so that their adjustment costs are lower, or they may not reposition to the new console at all. In sum, the degree of openness implied by a console's DRs differentially affects complementors' repositioning strategies depending on their developer-specific CATO costs.

5.4 Platform openness within a generation

After launch, a platform owner can choose to change its platform's degree of openness. We maintain that various complementors' CATO costs will inform a platform owner's within-generation openness decisions. These costs can differ from those associated with launching new platform generations because some developers will have already made platform-specific investments.

The video game market again provides a good illustration. The literature describes several console owners' decisions to attract developers after a new generation console is launched. For example, console owners sometimes sponsor educational initiatives such as developer conferences to increase openness by teaching developers how to write games for the new platform generation (Fang *et al.* 2021; Foerderer, 2020). While large developers may benefit from such educational efforts, smaller, less well-resourced developers are more likely to be the beneficiaries.

Less discussed in the research literature is the role of middleware, which our brief history of console generations above mentioned. Recall that the term "middleware" encompasses a wide range of software technologies that ease the development of applications either as a multi-module software development environment or as ready-to-use software modules that can be implemented within the application directly. Such middleware is also common in smartphone operating systems such as Apple iOS and Google Android and has become a mainstay of the video game console market since the sixth generation. Recall that video game middleware includes lightning, face animation, real-world physics, procedural landscape and tree generation, networking code, video compression, user interface, input/output, gaming artificial intelligence, and a variety of other tools and software libraries.

Introducing middleware within a console generation can lower developers' adjustment costs, thereby attracting more of them to support the console, increasing the console's attractiveness to consumers. For example, according to Mobygames.com, in the 8 years following Microsoft's introduction of its XNA middleware for Xbox360 in 2008, as many as 649 games using middleware were released for Xbox360 by independent and mostly small developers making simpler, casual games.

The decision by a console owner to increase openness after a console launch does introduce some risk, however. When a console owner introduces middleware only after developers who were early supporters of the console incurred significant console-specific investments, it is subjecting those developers to "hold-up" (Williamson, 1975; Klein *et al.*, 1978). This hold-up occurs because the introduction of middleware will attract new developers to the console who can avoid expenses incurred by earlier supporters, potentially dissipating the latter's profits if increased price competition outweighs the increase in game demand. Concerns about this kind of transaction cost could discourage developers from early support of a new generation console.

Early supporters of a console experience less impact from a hold-up, however, if they choose a strategy that lowers their adjustment costs. For example, a developer can lower its adjustment costs by programming a game that is distant from the bleeding-edge. Note that once again, *comparative* adjustment costs are central; if developers that reposition early have substantially lower comparative adjustment costs, then console owners can introduce middleware without causing substantial hold-up. Thus, empirical research indicates that early console supporters tend to assemble teams of developers who are experienced with multiple game genres (Ozalp *et al.*, 2021). A major reason may be to enable a quick pivot to developing different types of games if the relevant console owner lowers adjustment costs within a generation.

While the preceding discussion assumes that the console owner provides middleware, recall that another option for these owners is facilitating third-party provision through licensing and other activities. Engaging third parties can increase the speed of introducing middleware; companies specializing in middleware typically produce these programs faster than console owners, increasing platform attractiveness sooner than otherwise and locking in complementors.[16] While speed is an advantage, third-party middleware rarely offers technology that can take full advantage of the processing abilities of consoles with complex DRs. As such, the development of proprietary middleware (or codebase) by independent developers may yield greater performance benefits than using middleware provided by third-party providers, particularly for those bleeding-edge game developers who need to deeply cospecialize with underlying complex-design-rule consoles in order to advance the technological frontier.

Relying on third parties for middleware may also involve risks to developers. If some third-party middleware providers are substantially more capable than others, then developers could experience lock-in and expropriation. For example, many developers for the sixth-generation consoles had become dependent on the middleware provider Renderware. When a large developer, Electronic Arts, acquired Renderware, other major developers were forced to switch to less suitable and sometimes less capable providers, or else develop their own proprietary middleware, consuming time and resources that reduced their competitiveness.

