The London School of Economics and Political Science

The Dynamics of Digital Platform Innovation:

Unfolding the Paradox of Control and Generativity in Apple's iOS

Benjamin David Eaton

A thesis submitted to the Department of Management of the London School of Economics for the degree of Doctor of Philosophy.

London, June 2012

Declaration

I certify that the thesis I have presented for examination for the MPhil/PhD degree of the London School of Economics and Political Science is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it).

I warrant that this authorisation does not, to the best of my belief, infringe the rights of any third party.

I declare that my thesis consists of 71,300 words.

Abstract

Mobile digital platforms provide an architectural basis for third party innovation of platform complements. Platform owners have property rights, enabling them to establish a boundary of permissible innovation demarcating the permitted from the prohibited. This allows for the curation of complements, which provides a means of controlling for value creation. Consequently, the innovation of platform complements is occasionally refused by platform owners. When this occurs tensions may arise between the two parties over where the boundary of permissible innovation should lie. Tussles may break out, embodied in complex interactions, as each party attempts to get its way. Eventually an outcome is achieved, and a platform innovation is either allowed or prohibited.

A body of platform innovation literature is emerging from fields including information systems. Whilst this literature considers many aspects of platform innovation, the dynamics concerning the control of the innovation of platform complements is overlooked. This research attempts to address that gap. Its relevance to information systems concerns the digitalisation of platforms as digital infrastructures, which affects their capacity for innovation and regulation.

This research uses the method of narrative networks to analyse 45 examples of contested platform innovation. This approach, informed by empirical data sourced from over 4500 blog entries, identifies patterned sequences of actions across the examples. These sequences describe how tension builds, how control is asserted, and how control is then resisted. A theory of formal managerial control is used to explain how mechanisms of control are applied by platform owners as well as how developers respond to control.

The principle contribution of this research is to theory. It develops and presents a theory to describe and explain the dynamics of contested innovation of complements on curated digital platforms. In doing so, it challenges the understanding that the platform owner alone controls platform design rules concerning which platform complements are allowed, and which are not. Furthermore, the study opens up the possibility that the forces of digitalisation provide third parties with the power to influence platform architecture, but at the cost of additional means of being controlled.

To my son, Flynn, with love.

Axio era y no era ...

(It was and it was not ...)

A traditional opening to Majorcan folk tales (Jakobson 1960, p. 371).

Acknowledgements

I am grateful to Carsten Sørensen who has done a first class job of supervising my PhD. His support knows no bounds; not really knowing how to express my gratitude, I undertake the following: one day I would like to supervise doctoral students, and I hope that a commitment to nurture them in the way that I have been looked after, reflects my appreciation to him. In addition to Carsten, I would also like to thank both Silvia Elaluf-Calderwood and Youngjin Yoo, with whom I worked for a submission to MISQ. I have learnt much from all of you. I am extremely grateful to my second supervisor, Jannis Kallinikos, who has a knack for revealing and suggesting ideas at just the right time. In addition Edgar Whitley, Shirin Madon, Will Venters, Tony Cornford, Susan Scott and Chrisanthi Avgerou have all provided priceless advice and support. I have enjoyed my time at the LSE in the company of the other PhDs at the Information Systems Group, and in particular I would like to thank Aleksi Aaltonen, Patrizia Bertini, Carla Bonina, Jan Herzhoff, Boyi Li, Attila Marton and Roser Pujadas for their friendship and help.

Further afield, I was lucky enough to attend a number of doctoral consortia where I received much help. I am indebted to Suprateek Sarker and Cathy Urquhart who give me much confidence at the ECIS 2011 doctoral consortium in Helsinki. I am similarly grateful to Søren Askegaard and Benoît Heilbrunn who introduced me to structural semiotics at the 2010 PhD seminars on consumption theory at the University of Southern Denmark in Odense. I thank Richard Boland at the 2009 DESRIST doctoral consortium in Philadelphia who impressed on me the need to find a conundrum to study. Last, but not least, I would like to thank Eusebio Scornavacca and Hans Lehman, both at Victoria University of Wellington, New Zealand for their valuable ideas, time and help.

On a practical level, I am extremely grateful for sponsorship at various times from the Mobile VCE, the Engineering and Physical Sciences Research Council, as well as from the London School of Economics. I would also like to thank Imran Iqbal, Fran White and team who make "The System" work.

I am blessed with loving and supportive parents and good friends, thank you to you all. First amongst them is Sondra Hausner - thank you for your love, encouragement, and introduction to the concept of Fredric Jameson days.

Table of Contents

List of Figures and Tables

Figures

Tables

1 Introduction

In July 2009 Google released Google Voice, a voice over IP application (VoIP app), for both Android and Blackberry smartphone platforms. Shortly after, a number of apps, based on Google Voice and built by small developers independent of Google, appeared on Apple's App Store for download onto the iPhone. Later in July it transpired that Google had attempted to have Google Voice admitted into the App Store, but that Apple had refused its request, claiming that the app duplicated the functionality already present on the device. At the same time Apple removed all other apps based on Google Voice from the App Store. At the end of July the US Federal Communications Commission (FCC) opened an investigation into why Apple had barred Google Voice. In spite of the FCC's interest in their actions, Apple's block of Google Voice still remained in place in January 2010. In its original form, and like all other apps submitted to the App Store, Google Voice was written as a native app. This meant that the code for the application was precompiled for iOS, before being downloaded from the App Store as a binary file and stored locally on the iPhone, to be called upon and used whenever and as often the user required. But at this point Google decided to circumvent Apple's control of the App Store and its control of access to iPhone users, by releasing Google Voice as an HTML5 based Web App for the iPhone. This meant that users could simply access Google Voice functionality over the internet via their iPhone web browser. In September 2010, over a year later, Apple reversed its decision and finally allowed Google Voice into the App Store.

This story illustrates that, on occasion, the process of platform innovation can become convoluted, complicated by both social and technical factors. In some ways Apple seems in control of the process, and in other ways it does not.

As of the 24th March 2012, Apple's market capitalisation has grown to be worth more than $550 billion, making it the most highly valued company on the American stock market[1.1]. Its first quarter profits, at $13.3 billion[1.2], had doubled year on year in 2012 largely driven by sales of the iPhone smartphone and iPad tablet[1.3]. Whilst the direct contribution to Apple's profits made from sales of iPhone and iPad apps is relatively small, estimated at between $250 million and $500 million in the first quarter of 2012[1.4], the indirect contribution is significant. The functionality derived from these small packages of native code, typically developed by third parties and distributed from Apple's App Store, adds significant user value to iOS devices, which in turn contributes to the appeal of the iPhone and the iPad. The value generated is such that its competitors have followed Apple's lead and promoted equivalents of the App Store and the third party development of apps for their own platforms.

Meanwhile, the former market leader Nokia, has seen its share of the smartphone market fall from just under 40% in Q1 2010 to just under 15% in Q3 2011[1.5]. Nokia's decline was so rapid that it prompted the then newly appointed CEO Stephen Elop to release a 1300 word memo to Nokia employees containing the following passage[1.6]:

"I have learned that we are standing on a burning platform. And, we have more than one explosion - we have multiple points of scorching heat that are fuelling a blazing fire around us. [...] Apple disrupted the market by redefining the smartphone and attracting developers to a closed, but very powerful ecosystem. In 2008, Apple's market share in the $300+ price range was 25 per cent; by 2010 it escalated to 61 per cent. [...]Apple demonstrated that if designed well, consumers would buy a high-priced phone with a great experience and developers would build applications. They changed the game, and today, Apple owns the high-end range."

