MICRO-STRATEGIZING IN PLATFORM ECOSYSTEMS: A MULTIPLE CASE STUDY

Completed Research Paper

Ahmad Ghazawneh
Jönköping International Business School
Jönköping, Sweden
ahmad.ghazawneh@ihh.hj.se

Ola Henfridsson
Chalmers University of Technology
Gothenburg, Sweden
ola.henfridsson@chalmers.se

Abstract

The strategy by which a platform owner manages the future trajectory of its platform involves many unknowns. In particular, the ambition to simultaneously control the platform and distribute design capability to users is challenging. While there is an emerging literature on strategy in platform ecosystems, little empirical evidence exists about the series of strategic actions that platform owners conduct to create value in an ecosystem context. Drawing on a strategy-as-process perspective, this paper augments existing platform perspectives by seeking to understand the micro-strategizing of a platform owner. To this end, we report a multiple case study of Apple's use of application programming interfaces for generating value from the iPhone platform. Our comparative analysis identifies and explores five different micro-strategies that can be enacted proactively or reactively: counteracting, monetizing, resourcing, securing, and sustaining. The paper concludes with a number of theoretical and practical implications of these micro-strategies and their interaction.

Keywords: Third-party development, platforms, ecosystems, boundary resources, micro-strategies, APIs.

Introduction

In traditional strategy literature (see e.g., Porter 1980), the capability to strategically position a product relative to an industry's competing forces has been considered vital. Such strategic positioning is typically based on the assumption of clear industry boundaries and players, waiting for being strategized through generic strategies such as cost leadership, product differentiation, or segmentation (Porter 1980). It has been noted that this strategy logic has a significantly weaker ability to explain strategic conduct in dynamic settings, where the competitive environment is turbulent and evolving with actors' strategic moves (Sambamurthy et al. 2003).

Contrasting this traditional strategic positioning logic, recent literature (Evans et al. 2006; Gawer and Cusumano 2008; Messerschmitt and Szyperski 2003; Tapscott and Williams 2006) notes that strategy in a platform context is about building double-sided markets where actors are collaborators as much as they are competitors. This shift in strategy practice diminishes the value of using industry as the basis for strategic analysis and suggests the need of a metaphor that captures the coopetition between platform stakeholders. As an example, consider the simultaneous collaboration and competition of third-party developers and platform owners. Third-party developers contribute ideas, solutions, and an understanding of users' needs that platform owners might not possess (Tiwana et al. 2010). Simultaneously, such developers benefit from having a channel through which they can distribute their applications and services to a large user base (Ghazawneh 2011). Yet, this reciprocity is not static but is dynamic as the relationships between platform owner and application developers are renegotiated in action over time (Eaton et al. 2011; Ghazawneh and Henfridsson 2010). The notion of ecosystem has taken the role of sensitizing this kind of mutual benefit and tension in current thinking about platform strategy (El Sawy et al. 2010; Tiwana et al. 2010; Yoo et al. 2010).

Given that there exists a dissonance between traditional strategic models and the strategic reality of platform owners, there is a need of new knowledge about the role and nature of strategy in platform settings. To this end, recent literature has observed how platform boundary resources have emerged as an important means for enacting a platform strategy in practice (Ghazawneh and Henfridsson 2010). While a platform's technical architecture and organizing principles influence its evolutionary trajectory and differentiation (Tiwana et al. 2010; Yoo et al. 2010), boundary resources such as APIs, SDKs, incentives, intellectual property rights, and agreements serve as tools for the strategizing around a platform. To date, however, little is known about platform owner's use of platform boundary resources for implementing the seemingly minor strategic moves that make up its digital strategy. We refer to these elements of a strategy as micro-strategies. .

The research question addressed in this paper is: *What characterize the micro-strategies used by platform owners in attempts to create and sustain ecosystems through platform boundary resources?* In order to address this research question, we designed a multiple case study (Gerring 2007; Yin 2009) and engaged in detailed empirical analysis of Apple's resourcing their platform ecosystem. The contribution of this paper is the identification of a set of micro-strategies and a tentative understanding of how they play out in platform practice.

The remainder of the paper proceeds as follows. The next section provides a brief overview of the literature on platforms, platform boundary resources, and ecosystems. Following the micro-strategizing perspective that serves as a theoretical basis for this paper, we present the methodology and the results of our multiple case study. The subsequent section identifies and explores five micro-strategies and their implications. The last section concludes the paper.

Related Literature and Conceptual Basis

Platforms and Platform Boundary Resources

We define platforms as a set of interrelated specification layers that support interoperability between the technological modules of a system (Baldwin and Woodard 2009; Franke and von Hippel 2003; Gawer and Cusumano 2008; Morris and Ferguson 1993; West 2003) and provide a set of common resources from which to generate derivative products and services (Robertson and Ulrich 1998). Among software platform owners, it has become increasingly common to nurture communities of third-party developers to build complementary assets to increase the value of the platform. Such assets (e.g., applications and services) promises to extend the software platform's functionality, offer value to the platform's users (Huang et al. 2009), and address heterogeneous end-user needs (Adomavicius et al. 2007; Evans et al. 2006). Harnessing the collective power of third-party developers is enabled by providing specific platform boundary resources (Ghazawneh and Henfridsson 2010) and granting access to complementary development (West 2003).

Located at the interface between the platform owner and third-party developers, we define platform boundary resources as capabilities that facilitate the use of core functionality of a platform in application and service design. Examples of such capabilities are technical boundary resources like APIs and SDKs and social boundary resources like incentives, intellectual property rights, and agreements between the platform owner and third-party developers. Like any boundary object (Bergman et al. 2007; Carlile 2002; Star and Griesemer 1989), platform boundary resources are plastic enough to cut across multiple social worlds by providing enough structure to support several ecosystem parties in their development practices and ambitions (cf. Bergman et al. 2007; Star and Griesemer 1989). In addition, they can maintain a common identity across heterogeneous development settings while still being strongly structured in individual use (cf. Star and Griesemer 1989).

The ability of platform boundary resources to activate the capabilities of a platform makes them important for in everyday platform strategizing (Yoo et al. 2010). They provide the means with which a platform owner can handle the delicate challenge to simultaneously maintain platform control and stimulate third-party development in platform ecosystems (Ghazawneh and Henfridsson 2010).

Shifting Strategic Focus: From Industry to Ecosystem

Software platforms have taken a front-seat in shaping competitive dynamics in settings such as personal computers (Bresnahan and Greenstein 1999), video game consoles (Iansiti and Zhu 2007; Romberg 2007), smartphones (Tiwana et al. 2010; Yoo et al. 2010; Ghazawneh 2011), web systems (Evans et al. 2006; Ghazawneh, 2011), and automobiles (Henfridsson and Lindgren 2010). Recent research has therefore been occupied with developing a novel conceptual machinery that can help us understand this new strategic landscape (El Sawy et al. 2010; Tiwana et al. 2010; Yoo et al. 2010), The relevance of the traditional strategic emphasis on product positioning in pre-defined industries (cf. Porter 1980) dominated by scale and scope economics (Chandler 1990) is diminishing.