Outsourcing middleware provision to third parties also involves risks to the console owner. With strong incentives to provide middleware that can "port" (Baldwin and Clark, 2000) games from one console to a rival one, third-party middleware providers lowered the cost of multihoming, which increased rivalry among within-generation consoles. Multihoming, and hence porting middleware, became increasingly attractive as game development costs increased dramatically, especially from the sixth- to seventh-generation consoles. Porting lowered adjustment costs of developing games for more than one platform and allowed developers to recover fixed costs from sales across multiple consoles.

The emergence of porting middleware invited a new strategic decision for developers. If they planned to multi-home, should they do so simultaneously (develop and launch games for multiple consoles simultaneously) or sequentially (develop and launch games for one console at a time)? Simultaneously launching games on multiple platforms can provide a higher return on advertising. Yet adopting this strategy may not be feasible for games on the bleeding-edge of technology because of differing DRs across consoles or when multihoming limits the co-specialization of code for each console.[17] Indeed, Cennamo *et al.* (2018) show that many simultaneous multihoming developers did not take advantage of the full functionality of PlayStation 3. Instead, they optimized their games for the simpler rules of Xbox 360, causing PlayStation 3 versions of these multi-homed games to perform less well.

In some cases, developers choose sequential multihoming because they must wait for a different console owner to release its new console. In other cases, developers launch a game on one console and only port it to other consoles if the initial release is successful. This strategy is common for developers with limited access to financial resources and is selected or not based on the adjustment cost of porting from one console to another, which in turn depends on DRs and the availability of porting middleware. Notice that both simultaneous and sequential multihoming DRs have profound consequences for the CATO costs of individual developers and their strategies.

16 Recall that Sony's sixth generation was the first console to support extensive third-party middleware. Speed-to-market and anti-trust concerns may have been motivations for licensing to third parties because of concern that, like Nintendo, developers could sue them for refusing to license APIs.

17 An example of the challenges of porting bleeding-edge games is the sequential multihoming of VIS Entertainment's *State of Emergency*, published by Rockstar Games. First developed for PlayStation 2, the game required an eight-person, 8-month engineering effort during 2001–2002 to reprogram for Xbox, which was, at the time, considered a significant effort (https://www.gamedeveloper.com/production/porting-a-ps2centric-game-to-the-xbox-a-case-study-of-state-of-emergency).

Summarizing, the evolution of the console market illustrates how backward compatibility and openness decisions—important aspects of a platform's DRs—affect developer CATO costs, which in turn influence the availability and quality of games for a console. To further illuminate the value of the DR-CATO framework for explaining and predicting heterogeneous responses, we now examine specific developers' responses to new console introductions.

5.5 Specific developer responses

Tables 1–2 show the average number of days that elapsed before four categories of developers introduced games for various new consoles. This "transition data" can be used to examine whether developer response times to new console introduction were broadly consistent with key predictions from the DR-CATO framework; namely, developers with high comparative adjustment costs and low opportunity costs will delay repositioning, and as such are likely to move more slowly to launch games for new consoles. On the other hand, developers with low comparative adjustment and high opportunity costs are expected to reposition more quickly.

Tables 1 and 2 categorize developers as having high adjustment costs from generation 5 to generation 6 if during generation 5 they produced games for only one type of console (sample average = 2). Developers were coded as having high adjustment costs from generations 6 to 7 if during generation 6 they only produced games on one or two types of consoles (sample average = 2.82). This measure is consistent with Argyres *et al*. (2015), who found that the breadth of technological capabilities was associated with repositioning in response to a shock. Developers were characterized as having low opportunity costs of repositioning if (i) more than one-quarter of the games they produced during the fifth generation, or one-third of the games they produced during the sixth generation,[18] benefited less from the highest levels of technical functionality such as high-end graphics[19] or (ii) more than half of the games they produced enjoyed an established user base on an older console because they featured a licensed franchise (e.g., Batman and

18 The values are different between generations because of an overall increase in the production of the games in the aforementioned genres. Both values represent the top 40% of the distribution within each respective generation.