It would seem, at least to practitioners, that the establishment and management of a mobile platform that attracts innovation by third party developers is of commercial importance.

However the question that platform owners face, concerns more than how to obtain third party innovation by developers in terms of content and apps. It concerns how to obtain the *right kind of* third party innovation that will attract users to their platform. There may be numerous factors that interact and contribute to building compelling third party innovation on a platform. These may include an installed base of users who are prepared to pay for content and applications, the commercial terms on which developers get paid for their innovation, the functional capabilities of the platform and device, and the ease of developing content for a platform. Another factor concerns how open or closed a platform is to admitting innovation by others, namely, how discriminating it is of the content and apps that it makes available to end users, or the degree to which the owner of a platform controls the nature of allowable third party innovation.

The degree of openness or closure to admitting the innovations of third parties varies amongst platforms. Some, such as the walled garden model previously adopted by many mobile operators were closed. The owners of these platforms were highly selective of the limited content and applications that they allowed for distribution over their networks. What little content that was allowed onto these platforms did not appeal to users, and this may have contributed to their lack of success. Others, such as Google's Android platform, are open. These platforms admit all content and apps that are submitted to them, and only apply control by removing those that are found to be malicious post hoc. That there are now over 300,000 apps on Google's Android App Market[1.7] which have been downloaded over 10 billion times[1.8] bares testament to the popularity of the platform for both the developers and consumers of its apps. However, there are some[1.9] who claim that this laissez-faire approach to the control of platform innovation comes at the cost of low levels of quality amongst content and apps. Apple, on the other hand, selects and curates amongst the apps that are submitted, before allowing them on to their platform. Their platform is closed to the extent that they specify what is not allowed onto their platform. Yet they are largely open with regards to everything else, albeit with the possibility that they are free to specify new controls at any time. Apple's model of platform innovation requires the organisation to make decisions as to what apps should be allowed onto the iOS platform as innovation, as and when they are submitted by third party developers. To that end Apple employ an App Approval Process[1.10] to facilitate these decisions. This process has been employed at least half a million times, equivalent to the number of apps admitted into Apple's App Store[1.11].

Given the quantity of applications that have gone through the App Approval Process it would seem that most of the decisions have been relatively straightforward: an app is accepted into the App Store or it is rejected. However, as the example of Google Voice shows, there are cases where the process of platform innovation can become complex and dynamic, with multiple interactions between platform owner, developer and other parties. This dissertation uses empirical examples of complex platform innovation, such as Google Voice, in order to explain how platform innovation can come about as a result of tensions between platform owner and third party developers, as they seek to negotiate the boundaries of permissible innovation. In doing so the dissertation challenges a predominant view concerning the control of platform architecture.

1.1 Motivation and scope of research

Mobile platforms such as iOS are a type of industry platform, which Gawer (2009a, p45) defines as:

"building blocks (they can be products, services or technologies) that act as a foundation upon which an array of firms (sometimes called a business ecosystem) can develop complementary products, technologies or services."

This definition is informed by the same research into industrial innovation processes (Abernathy and Utterback 1978; Baldwin and Clark 2000; Clark 1985), that forms the basis of much of the platform research in reference disciplines (Baldwin and Woodard 2009), and which has historically tended to focus on hardware rather than software. This stream of literature is largely concerned with enabling architectural design with an emphasis on concepts such as modularity, in order to enable innovation, and design rules governing what can be innovated through centralised control. The literature concerning platform innovation has built on these concepts, whilst exploring the capacity of platforms for enabling decentralised innovation (Gawer and Cusumano 2002). With this in mind, scholars such as Baldwin and Woodard (2009) consider platforms to represent a set of modular components with low variety, which make up one part of a wider system based on a modular architecture. The second part consists of a set of platform complements, which consists of a further set of modular component that draw on the functionality of platform components. In contrast to platform components, the modular components of complements display high variety and can change on a regular basis. The assumption in the literature (Baldwin and Woodard 2009) is that platform owners possess decision rights over the architectural arrangements of the platform and the functionality that they offer up to complementary modules. However, third party developers of platform complements may be freed, depending upon architectural constraints, to innovate and arrange functionality within their modules as they wish (Gawer 2009a).

The platform owner lies mediates groups of consumers and producers of platform complements in a two sided market (Rochet and Tirole 2003). Platform owners hold property rights (Hart and Moore 1990) over a platform, and are consequently able to profit from transactions that their platform mediates. Consequently it is often in their interests to act as a "public regulator" (Farrell and Katz 2000) in order to provide governance to maintain the health and grow the platform ecosystem. Fortunately, two sided markets, which rely on complementary goods such as industry platforms are subject to network externalities (Katz and Shapiro 1986). In this way an installed base of consumers attracts producers of platform complementors, and if the complements are attractive, vice versa. A strategy that some platform owners pursue in order to retain existing consumers and attract new ones is to maintain both the pace and quality of the innovation of platform complements. This requires a balance of keeping a platform open and closed (Boudreau 2010; West 2003) for complementary innovation. Numerous strategies exist for attracting developers onto an open platform (Schilling 2003) including financial incentives, such as subsidisation, and easing the process of development through extensive developer support. Platform owners are able to police the quality of complements using their platforms, and apply "bouncer's rights" (Strahilevitz 2006) to eject or block entry to any innovations that fall short.

The innovation of platform complements on platforms such as Apple's iOS and Google's Android is manifested in the form of apps and platform enablers. Apps are small software applications, typically between 100kb and 10Mb in size, which are typically executed on mobile devices, such as smartphones and tablets, and provide users with hedonic or functional utility. The functions that apps are designed for is wide and varied. They include: Content (Entertainment, Lifestyle, Reference, News & Weather); Content Players (Internet Browsers, Media Viewers); Search; Games; Interaction (Phone, Messaging, Social Networking); and Tools (Device Specific, Information Management, Practical, Fun). Currently, the majority of smartphone apps come in two forms. The first concern native apps, where the code is precompiled for the device's operating system, before being downloaded and stored locally on the device, after which it is called upon and run whenever and as often the user requires. The second consist of web apps, which consist of cross platform code, such as Java and HTML5. Web apps are downloaded from the internet to the device and interpreted and run locally each and every time the user wishes to access their functionality. The principle focus within this dissertation is on the innovation of native apps rather than of web apps. Similarly, platform enablers are third party software supporting the development and maintenance of apps. Examples of these include Adobe Flash Player, which enables apps to play media in Flash format, and Adobe Flash Builder, which is a development environment enabling the compiling of code into native code across different platforms such as iOS

Moreover, platforms such as iOS are digital platforms, a particular form of digital infrastructure (Tilson et al 2010) that is configured as an industry platform and that makes available digitalised components (hardware, operating system, SDKs, APIs, and application delivery mechanisms) which act as a foundation upon which developers can build complementary services. They have therefore gone through socio-technical processes of digitalisation (Tilson et al. 2010) and share the characteristics of digital artefacts (Faulkner and Runde 2010). As boundary objects, digitalised artefacts open up new possibilities for enabling and constraining innovation (Ghazawneh and Henfridsson 2012). A digital platform is part of a complex assemblage of physical and digital elements arranged as in a layered modular architecture (Yoo et al. 2010a). Unlike in a traditional modular architecture (Baldwin and Clark 2000; Ulrich 1995), the component modules (for example apps) of a layered modular architecture are not pre-specified at the time of the architecture's design, they can be designed independently of the owner at a later date. Furthermore the openness of interfaces between the layers of this type of architecture is such that the designer of a module is free to design for multiple platforms. When this is combined with the inherent generative capacity of digitalised artefacts "to produce unprompted change driven by large, varied, and uncoordinated audiences" (Zittrain 2006, p. 1980) the potential for the innovation of platform complements is further enhanced. However, with this increased capacity for innovation that arises through digitalisation, comes the possibility of increased levels of undesirable innovation. Undesirable digital platform innovation can manifest itself in the form of poor quality apps that reduce the desirability of a platform, malware and platform enablers that challenge the platform owner's control of an ecosystem. To counter this, there is the possibility that digitalisation might facilitate controlling for the innovation of undesirable complements, through an increased ability to scan and filter digital content. Both of these constructive and constraining forces are compounded by boundary resources made available by owners of digital platforms. Technical boundary resources such as APIs and SDKs, as well as social boundary resources such as incentives, intellectual property rights and processes of control can resource and secure the innovation of platform complements respectively (Ghazawneh and Henfridsson 2012; Yoo et al. 2010a).