To this end, the concept of ecosystem has been adopted as an alternative conception for describing the new arena for innovation. Throughout this paper, we define an ecosystem as a functional unit consisting of a set of actors (e.g., platform owner, third-party developers, platform's partners and users) and a set of technology elements (e.g., software platform, boundary resources) that are mutually interdependent. Centered on the platform and its market for applications and services, these relationships are operated through the exchange of information, resources and artifacts (Jansen et al. 2009). The increasing importance of software ecosystem has triggered researchers to focus on their design (e.g., Gawer and Cusumano 2002), economies and associated business and managerial strategies (see, e.g., Bresnahan and Greenstein 1999; Farrell et al.1998), level of openness and associated criticality (Gawer and Cusumano 2002; Gawer and Henderson, 2007; Eisenmann et al. 2009; West 2003), governance (Ghazawneh and Henfridsson 2010), growth and appropriability (West, 2003) and even considered them as mediating markets (Caillaud and Jullien 2003; Parker and Van Alstyne 2005).

One chief challenge for a platform owner is to stay in control of the ecosystem and simultaneously distribute design capability to third-party developers. The control that a platform owner exercises over the ecosystem and third-party developers is critically affecting the ecosystem growth and success (Parker and Van Alstyne 2008). The main motivation for third-party developers to join a platform ecosystem is to signal compatibility of software applications (Chellappa and Saraf 2010). In doing so, third-party developers gain access to the installed base of the platform ecosystem (Huang et al. 2009). The ecosystem is formed to facilitate such access that fills holes in the product lines of a platform (Baldwin and Clark 2000; Evans et al. 2006). On the other hand, the nature of the ecosystem allows the platform owner to market applications and services that directly compete with ecosystem partners (Huang et al. 2009; Yoo et al. 2010). For example, Apple is providing applications for their iPhone, iPad, and iPod Touch that compete directly with applications provided by third-party developers.

Reviewing the literature, some of the research on platforms and platform ecosystem has indeed focused on the strategic implications of platform thinking. However, little empirical evidence exists about the specific process by which platform owners implement their strategies in the practice of ecosystems. To this end, we now present a micro-strategizing lens to better understand the dynamics of platform ecosystems.

A Micro-Strategizing Perspective

Strategy process research is usually distinguished from strategy content research with reference to its orientation towards *how* a particular firm's strategy emerges, rather than *what* strategic decisions are taken and how they relate to the firm's industrial context (Chia and MacKay 2007; Dess and Lumpkin 2001; Bourgeois 1980). The adoption of a strategy-as-process perspective (Chia and MacKay 2007; Johnson et al. 2003; Jarzabkowski 2008; Whittington 2006) provides a basis for understanding how the specifics of using platform boundary resources in a firm's strategizing around ecosystems.

The strategy-as-process perspective views the implementation of a strategy as a process of organizational becoming (Benson 1977; Orlikowski 2000; Tsoukas and Chia 2002). Similarly, we view the implementation of a platform strategy involving third-party development as process of ecosystem becoming. However, such strategizing is not so much about developing the contents of generic strategies such as cost-leadership, product differentiation, or segmentation (Porter 1980), as it is about dealing with the process of creating, sensing, and responding to emerging issues. In this regard, we view micro-strategizing as a series of seemingly minor strategic moves made by a platform owner in response to actions taken by members of its ecosystem.

Drawing on a strategy-as-process view (Johnson et al. 2003; Whittington 2006), this process can be understood by examining the interplay between three forces: strategy actors, strategic moves, and strategy practices. First, strategy actors refer to those who strategize in the ecosystem. Such actors exist on at least two levels. Strategy actors may be analyzed on a firm level, where specific firms may exercise control over the ecosystem through, e.g., their provision of valuable platform complements, or it might be firm representatives that shape strategy by exercising control over the embedding of strategy in administrative procedures such as allocation of resources (Jarzabkowski 2008). However, various actors accommodate the strategy in view of their current practice and enact their understanding of it to form new practices. Second, the contents of a strategy become material for sensemaking of strategy actors (Weick 1979). This sensemaking necessarily leads to new experiences, which, in turn, shapes the emergent praxis. Strategic moves are what strategy actors do, i.e., “all the various activities involved in the deliberate formulation and implementation of strategy” (Whittington 2006, p.619). Third, strategy practices are those rules and resources that strategy actors draw on in their praxis (Whittington 2006). Apart from the tacit practices that exist in any work setting, explicitly recognized practices exist both on the organization-specific level and on the extra-organizational level. Such practices can be activated by firms to gain advantage, either as a proactive act, or as a reactive act, i.e, as a response to other actors' strategic moves in an ecosystem (cf. Ackoff 1974).

Given this theoretical background, we embarked on a multiple case study with the objective of identifying and characterizing micro-strategies used by platform owners, as a particular strategy actor, to create and sustain ecosystems through platform boundary resources.

Methodology

Research Design

To better understand the micro-strategizing of platform owners, we conducted a multiple, comparative case study (Gerring 2007; Yin 2009) of Apple's iPhone ecosystem with a specific focus on platform boundary resources. A multiple case study method is powerful for conducting cross-case analysis, the extension of theoretical perspectives, and is typically considered yielding more general research results (Benbasat et al. 1987; Eisenhardt, 1989). In addition, analytical generalization logic in a multiple case study is reinforced through theoretical replication logic (Yin 2009) and supports a comparative analysis approach.

There is a chief reason why we embarked on a study of Apple's platform ecosystem and four important APIs. Apple's ecosystem and each of the four studied APIs represent extreme cases (Yin 2009), that is,"a case that is considered to be prototypical or paradigmatic of some phenomenon of interest" (Gerring 2007, P. 101). As Gerring (2007) argue, such cases are useful for theory generation because extremes or ideal types typically define theoretical concepts. Since this study engages in theory-generation rather than theory-testing, prototypical examples of the theoretical concepts are more important than having a case that is representative of the overall population. .

Data Collection

Digital platform ecosystems is an example of an IS research topic that requires much data and a data collection covering multiple actors and their actions. We employed Romano et al's (2003) methodology for analyzing Internet-based data on Apple's use of boundary resources for governing their iPhone ecosystem, as well as the responses to Apple's strategic moves by members of the ecosystem. As Romano et al. (2003) argue, secondary data, available on the Internet, provides a vast and rich data material that exceed well beyond what would be possible to collect through, e.g., an interview study.

We collected secondary data covering the period between January 2007 and February 2011 from nine primary sources (see Table 1).. Multiple data sources are valuable for creating valid generalization forms and constructs through triangulation (Jick 1979), as well as for improving data quality (Creswell 2007; Seale 1999; Soy 1997).

Table 1. Data Sources

Data Sources	Descriptions
Agreements and guidelines	All publically available case documents such as the Registered iPhone developer agreement, iPhone human interface guidelines, and SDK agreements.
Conferences, events, and workshops	Data collected from recorded and online-streamed Apple's conferences and events: - The Apple Worldwide Developers Conference (WWDC): WWDC 2007, WWDC 2008, WWDC 2009 and WWDC 2010. - Apple's SDK events: iPhone SDK 2008, iPhone SDK 2009 and iPhone SDK 2010. - Apple's special events: Rock and Roll event 2009 and Tablet event 2010. - MacWorld Conference & Expo: MacWorld 2007 and MacWorld 2008.
Corporate blog posts	Data from corporate blog posts of Adobe and Skype.
E-mail conversations	22 messages between Apple and developers, Adobe and developers, Apple/Adobe and media, Google and media, Federal Communications Commission (FCC)/ Federal Trade Commission (FTC) and media.
Interviews	One interview with Adobe's CEO, Shantanu Narayen: - 5:31 minutes video recorded interview by Bloomberg Television. Two interviews with Apple's CEO, Steve Jobs - 6 minutes video recorded interview by ABC News. Transcribed interview, by Time magazine.