19 The following game genres are in this category: adventure, strategy, role-playing, children's games, family games (e.g., Jeopardy and Monopoly), hunting, fishing, billiards, bowling, pinball, and other arcade-like games.

Table 1. Days to the introduction of the first game for new consoles: from generations 5 to 6

Panel (a)				
Low adjustment cost–low opportunity cost developers				
	PlayStation 2	Dreamcast	Xbox	GameCube
PlayStation1	728	258	306	267
Saturn	709	266	251	341
Nintendo 64	515	174	417	234
Panel (b)				
High adjustment cost–low opportunity cost developers				
	PlayStation 2	Dreamcast	Xbox	GameCube
PlayStation1	956	414	474	387
Saturn	N/A	N/A	N/A	N/A
Nintendo 64	955	N/A	555	522
Panel (c)				
Low adjustment cost–high opportunity cost developers				
	PlayStation 2	Dreamcast	Xbox	GameCube
PlayStation1	474	131	206	209
Saturn	440	115	174	188
Nintendo 64	382	123	143	155
Panel (d)				
High adjustment cost–high opportunity cost developers				
	PlayStation 2	Dreamcast	Xbox	GameCube
PlayStation1	790	282	397	413
Saturn	488	105	N/A	664
Nintendo 64	729	163	390	442

Table 2. Days to the introduction of the first game for new consoles: from generations 6 to 7

	PlayStation 3	Xbox 360	Wii
Panel (a)			
Low adjustment cost–low opportunity cost developers			
	PlayStation 3	Xbox 360	Wii
PlayStation 2	459	613	253
Xbox	429	622	115
GameCube	466	646	146
Panel (b)			
High adjustment cost– low opportunity cost developers			
	PlayStation 3	Xbox 360	Wii
PlayStation 2	786	1100	370
Xbox	692	676	277
GameCube	635	974	262
Panel (c)			
Low adjustment cost–high opportunity cost developers			
	PlayStation 3	Xbox 360	Wii
PlayStation 2	299	285	234
Xbox	286	262	203
GameCube	264	245	207
Panel (d)			
High adjustment cost–high opportunity cost developers			
	PlayStation 3	Xbox 360	Wii
PlayStation 2	465	730	423
Xbox	501	651	389
GameCube	386	631	385

Numerical entries are the average number of days since the release of the new generation console in the given console pair. Each average is calculated as the weighted average of the number of developers' releases on the console. Three consoles were excluded because they were short-lived: 3DO and Atari Jaguar in the transition from generations 5 to 6 and Sega Dreamcast in the transition from generations 6 to 7. N/A = not available because no developers were observed in that particular platform pair within the subsample.

Marvel characters). Developers with high opportunity costs of not repositioning mostly produced bleeding-edge, graphics-intensive games such as first-person shooter games.

This categorization makes several simplifying assumptions. First, adjustment and opportunity cost measures are each based on one dimension only; more detailed data would be needed to develop finer measures of these variables. Second, because of the thresholds mentioned above, developers closer to the medians of these variables are not categorized. Third, the data exclude developers without prior industry experience. Fourth, the analysis omits consideration of transaction costs. This last simplification is appropriate because the CATO framework emphasizes that adjustment and opportunity cost considerations can outweigh transaction cost considerations in the short term.[20]