If invention can broadly be defined as "the first occurrence of an idea for a new product or process" (Fagerberg et al. 2006, p. 4) then innovation can broadly be thought of as "the first attempt to carry it out in practice" (Fagerberg et al. 2006, p. 4). In an organisational context, the process of innovation describes the journey from invention, development and implementation of new products, programs, services or administrative arrangements (Van de Ven et al. 1999). Innovation processes are highly contextual. Whilst some can pass through orderly stable and sequential phases (Gordon and Greenspan 1988), most seem to have an indeterminate nature (Tushman and Anderson 1986) and are nonlinear and dynamic (Van de Ven et al. 1999).

The empirical investigation, presented in this thesis, is concerned with an aspect of the innovation of platform complements on digital platforms. The focus of the study is not on the overall process concerning the innovation of platform complements per se. Rather it emphasises a series of events that occur at a specific stage of the process of innovation. The focus is the implementation stage of innovation (Van de Ven et al. 1999). This is the point at which an innovation may be commercialised, it is launched into the market and adoption begins.

There are some arrangements of digital platforms, of which iOS is one, where the platform owner curates the platform complements which are made available to consumers. This involves the platform owner agreeing to the commercial implementation of some platform complements, and refusing the implementation of others. This is generally done to maintain the overall quality and value of the platform. This ability to curate platform complements depends on the capacity to selectively open and close a platform to producers of complements. In practice this is often achieved in two ways. The first concerns the monopolisation of the channel distributing platform complements to consumers. The second, concerns the platform owner selectively approving the admission of complements onto a platform and from there into the distribution channel. The platform owner must therefore make a decision as to whether a new platform complement is suitable for distribution to consumers. This decision takes place at the implementation stage of an innovation process, and often entails a series of events, which forms a boundary resourcing process of its own (Ghazawneh and Henfridsson 2012).

This process, which is referred to in this thesis as the “control of platform innovation”, is exemplified by Apple's App Approval Process (Apple 2012). This process is the means by which developers submit, and Apple approves, new apps into Apple's iOS platform ecosystem for distribution from Apple's App Store. The process, illustrated in figure 1.1, is broadly structured as follows: first, a developer enrols in the Apple Developer Program; second, the app is created; third, the app is submitted, with supporting information, to Apple via "iTunes Connect", which is a web-based tool that mediates the relationship between the developer and Apple; fourth, the app is then reviewed by Apple, who decide whether it should be accepted on to the platform, or whether it should be rejected; fifth, the developer is notified of the decision and status of the app via "iTunes Connect".

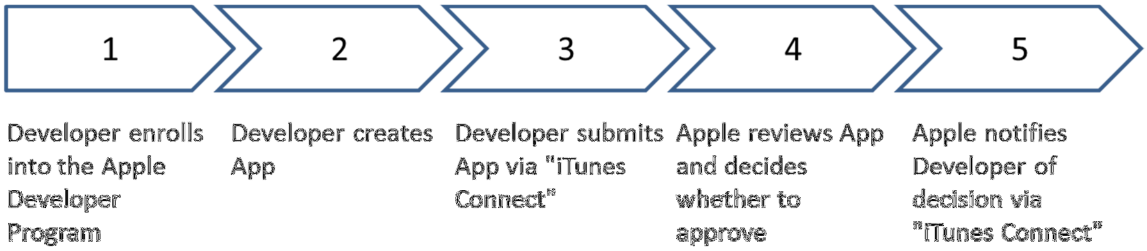

1
2
3
4
5
Developer enrolls into the Apple Developer Program
Developer creates App
Developer submits App via "iTunes Connect"
Apple reviews App and decides whether to approve
Apple notifies Developer of decision via "iTunes Connect"

Figure 1.1 Apple's app approval process

(Based on Apple (2012))

The control of platform innovation is of interest because the outcome of this process can lead to a tension which sometimes erupts into conflict. There are occasions when a potential platform complement, in the form of a new app or platform enabler, is rejected by the platform owner. Tensions may emerge between the two parties as they have opposing interests. On the one hand, developers are motivated to get their intended platform complement, accepted and implemented by the platform owner as an innovation. On the other hand, the platform owner is attempting to control whether the potential complement should or should not be distributed from its app store. As was seen in the earlier example concerning Google Voice, the tension following the rejection of a platform complement can turn into a conflict over contested innovation. This manifests itself in a series of complex interactions between the two parties as they seek to establish where a boundary of permissible innovation lies, governing what can or cannot be innovated as a valid platform complement. In this way, the tension is paradoxical (Tilson et al. 2010) as it transcends a situation where forces that encourage innovation through generative capacity are opposed by forces discouraging innovation. Common sense would suggest that the outcome of this tension would be a reduced level of innovation of platform complements. However the outcome is paradoxical because, in certain circumstances, this tension leads to an increased capacity for platform innovation.

The conflict and outcomes that come about as a result of this tension are of interest, particularly with respect to the nature of control in the innovation of platform complements. The implicit assumption throughout this stream of literature is that the division of activities and responsibilities between actors involved in innovation is both stable and controllable (Baldwin and Clark 2000; Baldwin and Woodard 2009; Gawer 2009a). It follows on from this, that the platform owner is the ultimate authority with regards to design rules (Baldwin and Clark 2000) which guide decisions such as where the boundary of permissible innovation lies on a platform. In short, the platform owner is in charge and in control. However, the earlier example of Google Voice would indicate that a platform owner's authority can be challenged, and that it is possible for a developer to influence what complements are allowed onto a platform. This leads on to as simple question: "How can this be?"

The notion of a tension between control and generativity within digital platforms is encountered across the growing body of information systems research into platforms (Eaton et al. 2012; Ghazawneh and Henfridsson 2012; Tilson et al. 2010; Tilson et al. 2012; Tiwana et al. 2010). Furthermore, the idea that other platform ecosystem members can challenge a platform owners' assumed control over platform innovation has also surfaced in previous IS platform research (Tilson et al. 2012). To this end, studies have begun to emerge which provide empirical evidence (Eaton et al. 2012) describing the conflict that can arise between platform owners and developers in instances of contested innovation concerning platform complements. There is, however, a need for more empirical evidence with regards to describing and explaining the dynamics concerning the control of innovation on digital platforms. Such studies are needed to provide more insight into the tension that arises between control and generativity. There is a specific need not just to describe the complex interactions that occur in instances of contested platform innovation, but also to explain them.

1.2 Problem statement and research question

The previous discussion summarises an understanding of the innovation of platform complements as being contested, and relates it to previous research before finding that it does not seem to be well understood or explained. Following on from this, concepts that were introduced, and that play an essential part in this thesis are summarised in table 1.1. Whilst many of these concepts are defined in more detail in the literature review, it serves to present them here in order to frame a problem statement and set of research questions. Furthermore, it must be noted that these definitions are constrained to the empirical study. Consequently, some definitions, such as the focus of platform innovation on the innovation of platform complements, are necessarily narrow, and they are later opened up in the discussion.