Online articles	620 articles from multiple online sources: - General magazines, newspapers and journals such as BusinessWeek.com, NYTimes.com and WSJ.com. - Technology-focused magazines and journals such as ComputerWorld.com, MacWorld.com, and TheRegister.co.uk. - Highly profiled Group-edited blogs about technology such as TechCrunch.com, GigaOM.com and Engadget.com. - Highly profiled tech news and analysis websites focusing on Apple news, its products and marketing strategies such as AppleInsider.com, iLounge.com and Roughlydrafted.com
Open letters	One open letter by Apple's CEO, Steve Jobs.
Press releases	- All press releases collected from Apple's online press release library (January 2007 – February 2011). 18 press releases were selected for further analysis. - Developer news and announcements published by Apple at the iPhone Dev Center.
Public government documents	- Three documents from the Federal Communications Commission (FCC) TO Apple, Google and AT&T. - Three response letters from Apple, Google and AT&T to the FCC.

Data Analysis

Using Romano et al.'s (2003) methodology, the data analysis can be described as a four-step process: elicitation, reduction, visualization, and comparative analysis (see Table 2). First, we used the nine data sources to elicit relevant data about the iPhone ecosystem and its APIs. Our initial search queries included keywords such as iPhone API(s), iOS API(s), Apple's platform APIs, and combinations of these keywords. On the basis of this material, we conducted focused coding (Charmaz 2006) on the identified pool of twelve strategic APIs to distinguish four cases that (1) provide the strongest possible inferences on our study, (2) can help in identifying left-out variables, and (3) generalizable enough to apply to cases similar to those under study. The second step of our data analysis reduced the massive material collected. We concentrated on the four selected APIs' cases and performed additional data collection on them with a focus on the key concepts of the micro-strategizing perspective. Using our conceptual basis, we identified major key events, key actors and key strategic moves for each of the four cases. The third step of our data analysis involved visualizing the four case episodes into five distinct micro-strategies. Finally, we conducted a comparative analysis across the cases to identify proactive and reactive elements of the micro-strategies across cases.

Table 2. Data Analysis

Steps	Tasks	Outputs
1. Elicitation	• Elicited data from nine data sources. • Stored the data in a Qualitative Data Analysis (QDA) software system based on the day/month data appeared online. • Identifying a pool of twelve strategic APIs • Initial focused coding	• Research data base • Numerous concepts related to main categories such iOS APIs, Third-party developers, platform ecosystem, and ecosystem strategizing. • Four main APIs were chosen for further investigation.

2. Reduction	• Reduced the massive data material from the broad initial coding categories • Coded the selected data. • Traced the historical process • Applied our conceptual basis including platform boundary resources, platforms ecosystems and micro- strategizing to identify key events, key actors and key strategic moves.	• Timeline of events. • Key actors. • Key strategic moves.
3. Visualization	• Identified and visualized five main micro–strategies: - Case I: 2 micro-strategies. - Case II: 1 micro-strategy. - Case III: 1 micro-strategy. - Case IV: 1 micro-strategy.	• Four case episodes. • Five main micro-strategies.
4. Comparative Analysis	• Compared the character of each micro-strategy across cases.	• Five reactive micro-strategies. • Five proactive micro-strategies.

Findings

Case Setting

Since the release of Apple's iPhone in July 2007, its ecosystem has been growing rapidly in terms of developers, applications, and downloads. For instance, the ecosystem's official distribution channel "the AppStore" has become the largest applications store worldwide. Given this success, it can be considered useful to more closely examine the strategies applied by Apple to create and sustain the ecosystem. (see Table 3).

Table 3. Apple iPhone's Ecosystem Growth

Ecosystem Growth	**2007**	**2008**	**2009**	**2010**	**Feb 2011**
Developers	0	N/A	+80,000	+145,000	N/A
Applications	0	+15,000	+120,000	+300,000	+350,000
Downloads	0	+ 0.5 Billion	+3 Billion	+7 Billion	+10 Billion

A central component of Apple's strategy to grow the ecosystem is the introduction and governance of APIs. An API (Application Programming Interface) is particular set of rules and specifications that an API consumer (third-party developer) can access and make use of the services and resources offered by an API producer (platform owner) that implements and publishes that API (de Souza et al. 2004a; 2004b). In what follows, we describe the strategizing around four important APIs in the iPhone ecosystem: push notification, in-app purchase, voice over IP, and flash APIs.

Case I: Push Notification

The Push Notification API is a functionality that enables third-party developers to maintain a persistent IP connection to iPhone, iPod Touch, and iPad devices. It is used to forward notifications from the servers of third-party developers to installed applications for notifying users about a particular event.

Steve Jobs and the Apple team's announcement of the iPhone's "software roadmap", SDK and set of APIs on March 6, 2008, involved little detail. Given that there already existed concerns among third-party developers about the (un)availability and limitations of some types of APIs, the fact that push notification was not introduced was a major disappointment.

About four months later, on June 9, 2008 at Apple Worldwide Developers Conference (WWDC 2008), the API was finally announced. Scott Forstall, the Senior VP of iOS Software at Apple, highlighted:

> *There has been one feature request that doesn't currently exist in the SDK, that developers have asked for. The request is mainly coming from developers of clients like instant messaging clients, where by the very nature they only get notifications, even when the user isn't currently running the application........ So we absolutely want to solve this problem, the question is How?......We've come up with a far better solution, and that is, we are gonna provide a Push Notification service to all developer... [explaining how it works]... So the Push Notification service, it is a unified push notification service for all developers......this will be available in September, but starting next month we are gonna be seeing developers so you get your hands on it early.*

As promised, one month later, Apple launched the tools needed and distributed the 'Push Notification' API to a selected handful of third-party developers. While Apple had promised that the 'Push notification' API and associated services would be rolled out to its broader third-party developer community in September, 2009, the API was suddenly pulled away from the iPhone firmware with a somewhat cryptic explanation: *"The notification code has been pulled for "further development" inside the company"*. Third-party developers wondered if Apple intended to remove the feature completely, it was just overlooked, or needed additional development before a full redeployment.

A few months later, on March, 17, 2009, Apple explained that the delay was related to a redesign of the server infrastructure for push notifications. Scott Forstall, the Senior VP of iOS Software at Apple, commented:

> *You know we are late on this one. We announced this last year, and we expected to have this up in production by the end of the year, and we didn't. There is a few reasons for this, ... huge number of developers came to us saying how excited they were about push notification and how they are gonna used it in volumes that we hadn't considered. And so we had to completely re-architect the server infrastructure for push notifications, and that's what we've spent this last six months doing. Completely re-architecting it to make it really really scalable .... So now we are good to go.*

This announcement was followed by the initiation of a trial phase [May 18, 2009] where developers were invited to download a pre-release app and started testing the servers. The message on the iPhone Developer forum read:

> *As a developer actively working with iPhone OS 3.0 beta, we would like your help in testing the Apple Push Notification service. We have selected a pre-release version of the Associated Press app for iPhone OS 3.0 to create a high-volume test environment for our servers.*

The 'Push Notification' API and associated services were entirely and officially released at Apple Worldwide Developers Conference (WWDC 2009) on June 8, 2009:

> *We have provided a generic Push Notification service for developers ...... There are three types of notifications you can push: text alerts like you see here, numerical badges for your home screen icon, and custom alerts sounds like you heard. And again this is only a few of more than a 1,000 new APIs that make up the SDK on iPhone OS 3.0, it is an incredible SDK*

Many considered the Push Notification API well developed and extremely simple in concept including safety and platform-agnosticism. In addition, it was free to use by any third-party developer that had applications listed in the iPhone's App Store. The service and the API guard against misuse, the sender, the user's device, and application were all identified and validated to secure an officially authorized process. This had affected users of unlocked iPhones in that they started facing a problem in receiving push notifications.