Given these limitations and others, the patterns in these tables are suggestive and not meant to be taken as a basis for formally examining DR-CATO-based predictions. Nevertheless, the broad patterns are consistent with those predictions. For example, the high adjustment cost developers were slower to reposition than the low adjustment cost developers in the generation 5 to 6 and 6 to 7 transitions—a pattern that holds for all consoles. (In Tables 1 and 2, compare panels (a) and (c) to panels (b) and (d).) This comparison indicates that comparative adjustment costs may play an important role in determining repositioning speed for these developers. Opportunity costs also appear to have played an important role, as high opportunity cost developers were faster to reposition than low opportunity cost developers for given adjustment costs in both transitions. (In Tables 1 and 2, again compare panels (a) and (c) to (b) and (d).) One exception to the general consistency of these data with our DR-CATO-based

20 Future research could examine, for example, the CATO-based prediction that developers with high adjustment and opportunity costs are most likely to rely on outsourced middleware to speed their movement to new consoles. In contrast, those with low adjustment and high opportunity costs are most likely to use proprietary technology (or middleware) to reposition at the highest level of technical game performance.

predictions is the relatively speedy repositioning of low opportunity cost developers to Wii. However, this repositioning is understandable because Wii's early success in attracting children and families created lucrative opportunities for the types of developers we categorized as "low opportunity cost" (in this unusual case their opportunity costs were actually high). Thus, discernable patterns in the tables are consistent with the importance of both comparative adjustment and opportunity costs in determining repositioning speed, in directions predicted by the DR-CATO framework.

To further illustrate the framework's value, consider the following examples of repositioning choices contingent on developers' comparative adjustment and opportunity costs from the Mobygames.com data. Recall that Nintendo 64's complex DRs included the use of game cartridges, while competitors mostly moved to optical media (i.e., CD-ROMs). Cartridge memory was limited and could not accommodate story- and content-heavy games. In addition, Nintendo 64's DRs were not friendly to 2D graphics. As a result, major developers, including Square (maker of the popular *Final Fantasy*, a game in the story- and content-heavy role-playing genre), and Capcom (maker of *Street Fighter*, which at the time relied heavily on 2D graphics), did not reposition to the Nintendo 64 console because of the high adjustment costs such repositioning entailed. Switching to new genres or developing and marketing entirely new games within the role-playing and fighting genres would also have entailed high adjustment costs for these developers. These and similar developers therefore repositioned to PlayStation 1 instead.

Developers who were focused on first-person shooter games, on the other hand, were well accommodated by Nintendo 64's DRs. Acclaim is an example of a developer that was able to develop a shooter game series, *Turok*, for Nintendo 64. Data from Mobygames.com indicate that developers that developed for both the Super NES in generation 4 and Nintendo 64 in generation 5 tended to be more experienced with Nintendo consoles, older in general (with broader capabilities across platforms), and much less focused on role-playing and fighting games. Therefore, such firms enjoyed lower comparative adjustment costs to develop for Nintendo 64, and lower opportunity costs to do so.

A similar pattern played out with Sony's introduction of PlayStation 3—another console that was based on complex DRs. Our data suggest that those developers who invested most heavily in writing games for that new console early on were older, with longer experience developing games for PlayStation 1 and PlayStation 2, and tended to focus on action and shooter games. Indeed, some developers, such as Monolith Productions, were so capable that they were able to develop and sell tools and middleware for PlayStation 3 and other competing consoles, in addition to games. Such firms thus faced lower comparative adjustment costs from repositioning to PlayStation 3 and higher opportunity costs from not doing so. On the other hand, developers who were slow to move to PlayStation 3 tended to have less experience developing for other consoles and lower opportunity costs because they emphasized story-based games. Such firms presumably faced lower adjustment and higher opportunity costs from adapting their games or developing new ones for Xbox 360 and even Wii (especially higher opportunity costs) relative to PlayStation 3.