Concept	**Definition**	**Representative references informing conceptualisation**
Digital Platform	Digital infrastructures that are configured as industry platforms and that make available digitalised components (hardware, operating system, SDKs, APIs, and application delivery mechanisms) so as to act as a foundation upon which developers can build complementary services.	Gawer 2009a Tilson 2010,2012 Yoo et al 2010 Eaton et al 2011
Platform complements	"Executable pieces of software that are offered as applications, services, or systems to end-users of a digital platform."	Ghazawneh & Henfridsson 2012
Generativity	The capacity of a digital platform to facilitate the unprompted creation of potentially novel platform complements by a varied and independent audience.	Zittrain 2009 Tilson et al 2010
Platform Innovation	The commercial implementation of a new platform complement.	Gawer 2008 Van de Ven 1999
Control of innovation	Control exercised by a platform owner concerning which platform complements can be commercially implemented on and distributed from a digital platform.	Baldwin and Clark 2000 Baldwin and Woodard 2009
Curated Platform	A type of digital platform where the platform owner actively controls which complements can be innovated on and distributed from the platform.	Zittrain 2012

Table 1.1 Definitions of core concepts used within this dissertation

The problem statement that informs the research questions addressed in this dissertation is:

"The literature concerning platform innovation regards the authority of the platform owner as absolute when determining architectural issues, such as who can interact with or modify which platform components in what ways. Consequently, and following a platform owner's decision regarding whether a potential platform complement should be permitted or not, it is assumed that the unfolding of events should be minimal. Either the new platform complement is admitted into the platform ecosystem, or it is barred. However, there is empirical evidence to suggest that this is not the case. In certain circumstances, this process controlling the innovation of platform complements appears to be complex and dynamic. Furthermore these instances provide evidence that the boundaries of permissible innovation may be negotiated between platform owner and third party developers. This would bring into question the absolute authority of the platform owner concerning these issues."

Following on from this a research question and two sub questions can be formulated to address the problem statement. The overarching research question is concerned with establishing three things. First, it needs to identify that there is a dynamic and contested nature to the control of the innovation of complements on digital platforms. Second, assuming that there is, it should describe this dynamic and contested nature. Finally, assuming that it can be described, it must explain the nature of contested innovation. On this basis, the research question that this dissertation attempts to address is:

What is the nature of control in digital platform innovation?

As seen in the example of Google Voice on the iOS platform, it appears that tensions between platform owner and developers over the boundary of permissible innovation can lead to an apparently complex set of interactions. The question arises as to whether these complex interactions can be simplified, described and understood. Furthermore, can other instances of contested platform innovation be found, and if so can they be simplified and understood too. If many instances of contested innovation can be found, is it possible to find patterns amongst them. If so, does it become possible to abstract and to produce a generalised description and understanding of the dynamics of the control of innovation. These are addressed in the first sub question:

How do the dynamics of the control of innovation change over time?

Assuming that common patterns of interactions can be found across different instances of contested platform innovation as the boundary of permissible innovation is negotiated, then further questions can be asked: can reasonable mechanisms or explanations be found for the events that occur as these common patterns of interactions unfold; is it possible to explain the way that they unfold; and finally, how is it possible for the dynamics of the control of innovation to change over time. These concerns are addressed in the second sub question:

How is control of innovation asserted and lost on digital platforms?

1.3 Research objective and approach

The overall objective of this dissertation is to develop a theory that describes, explains and enhances an understanding of the dynamics of control in platform innovation. As a type 2 theory (Gregor 2006) it goes beyond describing a phenomenon (type 1 theory), but does not go so far as to provide prediction (type 3 theory) regarding future instantiations of the phenomenon.

The theory developed in this dissertation is a low level process theory. It is low level as it seeks explanations for how and things happen in a real world but bounded situation. It is a process theory because it explains the sequential order, in which a discrete set of events unfold based on a group of stories (Huber and Van de Ven 1995, p. vii) concerning instances of contested platform innovation.

Empirical data is collected concerning multiple examples of contested digital platform innovation. These examples are expressed as stories involving complex sequences of interactions between platform owner and third party developers as they conflict over whether developed code should be allowed as innovation. This data is contained within Tech Blogs which are rich in narrative detail regarding examples of these tensions. The Tech Blogs, and the data they contain, are sourced from internet based blog aggregators (Davidson and Vaast 2009).

The first stage of the theory development is to establish and describe the phenomenon as an abstracted set of patterned sequences of actions. Stories found within the data set are analysed in order to produce a simple classification of the different actions employed by: the developers attempting to implement innovation; and the platform owner seeking to control innovation. This is facilitated by using a semiotic analysis to produce semiotic squares (Greimas and Rastier 1968) describing a compact set of actions. This classification is applied across the different stories concerning contested innovation in order to produce an abstracted set of patterned sequences of actions describing a tension. Narrative analysis is applied by means of narrative networks (Pentland and Feldman 2007). This enables an analysis of the different phases of interaction within the sequences to help produce a taxonomy of the generalised dynamics of the control of innovation.

The second stage of the theory development concerns an analysis of the mechanisms of control that are employed by the platform owner, and of the responses that are deployed by the developers. This is done in order to explain and understand the dynamics of the control of platform innovation. Theories of organisational control (Ouchi 1977; Ouchi 1978; Ouchi 1979) are used to explain how control is asserted and lost, and to help interpret why the sequences of actions take the course that they do.

The theory that is developed is constrained to describing, explaining and understanding the nature of the control of innovation within many, if not most, of the examples taken from the particular digital platform ecosystem that is studied. It is not intended to predict with any precision what might happen across other examples of the phenomenon in the same or alternative digital platform ecosystems. Furthermore it does not claim to be exhaustive in terms of identifying and analysing all the factors which may contribute to the phenomenon. To an extent this is because what is studied is part of an open system and there are most likely to be extraneous factors which have not been considered.

1.4 Targets for contributions

By achieving my research objective I believe it may be possible to contribute to the following areas.

The first area of contribution concerns that of theory. The understanding of the process that governs the control of innovation in digital platforms is not well understood in the information systems literature (Tiwana et al. 2010). This dissertation will use empirical evidence to challenge the accepted understanding of how the innovation of complements is controlled in order to develop a theory contested innovation of complements on digital platforms.

The second area of potential contribution concerns method. A different approach is needed for researching into issues the dynamics of the innovation of platform complements on digital platforms. A key concern is that the phenomenon is distributed across organisational boundaries and stakeholders. This issue makes the study of sequences of actions more challenging than within the boundaries of an organisation. It is possible that the methodology developed to elicit empirical evidence may produce a contribution in its own right.

Finally, the third area of potential contribution concerns practice. Digital platforms and their associated ecosystems are becoming an important source of direct and indirect value generation. Whilst there may be numerous factors that contribute to a successful digital platform ecosystem, the management of innovation of content and applications, which is the balance of controlling and encouraging third party innovation, is essential. This dissertation will describe in detail how, at the time of writing, one of the most successful digital platform owners manages that function. It is hoped that the insights generated as the theory is developed may make a contribution to practice.

The actual contributions that are made in the course of this research are presented in the conclusion and are measured against these targets.