In March, 2010, Apple hired Rich Dellinger as a senior UI Designer to support the 'Push Notification' API and its associated services, One of the first initiatives Apple took under his leadership was to increase the security of the service. On December 22, 2010 an official email was sent to third-party developers informing them that service would have stronger encryption. The email reads:

> *Dear Developer, On December 22, 2010, the production Apple Push Notification service will begin to use a 2048-bit TLS/SSL certificate that provides a more secure connection between your provider server and the Apple Push Notification service.*

Case II: In-App Purchase

The In-App Purchase API is a functionality that enables third-party developers to sell, e.g., subscriptions and extra content directly from their applications. The launch of the In-App purchase API coincided with the release of iPhone OS 3.0, including a new SDK and a set of 1,000 new APIs. At Apple Worldwide Developers Conference (WWDC 2009) on June 8, 2009, Scott Forstall, the Senior VP of iOS Software at Apple, announced the new API:

> *In-App Purchase allows developers to make financial transactions right from within the app. Now, this unlocks whole new categories of applications… [examples: magazine subscriptions and buying additional games packs]*

At the time, this was seen as perhaps the most important announcement of the new iPhone 3.0. MG Siegler, a well-known tech writer and analyst, stated:

> *I think in-app purchases are potentially the most exciting thing about the new iPhone 3.0 SDK for developers. I believe it will mean a boatload of money for a great many of them as well as Apple, which takes its 30% cut.*

However, one of the main limitations of this API was its availability for only paid apps. Apple explicitly stated that they wanted free apps to be completely free. For many third-party developers, this decision eliminated the plans to sell expansion packs and additional contents after hooking up users with free applications.

A few months later (October 15, 2009), however, Apple sent an email to third-party developers informing them that the In-App purchase API was changed to support free applications. The message promoted the API by encouraging third-party developers to use it in their apps and emphasized its use strategies:

> *In App Purchase is being rapidly adopted by developers in their paid apps. Now you can use In App Purchase in your free apps to sell content, subscriptions, and digital services…….You can also simplify your development by creating a single version of your app that uses In App Purchase to unlock additional functionality, eliminating the need to create Lite versions of your app.*

Apple kept its policy of cutting 30% of each transaction through the In-App purchase API. The firm applied coherent and strict regulations on that API use. In this regard, Apple virtually determined the type of content, functionality, services, and subscriptions available for in-app purchase. Four different types of In-App Purchase were identified, complemented with a set of guidelines for third-party developers.

The release of the In-App Purchase API for both paid and free applications was considered to have a massive impact on applications development and revenue streams. Apple started tightening its control of the AppStore. On February 1, 2011 it was reported that Apple rejected a Sony e-book reader app. Some application developers, including Sony said Apple told them:

> *They can no longer sell content, like e-books, within their apps, or let customers have access to purchases they have made outside the App Store.*

Sony commented on that:

> *Apple is now requiring "in-app" purchasing rather than linking out to our store. That's not what we submitted based on precedent set by other eBook retailers. We're working on a solution.*

Apple spokesperson Trudy Muller replied to Sony's claim:

> *We have not changed our developer terms or guidelines. We are now requiring that if an app offers customers the ability to purchase books outside of the app, that the same option is also available to customers from within the app with in-app purchase.*

Apple's new policy regarding the In-App purchase API is clearly illustrated in Apple's developer guidelines, that were published on September, 2010. Section 11.2, states:

> *Apps utilizing a system other than the In App Purchase API (IAP) to purchase content, functionality, or services in an app will be rejected.*

Following that incident, massively covered by media, Apple launched a new subscriptionservice on February 15, 2011. The new service was available to all publishers of content-based apps on the AppStore, including magazines, newspapers, video, music, etc. Steve Jobs, Apple's CEO, clarified the basics of the new service in a press release:

> *Our philosophy is simple—when Apple brings a new subscriber to the app, Apple earns a 30 percent share; when the publisher brings an existing or new subscriber to the app, the publisher keeps 100 percent and Apple earns nothing. All we require is that, if a publisher is making a subscription offer outside of the app, the same (or better) offer be made inside the app, so that customers can easily subscribe with one-click right in the app.*

While many were debating the consequences and implications of the new API and associated services, the Federal Trade Commission started investigating the ramifications of the charges. This was done with specific focus on consumers, such as children, who may not fully realize that they are spending real money without parents' permission. FTC Chairman Jon Leibowitz, commented [Febrauary 22, 2011] on this matter:

> We fully share your concern that consumers, particularly children, are unlikely to understand the ramifications of these types of purchases ..... Let me assure you we will look closely at the current industry practice with respect to the marketing and delivery of these types of applications.

Case III: Voice over IP

Voice over IP (VoIP) API is a functionality that enables third-party developers to deliver voice communications over the Internet connections of the iPhone, iPad or iPod touch. This allows apps to operate over Wi-Fi and cellular 3G networks.

The availability of the VoIP API and associated services was revealed at Apple's iPhone SDK Event on March 6, 2008. The indication came from Steve Jobs, Apple's CEO. At the press conference, he was clear about what would become a significant limitation::

> *Press (question): will there be a VOIP app? how will partners react?*
> *Steve Jobs (answer): We will only limit VoIP over the cellular network, but it'll be open over WiFi*

In view of this decision, analysts speculated that Apple was protecting their relations with mobile network operators, being afraid that voice apps over cellular networks would threaten their revenue streams.

A year later [March 17, 2009] at the iPhone OS 3.0 special event, the VoIP API over Wi-Fi was exclusively released. Scott Forstall, the Senior VP of iOS Software at Apple, highlighted this:

> *And in-game voice, if you have a game that plays over Wi-Fi, we have built in voice chat API so you can use to add voice into your game.*

After the exclusive and vague release of the API, third-party developers started to design and develop applications based on it. On March 31, 2009 Skype officially released their version of its VoIP application for the iPhone and the iPod touch. This put Skype in a more direct competition with mobile network operators such as AT&T Inc. and Verizon Wireless. Skype reported that the app was downloaded about one million times during the first two days after the release. Peter Parkes, Social Media Communications Lead at Skype, said he was:

> *Confident that it's one of the fastest-downloaded iPhone apps ever*

However, on July 27, 2009, Apple rejected Google's voice application started pulling all Google Voice-enabled applications from the AppStore. A Google spokesperson, commented on that:

> *Apple did not approve the Google Voice application we submitted six weeks ago to the Apple App Store. We will continue to work to bring our services to iPhone users — for example, by taking advantage of advances in mobile browsers.*

A few days later [July 31, 2009], the Federal Communications Commission (FCC) opened an inquiry regarding the rejection of Google Voice. They sent three separate letters to Apple, Google and AT&T. They mainly asked Apple about their rejection of Google's app and other related apps, asked Google to provide more details about their app, and asked AT&T to clarify any engagement on Apple's decision. In addition to other questions related directly to the matter.