6. Console DRs and rivals

Our focus to this point has been on how DRs for new and existing platforms affect complementors' CATO costs. However, new generation platforms involve shocks that are also felt by a platform owner's rivals. These rivals must then consider whether to reposition and, if so, when and to where. Platform owners often receive some advance information about the kinds of DRs chosen by rivals, but it is usually incomplete. Therefore, repositioning decisions by platform owners are based on expectations, rather than realizations, about which platform owner(s) will launch first, their various DRs, and demand for the various platforms.

Strategy research has examined within-generation competition among the console using market entry and network effect models (e.g., Zhu and Iansiti, 2012; Cennamo and Santalo, 2013). This research notwithstanding, we maintain that rival console owner repositioning decisions are likely to be informed by the DR-CATO framework, and therefore we explore this possibility.

As explained above, consoles such as PlayStation 1 and Xbox 360 were based on DRs that made them easier to program, thereby lowering adjustment costs for many developers. Indeed, at the end of 1 year following their launch, there were 188 games available for PlayStation 1 and 87 games available for Xbox 360 (Mobygames.com). While these numbers are large, a considerable fraction of these games were "causal titles" produced by lower-capability developers and were available for download only. In contrast, developers of first-person shooter games for the prior console generation did not reposition early to PlayStation 1 or Xbox 360.

Console owners whose rivals introduced simpler-design-rule consoles typically repositioned with complex-design-rule consoles aimed to attract higher-capability developers. The DRs for such consoles typically provided greater capability to support bleeding-edge games, with a corresponding increase in programming difficulty to fully access this capability. Also, the opportunity costs for more capable developers from delaying repositioning were lower because the simpler consoles only attracted less capable developers. That is, waiting to reposition to a more complex console imposed few opportunity costs for repositioning late in the generation because users who preferred bleeding-edge graphics-intensive games could be attracted based on game performance and realism. Thus, launching a new console generation with simpler DRs attracted one set of developers, which led rival console owners to reposition by offering DRs aimed at attracting a different set of developers.

A reverse case is Sony's PlayStation 2, which offered complex DRs capable of running bleeding-edge, graphics-intensive games. Sony's choice attracted higher-capability developers as early repositioners. As a result, rival console owners who repositioned later had to choose between offering a similarly complex design, which incurred high opportunity costs of repositioning late because first-mover advantage would be ceded to the innovator, and offering a console with simpler DRs that would lower developer adjustment costs and attract a set of developers that were likely lower capability on average.

In sum, an integrated DR-CATO perspective offers a way to interpret and predict rival console owner reactions to innovators in each generation. To be clear, we have not used this combined perspective to provide insight into which console owner innovates first in each generation. The reason is that other factors such as technology advancements in new microprocessors, which are often important to developing new generation consoles, are beyond the scope of this paper. Nonetheless, conditional on the generational innovator emerging, DR-CATO offers a plausible framework with which to understand subsequent industry dynamics.

7. Conclusion

Baldwin and Clark (2000) was a path-breaking book that laid the foundation for linking DRs to various outcomes, including the ones we take up in this paper: strategic repositioning in response to innovation shocks. We aimed to extend Baldwin and Clark's (2000) contribution by linking their notion of DRs with the CATO costs to generate a broader DR-CATO framework. We did so by examining key aspects of the evolution of the video game market, exploring how DR choices affected repositioning decisions through their effects on developers' and rivals' CATO costs.

Our analysis emphasizes that in the video game console market, console DR choices regarding backward compatibility and console openness were critical in shaping console performance largely because of their effects on developers' and rivals' CATO costs. Moreover, console owners often radically changed their consoles' DRs across generations and made significant strategic DR changes within generations. In making these choices, console owners no doubt attempted to take into account the impact of these choices on developers' and rivals' CATO costs.

We maintain that the DR-CATO framework offers a way to deepen our understanding of system-level innovation in modular systems, and to generate new research questions. For example, the literature on modularity has discussed the role of "system integrators"—firms that maintain knowledge about modules that they do not currently produce—in fostering such innovation (e.g., Brusoni and Prencipe, 2001). In what ways might system integrators serve to reduce CATO costs? Do some types of system integrators reduce CATO costs more than others? Do system integrators play different roles in platform ecosystems than in products?