1.5 Structure of the dissertation

Following this chapter of introduction, the literature concerning platform innovation is reviewed. The focus of this review is on research that has been undertaken with respect to the structure and architecture of platforms, platform economics and management strategies, governance and the open and closed nature of platforms, as well as the digital characteristics of software platforms. The literature review is then then followed by a chapter concerning the research methodology employed in this research. Given the research objective of developing a process theory, the methodology used in this research is concerned with capturing sequence data and processing it so that it can be interpreted and described as sequences of actions. Sequence methods based on narrative networks (Pentland and Feldman 2007) are employed for this. The thesis then proceeds to describe a theoretical framework based on formal organisational control (Ouchi 1977; Ouchi 1978; Ouchi 1979) which is used as a basis for describing mechanisms of control identified within the data. The fifth chapter is concerned with presenting the results of the empirical study and analysing them. This chapter is divided into two. The first part is concerned with describing sequences of contested innovation identified from within the empirical data using narrative networks. The second part is concerned with explaining the progression of these sequences by means of mechanisms of control based on the theory of formal control. The sixth chapter is concerned with discussing the theoretical implications of the empirical results. It starts by presenting a tentative theory of the control of platform innovation, and the proceeds to use it as basis for unravelling the paradox of control and generativity that can be found on digital platforms. The thesis closes with a conclusion which summarises this research, suggests both its contributions and its failings, before closing with suggestions for future research.

2 Literature Review

Platforms have emerged as an important concept within industry and the phenomenon has attracted increasing academic interest over recent years (Gawer 2009a; Gawer and Cusumano 2002). As platforms have evolved within industry, three different types have emerged: platforms bounded inside firms; platforms supporting supply chains; and platforms supporting whole industries (Gawer 2009a). Platforms bounded inside firms are typically concerned with enabling key components to be shared amongst a product family in order to achieve speed and flexibility in developing product variety, or mass customisation, at lower costs. Supply chain platforms effectively extend the goals of company specific platforms across a whole supply chain. Finally, industry platforms consist of networks of organisations which develop and market products that function together in a technical system. Industry platforms represent a looser arrangement than the other platform types, as the organisations involved do not necessarily enter commercial relationships with each other.

Prior to the recent interest that information systems has taken in platforms, research into this phenomenon has historically occurred within three academic areas, those of product development, technology strategy and industrial economics (Baldwin and Woodard 2009; Parker and Alstyne 2010).

Of the three typologies of platforms it is industry platforms that are most relevant to the research that is undertaken in this thesis. This is on account of what the unique structure of industry platforms enables. Industry platforms are defined by Gawer (2009a, p45) as:

"Industry platforms are building blocks (they can be products, services or technologies) that act as a foundation upon which an array of firms (sometimes called a business ecosystem) can develop complementary products, technologies or services."

Platforms of the type studied in this research mediate two sided networks, consisting of two distinct user groups whose members play different roles (Parker and Van Alstyne 2005; Rochet and Tirole 2003). On the one side are demand side platform users, or consumers that might purchase and consume smartphone applications. On the other, there are supply side platform users who offer platform complements, for example iOS apps, which are consumed by demand side users. Digital platforms such as Apple iOS, Microsoft Windows, Visa, Sony PlayStation, Facebook and Google Android encompass both a set of modular components and a set of rules employed in user transactions that they mediate (Boudreau and Hagiu 2009). These modular components are organised within an architecture that specifies how they all fit together (Henderson and Clark 1990). Rules determine how the users activities are coordinated (Baldwin and Clark 2000) in terms of what information can be exchanged, policies to constrain behaviour and contracts that specify the rights and responsibilities of participants.

The platform at the centre of the two sided market is created and maintained by the platform provider and sponsor, who may or may not be the same entity (Eisemann et al. 2009). Platform providers mediate the users' transactions, and serve as their primary point of contact with the platform. Platform sponsors, as owners of the platform, hold property rights over a platform, which gives them certain rights including: the right to modify a platform's technology; right to determine the design of the platform's components and rules; and the right to determine who may or may not participate in the within the two sided market (Eisemann et al. 2009). For the purposes of the research described in this thesis it is assumed that the platform provider and sponsor are the same entity, which will be termed the platform owner.

The structure of industry platforms is such that platform owners are able to capture value from external complementary innovation (Gawer 2009a). Producers and suppliers of complements capture value through transactions with the installed base of consumers using a platform, and from the network effects derived from complementary innovation (Katz 1994).

This literature review expands upon themes hinted at in this introduction, and that are either directly relevant or provide vital background to the constructs used in the research presented in this thesis. The first section picks up on the architectural notion of modularity and explains why this systems design principle lies at the heart of the innovative capacity of platforms. The second section explores how the possession of essential platform components allows a platform owner to develop strategies that enable the regulation, maintenance and profiting from a platform ecosystem. The third section examines how a platform owner can exploit the potential of a platform by either opening up a platform or by maintaining a platform closed to third party participation. The fourth section investigates the notion of network externalities and explores how platform owners can benefit from harnessing this phenomenon. Whilst these four areas provide the basis for much of the research that has been carried out to date with respect to platforms, the fifth section introduces research from a field that has only recently turned its attention to this phenomenon. The field of information systems has the potential to bring new ideas to platform research. Software platforms form the foundation for modern software and systems development (Tiwana et al. 2010), and information systems is able to bring unique insights regarding digitalised technologies (Yoo et al. 2010a), an aspect not yet explored in platform literature (Tilson et al. 2012).

2.1 Understanding platforms as modular architectures

Across most of the platform research completed to date, a consensus has gradually emerged that a platform can be described as a set of modular components that can be selectively used and reused across multiple platform complements (Boudreau 2005). The means by which these modules connect and interact by way of their architecture is of importance, as this is what enables a platform to function. An understanding of platform architecture therefore provides insights into the specificities of platform innovation.

The origins of the concept of modularity can be traced back to the work of Simon (1962) who postulated that complex systems would be more manageable if they are designed and constructed as hierarchical and decomposable structures. In this way, the modularity of a product design is the degree to which products are composed of loosely coupled modular components connected with standardised interfaces (Ulrich 1995). Modularity is a means of simplifying the design of complex systems by "black boxing" components of a system. This involves minimising the interdependence between discrete modules, which means that they can be replaced or remixed in new configurations without compromising a system's overall functionality or performance (Baldwin and Clark 1997; Langlois 1992).

2.1.1 System architecture

Discussions of design modularity are essentially debates concerning architecture, which is the scheme by which the function of a design is allocated to its physical components (Henderson and Clark 1990). In practice, the architectural design of a system takes into consideration: the arrangement of functional elements; the mapping of these functional elements to physical components; and the specification of interfaces amongst interacting physical components (Ulrich 1995). Ulrich (1995) distinguishes between modular and integral architectures. A modular systems architecture, he explains, consists of one to one mappings between functional elements and physical components and de-coupled interfaces, such that components are modules which can be easily replaced. In contrast, integral systems architectures are made up of complex mappings of functional elements to physical components, such that any one physical component may carry out more than one function. Furthermore, the interfaces of integral architectures permit the coupling of components, such that a change in one component may require and adjustment in another component, so that a system will function again. Both systems architectures have advantages in different circumstances. An integral approach can yield parsimony in physical design, which can lend itself to cheaper unit costs under mass production. A modular approach, allows flexibility in system design, which not only allows for cheaper maintenance but enables designs to be reconfigured and modified, which in turn lends itself to a platform approach to design.

2.1.2 Design rules

According to Baldwin and Clark (1997), a modular system is composed of modular components whose internal structure is designed independently, but when combined function as integrated whole. They go on to explain that designers achieve this modularity by partitioning information into visible design rules and hidden design parameters. These visible design rules can be categorised into three: architectures, which specify what modules are contained within a system, and what their functionality is; interfaces, which describe how the modules connect and communicate; and standards, against which the degree to which a module conforms to the overall design rules is both specified and measured. Since modular systems are hierarchical and decomposable, the number of distinct elements within a systems design can be reduced by grouping them together into subsystems, which reduces complexity (Langlois 2002).