Apple responded to the FCC [Aug 21, 2009] by stating that they did not actually reject the app. It was still under review. Apple added that AT&T was not engaged at all in the application approval process:

> *Apple has not rejected the Google Voice application, and continues to study it. The application has not been approved because, as submitted for review, it appears to alter the iPhone's distinctive user experience by replacing the iPhone's core mobile telephone functionality……Apple is acting alone and has not consulted with AT&T about whether or not to approve the Google Voice application.*

Similarly, AT&T responded to the FCC [Aug 21, 2009] by stating that they did not have any role in the application approval process:

> *AT&T had no role in any decision by Apple to not accept the Google Voice application for inclusion in the Apple App Store. AT&T was not asked about the matter by Apple at any time, nor did we offer any view one way or the other.*

The FCC also inquired about the VOIP not being supported over 3G Networks. AT&T responded that they will reconsider their policy and previous stance regarding VOIP over 3G networks:

> *We plan to take a fresh look at possibly authorizing VoIP capabilities on the iPhone for use on AT&T's 3G network. AT&T will promptly update the Commission regarding any such change in its policies.*

During that time, Google continued developing Google Voice for the iPhone's web browser. The application was built on HTML5, a standard that Apple praised and supported. The browser-based app became available on January 26, 2010. As a result, Google bypassed Apple's decisions regarding the Google Voice as a stand-alone app.

Two days later [January 28, 2010] Apple opened up the 3G-based VoIP API to third-party developers, which enabled VoIP apps to run over the 3G Networks. At Apple's Tablet event [January 27, 2010, Steve Jobs, Apple's CEO, announced that the restrictions on VoIP over 3G networks had been removed. In a statement, Apple said:

> *We revised our Program License Agreement in conjunction with our updated Software Development Kit for iPhone, iPod Touch and iPad Apps. Included in this update is the ability for developers to create VoIP apps that utilize cellular networks.*

Following this key action, Apple announced at the iPhone OS 4 Event [April 8, 2010] that the VoIP API started supporting the multitasking and background functionalities. This meant that apps like Skype would be able to receive calls even if it was not in the foreground, run other apps at the same time, and use the double high status bar when being on a call (i.e., making the app much more like a traditional phone).

For the first time, Apple [Sep 9, 2010] released AppStore guidelines for third-party developers. The guidelines document was intended to help third-party developers to understand the app review process and keeping them in line. The guidelines abstractly revealed which applications would not be allowed on the AppStore. Since none of the rules applied to Google Voice, it was resubmitted and Apple accepted the application [Sep 27, 2010] for inclusion in the AppStore as a stand-alone application.

Case IV: Flash

Flash API is a functionality that enables third-party developers to encode their apps and content in Adobe's Flash technology, and run them in the iPhone, iPad or iPod Touch devices.

When Apple shipped its first iPhone device in June 2007, the Safari web browser was the only means for third-party developers to distribute and run their applications. Since the iPhone's debut, the Safari web browser has not been compatible with Flash. As a result, the growing body of third-party developers that ported their apps to Safari had to avoid integrating Flash in their apps.

On March 4, 2008, Steve Jobs clarified that Flash was not suitable for the iPhone, He believed that the desktop version of Flash ran too slowly on the iPhone and that the mobile version of Flash was not functional enough. Jobs added:

> *There's this missing product in the middle*

Apple released the iPhone's software roadmap, SDK and set of APIs [March 6, 2008] without introducing any Flash API. Ryan Stewart, Adobe's chief spokesman for its Internet-based application, commented:

> *No one aside from [Apple's CEO] Steve Jobs has any idea if or when it's coming …. Everyone I talk to doesn't know anything.*

A quick response came from Adobe on March 19, 2008. Shantanu Narayen, Adobe's CEO, pointed out that they were planning to develop a specific Flash tool for iPhone:

> *We believe Flash is synonymous with the Internet experience, and we are committed to bringing Flash to the iPhone …. We have evaluated [the iPhone SDK] and we think we can develop an iPhone Flash player ourselves"*

In a follow-up statement [March 20, 2008]. Adobe said that Apple had to be involved to bring such software to the iPhone. The company said:

> *To bring the full capabilities of Flash to the iPhone Web-browsing experience, we do need to work with Apple beyond and above what is available through the SDK and the current license around it*

With no response coming from Apple, Adobe during their Q2 [June 17, 2008] revealed a Flash for the iPhone running on emulation software. The company clarified:

> *We have a version that's working on the emulation. This is still on the computer and you know, we have to continue to move it from a test environment onto the device and continue to make it work. So we are pleased with the internal progress that we've made to date.*

A few months later [Sep 30, 2008], at the Flash On The Beach (FOTB) conference, Adobe confirmed that the company was working on a solution. However, it insisted that the iPhone was a closed platform, and that this made it uncertain whether their solution would ever be realized on the iPhone platform. In an interview with Bloomberg Television [Feb 2, 2009], Shantanu Narayen, Adobe's CEO said that the company faced a real technical challenge:

> *It's a hard technical challenge, and that's part of the reason Apple and Adobe are collaborating …. The ball is in our court. The onus is on us to deliver.*

While Adobe was struggling with finding a solution for Flash, Apple supported and praised HTML5, which was a web technology that could substitute Flash in web browsing. On February 24, 2009 Apple released Safari 4, a new version of their web browser that built around HTML5. Philip Schiller, Apple's senior vice president of Worldwide Product Marketing, promoted the new Safari version:

> *Safari 4 is the fastest and most efficient browser for Mac and Windows, with great integration of HTML 5 and CSS 3 web standards that enables the next generation of interactive web applications.*

Without Apple support, Adobe announced [October 9, 2009] the inclusion of a 'Packager for iPhone apps' in the Flash developer tool (Creative Suite 5 -CS5). While Flash applications still would not run on the iPhone, the CS5 simply turned Flash applications into iPhone applications automatically.

Prior to the anticipated launch of the CS5 on April 12, 2010, Apple responded [April 8, 2010] to Adobe new move by releasing a new developer license agreement. In this new agreement, there was a passage with major implications for third-party developers, and disastrous consequences for Adobe's anticipated release of Flash CS5. Apple banned applications using intermediary layers. Section 3.3.1 in the iPhone SDK agreement read:

> *Applications may only use Documented APIs in the manner prescribed by Apple and must not use or call any private APIs …. Applications that link to Documented APIs through an intermediary translation or compatibility layer or tool are prohibited.*

Adobe's response was quick [April 20, 2010]. Adobe dropped future support for iPhone development in Flash. Mike Chambers, the Principal Product Manager for developer relations for the Flash Platform at Adobe, stated:

> *We will still be shipping the ability to target the iPhone and iPad in Flash CS5. However, we are not currently planning any additional investments in that feature.*

On April 29, 2010, Steve Jobs wrote an open letter explaining why he refused to support Flash on iPhone, iPads and iPod devices. The letter, titled "Thoughts on Flash", revealed six main reasons behind Apple's position. Besides claiming that Flash was closed and proprietary and had major technical drawbacks (reliability, security and performance, battery life and touchable devices), the most important reason was:

> *Letting a third party layer of software come between the platform and the developer ultimately results in sub-standard apps and hinders the enhancement and progress of the platform … We cannot be at the mercy of a third party deciding if and when they will make our enhancements available to our developers.*

A few days later [May 3, 2010], it was reported [New York Post] that the Department of Justice (DoJ) and Federal Trade Commission (FTC) had started negotiating over which of them would begin an antitrust inquiry into Apple's new policy of third-party developers required only to use Apple's own development tools. Apple moved forward [September 9, 2010] and updated their policy by relaxing all restrictions on development tools as long as the resulting apps do not download any external code. Apple claimed that this move would give the third-party developers the flexibility they wanted, while Apple preserved the security it wanted.