We look forward to future efforts to explore the value of linking Baldwin and Clark's (2000) DRs to the CATO framework. A key task in any such application is to formulate context-specific measures of comparative adjustment costs, transaction costs, and opportunity costs, as we did here, albeit with limited data. Future research will hopefully develop more precise measures based on more extensive data. We expect that such efforts will generate new insights into strategy and industry evolution, and help guide managers' strategic decisions in complex environments.

References

Adner, R. (2017), 'Ecosystem as structure,' *Journal of Management*, **43**(1), 39–58.

Argyres, N., L. Bigelow and J. Nickerson (2015), 'Dominant designs, innovation shocks and the follower's dilemma,' *Strategic Management Journal*, **36**(2), 216–234.

Argyres, N., J. Mahoney and J. Nickerson (2019), 'Strategic responses to shocks: comparative adjustment costs, transaction costs and opportunity costs,' *Strategic Management Journal*, **40**(3), 357–376.

Baldwin, C. (2008), 'Where do transactions come from? Modularity, transactions, and the boundaries of firms,' *Industrial and Corporate Change*, **17**(1), 155–195.

Baldwin, C. (2015), 'Bottlenecks, Modules and Dynamic Capabilities,' *Harvard Business School Working Paper No. 15-028*, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2512209.

Baldwin, C. and K. Clark (2000), *Design Rules, Volume 1: The Power of Modularity*. MIT Press: Cambridge, MA.

Baldwin, C. and E. von Hippel (2011), 'Modelling a paradigm shift: from producer innovation to user and open collaborative innovation,' *Organization Science*, **22**(6), 1399–1417.

Baldwin, C. and C. Woodard (2009), 'The architecture of platforms: a unified view,' *Working Paper*, Harvard Business School Division of Research.

Bigelow, L., J. Nickerson and W.-Y. Park (2019), 'When and how to shift gears: dynamic trade-offs among adjustment, opportunity and transaction costs in response to an innovation shock,' *Strategic Management Journal*, **40**(3), 377–407.

Boudreau, K. and A. Hagiu (2008), 'Platform rules: multi-sided platforms as regulators,' *Working Paper*, Harvard Business School Division of Research.

Brandenburger, A. and H. Stuart (1996), 'Value-based business strategy,' *Journal of Economics and Management Strategy*, **5**, 5–24.

Brusoni, S. and A. Prencipe (2001), 'Unpacking the blackbox of modularity: technologies, products and organizations,' *Industrial and Corporate Change*, **10**(1), 179–205.

Cennamo, C. (2018), 'Building the value of next generation platforms: the paradox of diminishing returns,' *Journal of Management*, **44**(8), 3038–3069.

Cennamo, C., H. Ozalp and T. Kretschmer (2018), 'Platform architecture and quality trade-offs of multihoming complements,' *Information Systems Research*, **29**(2), 461–478.

Cennamo, C. and J. Santalo (2013), 'Platform competition: strategic trade-offs in platform markets,' *Strategic Management Journal*, **34**(11), 1331–1350.

Cennamo, C. and J. Santalo (2019), 'Generative tension and value creation in platform ecosystems,' *Organization Science*, **30**(3), 617–641.

Chesbrough, H. and D. J. Teece (2002), 'Organizing for innovation: when is virtual virtuous?' *Harvard Business Review*, **80**, 127–135.

Corts, K. and M. Lederman (2009), 'Software exclusivity and the scope of indirect network effects in the U.S. home video game market,' *International Journal of Industrial Organization*, **27**(2), 121–136.

de Vaan, M. (2015), 'Interfirm networks in periods of turbulence and stability,' *Research Policy*, **43**(10), 1666–1680.