Whilst the design rules must be communicated openly such that modules can work together, decisions concerning the design within a module do not need to be communicated beyond the immediate design team. Hence the term hidden design parameters (Baldwin and Clark 1997), which is equivalent to information hiding (Parnas 1972) in order to simplify the design and construction of complex systems (Langlois 2002). Baldwin and Clark (1997, 2000) argue that it is the standardised interfaces that separate the modules that allows for their decoupling. Through decoupling, different organisations can take responsibility for the design of separate module, and can be confident that the combined system will function, as long as the design rules are followed (Baldwin and Clark 1997). Furthermore, the reduction in interdependence between modules that arises through modular architectures (Ulrich 1995), enables these components to evolve without affecting the ability for the system design to continue working (Tiwana et al. 2010).

However, in practice, decomposability and modularity are "relative" attributes of complex systems (Campagnolo and Camuffo 2010). For example, within the same system, there may be different levels of analysis which lead to different levels of modularity. These leads to limits to the degree to which designs can be decomposed, and that in practice most are integral-modular hybrids (Ulrich 1995).

2.1.3 The benefits of modular design

A number of key benefits are derived from taking a modular approach to systems design including flexibility, organisational freedom, innovation, and value capture.

The decoupling between modules that is brought about by standardised interfaces allows for significant flexibility in the design of modular systems. Modular systems theory (Schilling 2000) starts with the basic premise of increasing modularity results in increasing flexibility. Consequently decomposable modular components are advantageous when designing for unstable system designs that are likely to change. However, as Langlois (2002) notes, the overheads of agreeing and establishing standardised interfaces a priori can be costly, such that there may be advantages to non-decomposable integral designs when there are requirements for stable systems, which are unlikely to change.

Decoupling has an additional benefit as it enables the disintegration of both vertical and horizontal organizational structures. The consequence of this is that individual modules of a system can be produced by means of a decentralised production network (Langlois 2003), and firms who coordinate systems are able to appropriate rents of innovation by opening their technologies to outside firms (Langlois and Robertson 1992). As Langlois (2003) explains, this results in significant gains in overall innovative capacity.

For example, modularisation enables different companies to focus on particular modules, and to work independently on them, which provides the potential for innovation autonomous of the system owner. This division of labour allows for innovation to be opened up to a much wider set of capabilities. Rather than being limited to the internal capabilities of the corporation, modular systems, when opened up, can benefit from the cumulative skill, experience and technology of the wider economy. When modular systems are opened up, it allows for companies to innovate in parallel, sometimes in competition. This not only increases the pace of innovation, but allows for experimentation in what Nelson and Winter (1977) refer to as rapid trial and error learning. In this way a modular system is no longer limited by "the weakest link in corporate capabilities, but can avail itself of the best that the market can offer" (Langlois 2003, p375). Similarly, the speed of innovation and flexibility of modularity has benefits to offer the demand side. The capacity to reconfigure and to substitute facilitates mass-customisation, which allows for the "blanketing of product space" (Langlois and Robertson 1992), by tailoring products and services for different customer segments.

The establishment of modular as well as integral system architectures allows for new forms of innovation beyond the simple dichotomy of incremental or radical innovation. Modularised systems, in which architectures can be decomposed into modular components, are also open to innovation within their modular components, as well as architectural innovation with respect to the arrangement and connections between existing components (Henderson and Clark 1990). Whilst this may provide opportunities for some organisations, architectural innovation, in particular, is potentially threatening to established organisations. By enabling changes in the way that a system architecture is configured, architectural innovation can erode the usefulness of an incumbent's knowledge base (Henderson and Clark 1990). Established organisations may struggle to adapt to architectural innovation that has taken place within an industry, as what was the previous dominant design (Abernathy and Utterback 1978) has become engrained in their organisation structure and routines (Nelson and Winter 1982), leaving them to focus on modular and incremental innovation (Henderson and Clark 1990).

Significant opportunities for value appropriation are opened up by modular architectures (Jacobides et al. 2006). By designing and creating bottlenecks within architectures, such that the owner of the bottleneck is bestowed with decision rights over the inclusion of possible producers into particular segments of the industry architecture, positions of power are generated. These include powers to direct innovative activity and determine how innovations create and distribute value (Baldwin and Clark 2006). In this way owners of powerful architectural bottlenecks create their own "rules of the game" to design and to exploit architecture for competitive advantage. This is typically effected by facilitating the entry of others for the production of complements, which increases the value of an architecture through network effects, whilst actively protecting their own bottlenecks and source of power (Jacobides et al. 2006).

2.1.4 Modularity in platforms

Industry platforms lie at the heart of a particular form of modular architecture, which is described in detail by Baldwin and Woodard (2009). They claim platforms represent a set of modules with low variety and high reusability, consistent with standard definitions of platforms. Platform complements, which are modules with high variety and low reusability, surround this central set of platform modules, and selectively call upon their functionality according to what is permissible within the design rules (Baldwin and Clark 2000). Together these two sets of modular components, platforms and complements, are distinct parts of an overall systems architecture. Furthermore, Baldwin and Woodard (2009) claim that whereas the set platform modules tend to remain relatively fixed over time, complements are free to change over time. Consequently a platform embodies a stable set of design rules, which governs the interfaces between the modules and the ways that they can be bound together. As with other modular systems, this arrangement enables the disaggregation of design leading to all the implications described above, including the potential establishment of "business ecosystems" of complementary and competing firms.

However, as Gawer (2009a) notes, the logic of design in an industry platform is inverted when compared with many other types of modular systems. The platform owner, the owner of the modular system, is no longer the "master designer" who is responsible for combining components. This responsibility now rests with those third parties providing complements. In this way, Gawer (2009a, p56) goes on to conclude, "the final result of assembly is either unknown ex ante, or is incomplete".

This view must tempered by the fact that even though the logic of design is inverted, and that platform owners may not be responsible for arranging modular components to build platform complements, they have “decision rights that determine who can interact with or modify which components in what ways” (Baldwin and Woodard 2009, p.25).

2.2 Platform governance

The arguments for governing or regulating the activity that occurs on a platform fall into two camps. The first concerns a general need to manage and maintain the health of a platform (Iansiti and Levien 2004), by ensuring the quantity and quality of innovation that takes place (Parker and Alstyne 2010). If this is done, the value of the platform is maintained or increased, and more users are encouraged to adopt the platform and its complements (Gawer and

Cusumano 2008). Actions concerning attracting complementors to join a platform (Boudreau and Hagiu 2009), make investments in building new complements (Gawer and Henderson 2007) and assuring "coherent" technical development and coordination (Gawer and Cusumano 2008) are consistent with this intent. The second argument for regulating platform activity follows on from the first and concerns economic reasons of profitability for the platform owner (Parker and Alstyne 2010). Platform Owners are in a position to be focal private regulators by virtue of their one to many asymmetric relationship with other players (Boudreau and Hagiu 2009). As the "central player" within a two sided market, there is scope for the platform owner to coordinate ecosystem members in the interests of creating greater net value for the platform, than might otherwise occur if it were left ungoverned (Katz and Shapiro 1986). The incentive for the platform owner to carry out this role as "regulator" is that they are in a position to capture a share of the value created within the platform, for example through taxing each platform transaction, or through complementary platform sales (Boudreau and Hagiu 2009).