A few hours after Apple's press release was published, Adobe decided to resume their support of Adobe's Packager feature. Adobe welcomed Apple's new policy change:

> *This is great news for developers and we're hearing from our developer community that Packager apps are already being approved for the App Store. We do want to point out that Apple's restriction on Flash content running in the browser on iOS devices remains in place.*

Discussion

Platform ecosystems have emerged as an increasingly powerful conception of software development environments characterized by multiple actors co-creating value in a coopetitive way (El Sawy et al. 2010; Tiwana et al. 2010; Selander et al. 2010). In this paper, we set out to provide an empirically grounded understanding of the micro-strategies used by platform owners in attempts to create and sustain ecosystems through platform boundary resources. Our focus has been on four key APIs used in growing Apple's iPhone platform. In what follows, we first compare and discuss the micro-strategies (counteracting, resourcing, securing, monetizing, and sustaining), strategic moves, and strategy actors that emerged from our case analysis (see Table 4). We then outline implications, and conclude by presenting limitations and possible future research opportunities.

Table 4. Micro-Strategizing across the Four Cases

Case	Micro-Strategizing	Strategic Moves	Actors
	Proactive resourcing	-Released the 'Push Notification' API to selected third-party developers.	

Case I		-Started the testing of the API and the allocated servers. -Officially released the API. -Hired Rich Dellinger (inventor of notification system used in webOS at Palm) as a senior UI Designer. -Enhanced the API with stronger encryption method (*a 2048-bit TLS/SSL certificate*)	-Apple. -Third-party developers.
	Proactive securing	-Filed a patent application for the 'Push Notification' API.	
	Reactive resourcing	-Delayed the release of the API due to "re-architecting the server infrastructure"	
	Reactive securing	-Pulled the API from the iPhone firmware.	
Case II	Proactive monetizing	-Released In-app purchase API for paid apps. -Released In-app purchase API for free apps. -Applied 30% transactions cut.	-Apple. -Sony. -Federal Trade Commission (FTC). -Third-party developers.
	Reactive monetizing	-Applied strict regulations on content, etc. -Tightened the control of the AppStore. -Launched a new 'subscription' service for content publishers.	
Case III	Proactive sustaining	-Released VoIP API over Wi-Fi networks. -Accepted and rejected applications based on partnership/competitor issue. -Published AppStore Guideline for third-party developers (first time ever).	-Apple. -Google. -AT&T. -Skype. -Federal Communications Commission (FCC). -Third-party developers.
	Reactive sustaining	-Released VoIP API over 3G Networks. -VoIP API supporting multitasking and background functionalities	
Case IV	Proactive Counteracting	-Blocked Flash on Safari web browser. -Blocked Flash API from the SDK. -Changed developer license agreement and Interdicts Adobe's "work-around technique". -Supported HTML5 with the intention to substitute Flash on Safari.	-Apple -Adobe -Department of Justice -Federal Trade Commission -Third-party developers
	Reactive Counteracting	- n/a	

Comparing the four cases with the intention to characterize the nature of strategies enacted by Apple, we identified five different strategies that were enacted by Apple through the use of boundary resources:

Resourcing: Our analysis shows that resourcing is an important micro-strategy for a platform owner in ecosystems. We define resourcing as the provision of platform boundary resources for enriching a platform with new capabilities. Resourcing can either be proactive or reactive**.** Proactive resourcing refers to the provision of platform boundary resources for enriching a platform with new capabilities. We observed how Apple initiated a 'proactive resourcing' strategy aiming to enrich the platform with new boundary resources. The release of the 'Push Notification' API provided the platform with new capabilities requested by a huge number of third-party developers. In addition, the release was accompanied with other strategic moves. For instance, Apple improved the capabilities of the 'Push Notification' API, and simultaneously started using a 2048-bit TLS/SSL encryption method to enhance the security of push notification. At the same time, we observed how Apple was resourcing the platform as a response to other actors' strategic moves in the ecosystem. We refer to reactive resourcing as the provision of platform boundary resources for enriching a platform as a response to other actors' strategic moves in an ecosystem. As an example in the push notification case, Apple delayed the release of the API to improve the server infrastructure to the extent to which it would handle the huge number of service-requests by third-party developers using the API.

Securing: Another micro-strategy that emerged from our data analysis was something that we refer to as securing. We refer to securing as acting on other ecosystem members' strategic moves that would risk infringing the platform. Securing can be either proactive or reactive. Securing is proactive if it involves action on the basis of predicted strategic moves by other ecosystem actors As an example, in the push notification case, Apple filed a patent application to protect the intellectual property around the released API. Securing can also be reactive. Reactive securing means acting on realized platform-infringing strategic moves. As an example, pulling the push notification API entirely from the iPhone firmware for some time during the resourcing strategy phase was essential.

Monetizing: Generating new business opportunities through platform boundary resources is another key micro-strategy. We refer to this strategy as monetizing, that is, the act of augmenting ecosystem relationships through platform boundary resources. Such monetizing is proactive when it involves the creation of a new line of business. For instance, Apple released the In-app purchase API and managed to build a source of income by charging 30% of the purchase each time contents were sold through an application in the appstore. Monetizing can also be reactive. In order to response to Sony and other third-party developers' strategic moves, Apple accomplished reactive monetizing by tightening control of the distribution channel AppStore, applying strict regulations on contents sold using the API, and launching a new subscription service for content publishers.

Sustaining: We refer to sustaining as attempts to maintain existing ecosystem relationships through platform boundary resources. A platform owner needs to strategically sustain partnerships and deal with both competitors and authorities over time. Sustaining can be done both proactively and reactively. When done proactively, the platform owner augments ecosystem relationships through platform boundary resources. For instance, Apple dealt with the partnership with AT&T through proactive sustaining as they decided to limit the VoIP API to WiFi networks. This enabled Skype and other third-party developers to resource the platform, while maintaining a useful relationship with the important operator. Sustaining can be reactive too. In response to concerns voiced by FCC, Apple published new guidelines for their distribution channel 'AppStore' enabling them to deal strategically with competitor requests (such as Google's). They also released the VoIP API over 3G networks and supported multitasking.

Counteracting: We define counteracting as taking active measures against foreign meta-platforms and external boundary resources to avoid getting the platform infringed by competitors. Counteracting is usually proactive, although reactive counteracting also can be envisioned. As an example of proactive counteracting, Apple effectively hindered Adobe's Flash boundary resources. This counteracting involved a series of strategic moves. Flash boundary resources were blocked on both web and mobile application versions. A new developer license agreement was developed to interdict Adobe's attempts to bring Flash to the iPhone. In addition, Apple strongly supported HTML5 as an alternative to Flash. Lastly, although our analysis of the four cases did not reveal any evidence, we believe that counteracting can be reactive too. Consider the increasing interest in platform ecosystems in the automotive industry (see e.g., Henfridsson et al. 2009), where open platforms promise to offer an entire range of new functionality for the car driver. In this use setting, it is possible to imagine a case where authorities would demand that an automaker takes measures to reduce the possibilities for outside actors to aligning additional communities and applications.

Implications and Limitations of the Study

There are a number of implications of our research. First, our perspective on micro-strategizing complements and extends the literature on software platforms (Ghazawneh and Henfridsson 2010; Tiwana et al. 2010; West 2003; Yoo et al. 2010) by applying a strategy-as-process view (Chia and MacKay 2007; Johnson et al. 2003; Jarzabkowski 2008; Whittington 2006) to platform ecosystems. Rather than closely analyzing platform strategy as a single and discrete strategy, the perspective proposes how a platform strategy emerges through the enactment of different micro-strategies as the platform owner discover new opportunities or react as a response to strategic moves by ecosystem members. Our multiple case study identified five such strategies: counteracting, monetizing, resourcing, securing, and sustaining. Second, our research provides a perspective on the nature of strategy in the context of platform ecosystems. Rather than strategically position its offer on a clearly defined market or industry (Porter 1980), our study shows how multiple strategies need to be exercised and combined in practice to deal with the 'living' character of ecosystems. Third, the results of our study suggest that platform owners need to devise repertoires of micro-strategies that can be implemented swiftly when needed. The medium for such micro-strategizing is boundary resources, located at the boundaries between the platform owner and actors in the ecosystem. Finally, we contribute to the continued investigation of digital innovation (Eaton et al. 2011; Henfridsson et al. 2009; Svahn et al. 2009; Yoffie 1997; Yoo et al. 2010) by illustrating how micro-strategizing is a useful starting-point for further study of the dynamics of platform ecosystems.