Evans, D., A. Hagiu and R. Schmalensee (2006), *Invisible Engines: How Software Platforms Drive Innovation and Transform Industries*. MIT Press: Cambridge, MA.

Fang, T., A. Wu and D. Clough (2021), 'Platform diffusion at temporary gatherings: social coordination and system emergence,' *Strategic Management Journal*, **42**(2), 233–272.

Foerderer, J. (2020), 'Interfirm exchange and innovation in platform ecosystems: evidence from Apple's worldwide developers' conference,' *Management Science*, **66**(10), 4772–4782.

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

Hagiu, A. and D. Spulber (2013), 'First-party content and coordination in two-sided markets,' *Management Science*, **59**(4), 933–949.

Jacobides, M., C. Cennamo and A. Gawer (2018), 'Towards a theory of ecosystems,' *Strategic Management Journal*, **39**(8), 2255–2276.

Kapoor, R. and S. Agarwal (2017), 'Sustaining superior performance in business ecosystems: evidence from application software developers in the iOS and Android smartphone ecosystems,' *Organization Science*, **28**(3), 531–551.

Klein, B., K. Crawford and A. Alchian (1978), 'Vertical integration, appropriable quasi-rents, and the competitive contracting process,' *Journal of Law and Economics*, **21**(2), 297–326.

Kotaku. (2014), 'How Much Does It Cost to Make a Big Video Game?' (January 15). http://kotaku.com/how-much-does-it-cost-to-make-a-big-video-game-1501413649.

Kretschmer, T. and J. Claussen (2015), 'When less can be more – setting technology levels in complementary goods markets,' *Research Policy*, **44**(2), 328–399.

Ozalp, H., C. Cennamo and A. Gawer (2018), 'Disruption in platform-based ecosystems,' *Journal of Management Studies*, **55**(7), 1203–1241.

Ozalp, H., J. P. Eggers and F. Malerba (2021), 'Hitting reset: industry evolution, technological change, and the dynamic value of firm experience,' *Working Paper*, University of Amsterdam.

Reimer, J. (2005), 'Cross-Platform Game Development and the Next Generation of Consoles,' arsTechnica. http://arstechnica.com/articles/paedia/hardware/crossplatform.ars.

Rietveld, J. and J. P. Eggers (2018), 'Demand heterogeneity in platform markets: implications for complementors,' *Organization Science*, **29**(2), 304–322.

Rietveld, J., M. Schilling and C. Bellavitis (2019), 'Platform strategy: managing ecosystem value through selective promotion of complements,' *Organization Science*, **30**(6), 1232–1251.

Saadatmand, F., R. Lindgren and U. Schultze (2019), 'Configurations of platform organizations: implications for complementor engagement,' *Research Policy*, **48**(8), 1–17.

Schilling, M. (2003), 'Technological leapfrogging: lessons from the US video game console industry,' *California Management Review*, **45**(3), 6–32.

The Economist (2014), 'Why video games are so expensive to develop,' *The Economist*, 25 September 2014.

Teece, D. J. (1996), 'Firm organization, industrial structure, and technological innovation,' *Journal of Economic Behavior & Organization*, **31**(2), 193–226.

Tschang, F. T. (2007), 'Balancing the tensions between rationalization and creativity in the video games market,' *Organization Science*, **18**(6), 989–1005.

Venkatraman, N. and C.-H. Lee (2004), 'Preferential linkage and network evolution: a conceptual model and empirical test in the U.S. video game sector,' *Academy of Management Journal*, **47**, 876–892.

Williamson, O. E. (1975), *Markets and Hierarchies: Analysis and Anti-Trust Implications*. The Free Press: New York.

Williamson, O. E. (1991), 'Strategizing, economizing and economic organization,' *Strategic Management Journal*, **12**(S2), 75–94.

Zhu, F. and M. Iansiti (2012), 'Entry into platform-based markets,' *Strategic Management Journal*, **33**(1), 88–106.