2.2.1 The right to regulate

Certain platform owners control and possess the scarce and critical assets that facilitate interactions within a platform ecosystem. This results in their unique position at the nexus of the one to many asymmetric relationships with all complementors in an ecosystem (Boudreau and Hagiu 2009). Furthermore, this position at the heart of the ecosystem benefits the platform owner with privileged information (Boudreau 2005). It is therefore bestowed with power that provides it with the potential for controlling activities within the ecosystem in the manner of a "public regulator" (Farrell and Katz 2000).

In this way the platform acts as a bottleneck (Jacobides et al. 2006). Ownership and control of this bottleneck provides the platform owner with bargaining power (Boudreau 2010) in their dealings with platform complementors.

The platform owner holds property rights over its platform, and this provides it with the "power of exclusion" associated with legal asset ownership. The platform owner is able to use these "Bouncer's Rights" (Strahilevitz 2006) to exclude outsiders from the platform (Boudreau and Hagiu 2009). Furthermore this ability to exclude also implies the power to set the terms of access to a platform, such that the platform owner can be thought of as a "licensing authority"

(Rochet and Tirole 2003). This ability to include or exclude complementors can be used by the platform owner as the basis of an incentive scheme for the regulation of a platform ecosystem (Boudreau 2005; Hart and Moore 1990; Holmstrom 1999).

A platform owner has the power to regulate a complementors access to and interactions with a platform (Boudreau and Hagiu 2009). The first type of regulation is used to control the complementor's participation with a platform and whether or not they can contribute innovation. Once a complementor is granted access to the platform, the platform owner has the power to regulate their interactions with the platform. The platform owner typically uses this to control for the type of innovation generated by the complementor.

Beyond the threat of preventing complementors access to a platform, the power that platform owners wield, enables them to determine other dimensions of a platform. They are entitled to determine the design rules of the platform, embedding "laws" within technology (Lessig 2006), which provide "arm’s length coordination" within the ecosystem (Boudreau 2010). The ownership of the platform as bottleneck also provides the ability to determine the direction of innovative activity, as well as the distribution of the value created by a platform (Jacobides et al. 2006). Gawer and Cusumano (2002) refer to this ability for a platform owner to regulate a platform ecosystem as "platform leadership".

2.2.2 Market price mechanisms

Literature concerning the regulation of platform has tended to emphasise price setting as the primary mechanism used by platform owners to control the interactions of others with a platform (Boudreau and Hagiu 2009). However Boudreau and Hagiu (2009) suggest that pricing mechanisms alone are not sufficient to manage platform ecosystems, as they are prone to market failures. They go on to state that reasons for these failures include externalities, information asymmetries, complexity, non-pecuniary motivations and uncertainty. This opens up the need to use other regulatory mechanisms in addition to market price.

2.2.3 Non price mechanisms for regulation

Over and above the use of price setting, platform owners regulate platforms through a number of different mechanisms (Boudreau and Hagiu 2009) including:

- Informational mechanism for coordination: to manage and coordinate through the communication of rules and procedures, the division of tasks, and the provision of support
- Technical Mechanisms: to ensure technical interoperability of complements and platforms through the establishment of design rules (Baldwin and Clark 2000).
- Contractual and legal Mechanisms: to establish property rights and to enable access to architectural intellectual property though licensing

The boundary of the platform (Boudreau 2005) marks those platform components that are opened up, possibly through licensing arrangements, for use by complementors. In general owners of smartphone platforms typically grant complementors access to the platform's operating system, low level system software, application programming libraries (APIs), high level system architectures definitions, and software development kit (SDK) (Parker and Alstyne 2010).

2.2.4 Platform strategy

As a consequence of their unique position in the nexus of a two sided market, platform owners have considerable power. With power comes the possibility of exercising it judiciously through the use of platform strategy to address their needs.

Platform owners have two major concerns (Gawer and Cusumano 2008; Parker and Alstyne 2010). The first concerns whether to grow the platform by providing incentives for third party developers to invest in innovating complements. The second concerns whether to harvest and profit from the platform. Whilst the two paths may not be mutually exclusive the platform owner needs to make strategic decisions.

The platform owner must make key decisions concerning platform design and intellectual property (Gawer 2009a). The challenges here include designing an appropriate architecture, such that they can create a competitive advantage (Baldwin and Clark 2006) and ultimately a dominant design (Abernathy and Utterback 1978). As part of this, the platform owner must also design appropriate interfaces at the boundaries of the platform, as well as deciding which platform modular components that it should grant complementors access to (Gawer and Cusumano 2008).

Second, the platform owner must manage the relationships the third parties who develop platform complements with care. As part of this come decisions as to how to enable and incentivise complementors in order that the platform can build market momentum (Gawer and Cusumano 2008; Parker and Alstyne 2010). Furthermore the platform owner needs to decide to what extent that it competes directly with third party developers by producing and commercialising its own platform complements.

Whilst these are beyond the scope of this research, there is an extensive literature regarding strategies followed by platform owners (Gawer 2009a; Gawer and Cusumano 2002).

2.3 Open and closed platform strategies as a matter of control

As the discussion on modular architectures indicated, industry platforms are suited to being opened up to outsiders. In this way, and depending on the ability of the platform owner to control their platform, modular components can be offered up one at a time (Boudreau 2010). This allows platform owners to capitalise on the innovative capabilities of external firms and build ecosystems of partners, in order to advance its development and commercialisation (Gawer 2009a), in an approach similar to open innovation (Chesbrough 2003).

Whilst there are obvious links with concepts of open innovation (Chesbrough 2003), platform innovation involves complex trade-offs between what is "open" and what is "closed" (Gawer 2009a; West 2003). Closed technologies are owned and controlled by a single entity, who restricts their use to outsiders by means of de jure and de facto protection (Boudreau 2010; West 2003). Open technology, on the other hand, is placed within the public domain, and is accessible by all. Depending upon licensing restrictions, the opening of technology may allow it to be openly supplied, and to be modified, transformed and built upon (Boudreau 2010; West 2003). The implications of this can potentially be far reaching, to open a platform or not is a critical decision involving two trade-offs.

The first concerns a trade-off between growth and appropriability (West 2003). Opening a platform can encourage adoption and growth by harnessing network effects. It can do this by reducing users’ fear of lock-in to a single vendor on the one hand, and stimulating production of differentiated goods that meet needs of different segments on the other. However at the same time it reduces the ability for platform owners to appropriate rents. This is because opening a platform typically reduces users’ switching costs and increases competition amongst platform providers by lowering barriers to entry (Eisemann et al. 2009).

The second concerns a trade-off between diversity and control (Boudreau 2010). On the one hand, an open system may benefit from the input, ideas and knowledge of a broader and more diverse pool of contributors (Chesbrough 2003; von Hippel 2005). On the other hand, a broad pool of contributors may lead to coordination issues which reduce the attractiveness of a platform for investment by third parties. As more parties try to innovate simultaneously, there is a risk that coherence may be lost across the system (Gawer and Cusumano 2002).

The issue with regards to whether to open up a platform for innovation concerns the question of, in the words of West (2003), "How open is open enough?". To address this, one needs to first consider what it is that makes a platform open. This can be done by examining the distinct roles that fall within a platform ecosystem, and that were described earlier. Each of these roles may be open or closed, and these need to be taken into consideration when characterising a platform as "open" or not (Eisemann et al. 2009). For example, Eisemann et al. (2009, p 133) compares the example of Linux as an industry platform ecosystem in terms of the openness of its constituent roles against those of the iOS platform ecosystem. Whereas all the roles (demand side user, supply side user, platform provider, and platform sponsor) are open in the case of Linux, in that parties are free to join any of these roles, they are all closed in the case of iOS, with the exception of demand side users. In this way it can be determined that whilst Linux is an open platform that embraces collaborative innovation, Apple's iOS is not. In the case of iOS, Apple is the unique platform provider and sponsor, and it operates a restrictive supply side in terms of carefully licensing application developers and vetting the complements that they produce.