Future studies could address several limitations in our work. First of all, it would be useful to compare our results with investigations of other platform owners' attempts to strategize ecosystems through platform boundary resources. Apple's iPhone ecosystem is an extreme case, which means that it is useful for generating new theoretical insights (Gerring 2007). However, it would be difficult to argue that it is representative. Consequently, studies across platform ecosystems would be valuable to increase our knowledge about the nature of boundary resources in ecosystem strategy. Moreover, it would be useful to engage in theory-testing research for creating variance theories about this topic. This could be done on the entire range of micro-strategies, or for each of the specific micro-strategies.

Conclusion

Software platforms are increasingly becoming the center of gravity of digital innovation. It is therefore not surprising that there is an emerging literature on platform ecosystems in our discipline. Our paper is intended as a contribution to this intellectual debate. As reported in the paper, we learned that platform owners identify, configure, enact, and practice several micro-strategies in attempts to leverage ecosystems through platform boundary resources. Our multiple case study identified and explored five different micro-strategies that can be enacted proactively or reactively: counteracting, monetizing, resourcing, securing, and sustaining. The sheer volume of different micro-strategies suggest that the strategy process in platform ecosystems is characterized by a series of different responses that jointly determine the relative success of a platform owner's efforts to cultivate its ecosystem. As a result, a core competence of a platform owner is continuous monitoring of the ecosystem's evolution to build the capability of timely responses to other actors' strategic moves, and to proactively use boundary resources for increasing the viability and value-creation of the ecosystem.

References

Ackoff, R.L. (1974). Redesigning the Future: A Systems Approach to Societal Problems. New York: Wiley.

Adomavicius, G., Bockstedt, J.C., Gupta, A., and Kauffman, R.J. (2007). "Technology Roles and Paths of Influence in an Ecosystem Model of Technology Evolution," *Information Technology and Management* (8:2), pp. 185- 202.

Baldwin, C., and Kim C., (2000), Design Rules, Vol. 1: The Power of Modularity, Cambridge, MA: MIT Press.

Baldwin, C. and J. Woodard. (2009). The Architecture of Platforms: A Unified View. A. Gawer, ed. Platforms, Markets and Innovation. Edward Elgar, London, UK.

Benbasat, I., Goldstein, D. K., and Mead, M. "The Case Research Strategy in Studies of Information Systems," *MIS Quarterly* (11:3), 1987, pp. 369–386.

Benson, K. (1977). Organizations: A dialectical view. *Administrative Science Quarterly*, 22: 5- 18.

Bergman, M., Lyytinen, K. and Mark, G. "Boundary Object in Design: An Ecological View of Design," Journal of the AIS, (8:11), 2007, PP.546-568.

Bourgeois, L. J., 11 (1980) Strategy and environment: A conceptual integration. Academy of Management Review, 5, 25-39.

Bresnahan, T.F., and Greenstein, S. (1999) "Technological Competition and the Structure of the Computer Industry," *Journal of Industrial Economics* (47:1), pp. 1-40.

Caillaud, B. and B. Jullien. (2003) "Chicken & Egg: Competition Among Intermediation Service Providers." Rand Journal of Economics 34(2).

Carlile, P. "A pragmatic view of knowledge and boundaries: Boundary objects in new product development," *Organization Science*, (13:4), July/August 2002, pp. 442-455.

Chandler, A.D. 1990. *Scale and Scope: The Dynamics of Industrial Capitalism*. Cambridge, MA and London, England: The Belknap Press of Harvard University Press.

Charmaz, K. *Constructing Grounded Theory: A Practical Guide through Qualitative Analysis*. Thousands Oaks, CA: SAGE Publications, 2006.

Chellappa, R., and Saraf, N., (2010) "Alliances, Rivalry, and Firm Performance in Enterprise Systems Software Markets:A Social Network Approach," *Information Systems Research*, (21:4), December, pp. 849–871.

Chia R. and McKay, B. (2007) 'Post-processual Challenges for the Emerging Strategy-as-Practice Perspective: Discovering Strategy in the Logic of Practice', Human Relations, 60(1): 217-242.

Creswell, J. W. *Qualitative inquiry and research design: choosing among five traditions*, 2nd ed., Sage Publications, Thousand Oaks, Calif., IS, 2007.

Creswell, J. W. *Qualitative inquiry and research design: choosing among five traditions*, 2nd ed., Sage Publications, Thousand Oaks, Calif., IS, 2007.

Dess, G. G., and Lumpkin, G. T. (2001). Linking two dimensions of entrepreneurial orientation to firm performance: the moderating role of environment and industry life cycle. Journal of Business Venturing, 16, 429-451.

de Souza, C. R., Redmiles, D., Cheng, L., Millen, D., and Patterson, J. 2004. Sometimes you need to see through walls: a field study of application programming interfaces. In Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (Chicago, Illinois, USA, November 06 - 10, 2004), 63-71.

de Souza, D. F. Redmiles, L.-T. Cheng, D. R. Millen, and J. Patterson, 2004. "How a Good Software Practice Thwarts Collaboration - The Mulitple Roles of APIs in Software Development," in Proc. Foundations of Software Engineering (FSE 2004), Newport Beach, CA, 2004, pp. 221-230.

Eaton, B., S. Elaluf-Calderwood, C. Sørensen, & Y. Yoo (2011): Dynamic Structures of Control and Generativity in Digital Ecosystem Service Innovation: The Cases of the Apple and Google Mobile App Stores. LSE working paper.

Eisenhardt, K. (1989). Building theories from case study research. Academy of Management Review (14:4) 532–550.

Eisenmann, T, G. Parker & M. Van Alstyne (2009) "Opening Platforms: How, When & Why" in Platforms, Markets and Innovation, Ed. A. Gawer, Edward Elgar Publishing.

El Sawy, O., Malhotra, A., Park, Y., and Pavlou, P., (2010) "Research Commentary—Seeking the Configurations of Digital Ecodynamics: It Takes Three to Tango," *Information Systems Research*, (21:4), December, pp. 835-848.

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

Farrell, J., Monroe, H. & Saloner, G. (1998) “The Vertical Organization of Industry: Systems Competition vs Component Competition” Journal of Economics & Management Strategy (7:2), pp 143-82.

Franke, N. and E. von Hippel., (2003). "Satisfying Heterogeneous User Needs via Innovation Toolkits: The Case of Apache Security Software, " Research Policy 32(7) 1199-1215.

Gawer, A., M. Cusumano. 2008. How companies become platform leaders. MIT Sloan Management Rev. 49(2) 28.

Gawer, A. and M. Cusumano. 2002. Platform Leadership: How Intel, Microsoft and Cisco Drive Industry Innovation. Harvard Business School Press, Boston, MA.

Gawer, A. and R. Henderson (2007), ‘Platform owner entry and innovation in complementary markets: Evidence from Intel,’ *Journal of Economics & Management Strategy*, 16, 1-34.

Gerring, J., (2007) Case study research: Principles and practices. Cambridge, UK: Cambridge University Press.

Ghazawneh, A. (2011),‘The power of platforms for software development in open innovation networks’, Int. J. Networking and Virtual Organisations, Vol. 9, No. 2, pp.140–154.

Ghazawneh, A., and Henfridsson. O., 2010. Governing Third-Party Development through Platform Boundary Resources. In *Proceedings of International Conference on Information Systems (ICIS)* 2010, St.Louis, MO.

Henfridsson, O., and Lindgren. R., (2010). “User involvement in developing mobile and temporarily interconnected systems.” *Inform. Systems Journal*. (20:2) 119–135.

Henfridsson, O., Yoo, Y., Svahn, F., (2009). “Path Creation in Digital Innovation: A Multi-Layered Dialectics Perspective,” *Sprouts: Working Papers on Information Systems*, 9(20). http://sprouts.aisnet.org/9-20

Huang, P., Ceccagnoli, M., Forman, C., and Wu, D.J., (2009) "When Do ISVs Join a Platform Ecosystem? Evidence from the Enterprise Software Industry". In Proceedings of International Conference on Information Systems (ICIS), Proceedings. Paper 161.

Hung, A. 2007. Mobile Minute: iPhone APIs are like life - they're full of compromises, available at: http://downloadsquad.switched.com/2007/08/13/mobile-minute-iphone-apis-are-like-life-theyre-full-of-compr/ [retrieved April 03, 2011].

Iansiti, M., and Zhu, F., (2007) "Dynamics of Platform Competition: Exploring the Role of Installed Base, Platform Quality and Consumer Expectations". In Proceedings of International Conference on Information Systems (ICIS) 2007, Proceedings. Paper 38.

Jansen, S, Finkelstein, A, and Brinkkemper. S., (2009). A sense of community: A research agenda for software ecosystems. In 31st International Conference on Software Engineering, New and Emerging Research Track, pages 187-190.

Jarzabkowski, P. 2008. Shaping strategy as a structuration process. Academy of Management Journal (51:4).pp. 621-650.

Jarzabkowski, P. and Wilson, D. C. (2002). ‘Top teams and strategy in a UK university’. *Journal of Management Studies*, 39, 3, 355–81.

Johnson, G. and Huff, A. S. (1998). ‘Everyday innovation everyday strategy’. In Hamel, G., Prahalad, C. K., Thomas, H. and O’Neal, D. O. (Eds), Strategic Flexibility: Managing in a Turbulent Environment. New York: John Wiley & Sons, 13–27.

Johnson, G., L. Melin, and R. Whittington 2003 ‘Micro-strategy and strategizing towards an activity-based view’. Journal of Management Studies 40/1: 3–22.

Jick, T. (1979) Mixing qualitative and quantitative methods: Triangulation in action. Administrative Science Quarterly, 24, 602-61 1.

Messerschmitt, D.G., and Szyperski, C. (2003). “Software Ecosystem: Understanding an Indispensable Technology and Industry,” MIT press.

Meyer, M. H. and Seliger, R., (1998). Product platforms in software development, Sloan Management Review, Fall 1998, 40(1), 61-74.

Morris, C. and Ferguson, C., (1993). “How architecture wins technology wars,” Harvard Business Review, (71:2), pp. 86–96.

Orlikowski, W. (2000) ‘Using technology and constituting structures: A practice lens for studying technology in organizations’. Organization Science 12/4: 404–428.

Parker, G. and M. Van Alstyne (2005). “Two-Sided Network Effects: A Theory of Information Product Design." Management Science Vol. 51, No. 10.

Parker, G. and Van Alstyne, M., (2008) "Managing Platform Ecosystems". In Proceedings of International Conference on Information Systems (ICIS). Paper 53.

Porter, M.E. 1980. *Competitive Strategy: Techniques for Analyzing Industries and Competition*. New York: Free Press.

Robertson, D. and Ulrich, K., (1993). "Planning for product platforms." *Sloan Management Review*, Summer, pp.19–31.

Romano, N. C., Donovan, C., Chen, H., & Nunamaker, J. F. "A methodology for analyzing web-based qualitative data," *Journal of Management Information Systems*, (19:4), 2003, pp. 213-246.

Romberg, T., (2007), Software platforms - how to win the peace, Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS-40), 1-10.

Sambamurthy, V., Bharadwaj, A., and Grover, V. 2003. "Shaping Agility through Digital Options: Reconceptualizing the Role of Information Technology in Contemporary Firms," MIS Quarterly (27:2), pp 237-263.

Seale, C. *The quality of qualitative research*, Sage Publications, London; Thousand Oaks, Calif., 1999.

Selander, L., Henfridsson, O., and and Svahn, F. (2010) "Transforming Ecosystem Relationships in Digital Innovation". In Proceedings of International Conference on Information Systems (ICIS), Proceedings. Paper 138..

Soy, S. K. 1997 "The case study as a research method,"University of Texas, Graduate School of Library and Information Science Website. Available at: http://www.gslis.utexas.edu/~ssoy/usesusers/l391d1b.htm. [retrieved 27 Feb 2011].

Star, S. L. and Griesemer, J. R. "Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39," *Social Studies of Science*, (19:3), August 1989, pp. 387-420.

Svahn, F., Henfridsson, O., and Yoo, Y. "A Threesome Dance of Agency: Mangling the Sociomateriality of Technological Regimes in Digital Innovation," in: *Proceedings of ICIS 2009*. Phoenix, AZ, 2009.

Swanson, E. B. 1994. Information systems innovation among organizations. Management Sci, (40:9), pp. 1069–1092.

Tapscott D. and Williams A.D. "Wikinomics: How Mass Collaboration Changes Everything," New York (N.Y.): Portfolio, 2006.

Taudes A., Feurstein M., and Mild A., (2000) "Options Analysis of Software Platform Decisions: A Case Study," MIS Quarterly, (24:2), June, pp. 227-243.

Teece, D.J. "Profiting from Technological Innovation: Implications for Integration, Collaboration, Licensing and Public Policy," Research Policy, (15), 1986, pp 285-305.

Tiwana, A., Konsynski, B., and Bush, A., (2010) "Research Commentary: Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics," *Information Systems Research*, (21:4), December 2010, pp. 675–687

Tsoukas, H, and Chia R. (2002) 'On organizational becoming: Rethinking organizational change'. Organization Science (13:5), pp.567-582.

Von Hippel, E. and R. Katz, "Shifting Innovation to Users via Toolkits," *Management Science*, (48:7), June 2002, pp. 821-834.

Weick, K. (1979) The social psychology of organizing. Reading, Mass.: Addison-Wesley.

West, J. 2003. "How Open Is Open Enough? Melding Proprietary and Open Source Platform Strategies," *Research Policy* (32), pp 1259-1285.

Whittington, R. (2006). Learning more from failure: Practice and process. *Organization Studies*, (27:12), pp. 1903-1906.

Yin, R.K. *Case Study Research: Design and Methods*, (2009). Thousand Oaks: SAGE Publications.

Yoffie, D.B. "Introduction: Chess and Competing in the Age of Digital Convergence," in: *Competing in the Age of Digital Convergence*, D.B. Yoffie (ed.). Boston, MA: Harvard Business School Press, 1997, pp. 11-35.

Yoo, Y., Lyytinen, K., and Boland, R.J. (2008) "Distributed Innovation in Classes of Networks," Proceedings *of the Fourty-First Annual Hawaii International Conference on System Sciences* (CD-ROM), January 7-9, Computer Society Press.

Yoo, Y., Henfridsson, O., and Lyytinen, K., (2010) "Research Commentary: The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research," *Information Systems Research*, (21:4), December, pp. 724–735.