This approach is taken further by Boudreau (2010), who identifies two fundamentally different means of opening up a platform, which are separated by their approach to control. The first is reflective of Apple's approach to managing its platform, where third parties may be granted access to the common functionality of a platform, thereby opening up markets for complementary innovation around the platform. In this way the platform owner carefully controls and regulates its platform, and ensures that all of its components remain interoperable. The alternative is to relinquish some, or all, control of the platform itself. This course of action is sometimes taken in order to establish the commitment of the platform owner to complementors, and to restore incentives for them to contribute (Boudreau 2010). The risk of giving up control is that platform fragmentation might ensue, which may hamper innovation. For this reason, the act of relinquishing control is often accompanied by the establishment of institutions (e.g. World Wide Web Consortium) to facilitate the coordination and consolidation of contributions (Boudreau 2010).

Given the growing literature concerned with this debate (Boudreau 2010; Eisemann et al. 2009; Parker and Alstyne 2010; Schilling 2009; West 2003), it is clear that there are relative merits for both opening or maintaining a closed platform, and that these are likely to be highly contingent on the context in which a platform owner is situated. Within the context of the research presented in this thesis, it is platforms that are controlled and relatively closed that are of interest. This would appear to be a crucial factor in the tensions between platform owner and third party developer in instances of contested innovation. The decision of a platform owner to follow a closed strategy may engender consumer trust, and hence encourage network effects, through which they can profit (Schilling 2009).

2.4 Profiting from network externalities

In certain markets the value of a good increases as a function of the total number of users of that good (Parker and Van Alstyne 2005; Schilling 2003), and the positive consumption benefits that arise are known as network externalities (Katz and Shapiro 1986). Classic examples of markets demonstrating network externalities are those that depend upon physical networks. This is characterised by the telecommunications industry, where the utility of a telephone is corresponds to the size of the telephone network, and the number of people who can be reached.

Network externalities also occur in markets that rely on complementary goods (Katz 1994), for example the market for record players and records. In these types of markets, products that have an established and significant installed base are likely to attract developers of complementary goods. Furthermore, users are attracted to those products that have a broad availability of complementary goods. This mutual attraction produces a self-reinforcing cycle that may grow the overall market for the base product (Schilling 2009)

Industry platforms, which can be considered as two sided markets, are characterised by similar effects caused by network externalities (Parker and Alstyne 2010; Parker and Van Alstyne 2005; Rochet and Tirole 2003). As a consequence of this, if the participation on one side of a platform increases, then participation on the other typically increases too (Boudreau and

Hagiu 2009). In turn the harnessing of network effects can therefore spur platform adoption (Baldwin and Woodard 2009).

The cycle of increasing returns that characterises these types of industries (Schilling 2009) is extremely attractive to platform owners for a number of reasons. The self-reinforcing dynamic has a potential to lead to "winner takes all" markets, which can lead to natural monopolies. A platform owner in a monopolistic position is able to extract supernormal profits. Furthermore, as the platform owner achieves this monopolistic position, its technology may emerge as the dominant design within the industry (Abernathy and Utterback 1978; Arthur 1989). As the owner of the dominant design, it will have architectural control (Henderson and Clark 1990) and be able to exert influence over an entire industry surrounding it (Schilling 2009).

The question of how to grow matched markets has been described as a chicken and egg problem (Parker and Van Alstyne 2005), as complement producers want consumers, and consumers want complements, before either side commits. The means of harnessing network effects in order to establish platform dominance is of significant interest to platform researchers. Research has tended to focus on opening up platforms for innovation (Boudreau 2010; Eisemann et al. 2009; West 2003) or subsidising one of the two sides of the platform market (Parker and Van Alstyne 2005; Rochet and Tirole 2003).

Finally the need to drive network effects to establish a platform has also been the focus of research by technology strategists (Gawer and Cusumano 2002). The broad options that a platform owner has to encourage network effects is to build the installed base or it can increase the availability of complementary products (Schilling 2009). Both sides of the platform can be developed with techniques including and in addition to subsidisation. For example the platform owner can encourage potential users' trust in the platform by ensuring that it is compatible with previous generations of technologies or with competitors' offerings (Schilling 2003). Similarly it can attract producers of complements by means of facilitating their development though development kits or extensive developer support.

2.5 Software platforms as digitalised artefacts

Within information systems, platforms, which have historically been under-researched, are gradually gaining awareness. This increase in awareness is on account that “platform based software ecosystems are emerging as a dominant model for software development and software based services” (Tiwana et al. 2010). To date, much of the information systems research on platforms has adopted concepts, definitions and terminology from the research already carried out within reference disciplines. For the sake of consistency with recent information systems research into platforms, these terms and definitions will be used in this thesis.

So, for example, platforms have been conceptualised as software platforms within the field of Information Systems, and defined as "the extensible codebase of a software based system that provides core functionality shared by the modules that interoperate with it and the interfaces through which they interoperate" (Ghazawneh and Henfridsson 2012; Tiwana et al. 2010). This definition is based on a more generalised industry platform definition proposed by Baldwin and Woodard (2009). However it can be claimed that this definition underplays the socio-technical nature of platforms as an instantiation of digital infrastructures, and this will be returned to shortly.

Platform research within the field of information systems conceptualises complements as "an add-on software subsystem that connects to the platform to add functionality " (Tiwana et al. 2010, p 676), which builds on the traditional notion of modularity (Baldwin and Clark 2000). Within the scope of this research, complements are constrained to third-party applications (apps) and platform enablers. Consequently Ghazawneh and Henfridsson’s (2012, P2) definition of complements as "executable pieces of software that are offered as applications, services, or systems to end-users of the platform", which is reflected in this dissertation.

The Information Systems notion of platforms adopts and adapts the conventions of modular architecture (Ulrich 1995), such that software platforms are governed by design rules (Baldwin and Clark 2000) concerning architectures, interfaces and standards. In this way software platform architecture is partitioned into a platform codebase with low variety and high reusability, and modules that exhibit high variety and low reusability (Tiwana et al. 2010).

Whilst information systems has adopted much from reference disciplines with regards to its understanding of platforms, it has much to contribute to research into this phenomenon (Tiwana et al. 2010). Much of the traditional platform literature has evolved from product development literature (Baldwin and Clark 2000) and this has very much shaped conceptualisations of the characteristics of platforms. However many platforms are based on digitalised technologies, whose properties are inherently different from tangible goods (Yoo et al. 2010a). Non IS related research into platforms has ignored the digitalised nature of this set of artefacts. Information systems is well positioned to progress an understanding of the digital nature of platforms, and software platforms in particular, as a social-technical phenomenon (Tilson et al. 2010).

The remainder of this section reviews information systems research relevant to this understanding of platforms as a digitalised artefact. First the unique properties of digital and digitalised artefacts, which distinguish them from other types of artefact, are explored. Second the capacity for innovation on digitalised technologies through architectural form is revisited. Finally the impact of digitalisation on platforms and the consequences for the generation of innovation and for its control are considered.

2.5.1 What is digitalisation?

According to (Tilson et al. 2010; Yoo forthcoming) the process of digitalisation "refers to the encoding of analogue information into a digital format and the possible subsequent reconfigurations of the socio-technical context of production and consumption of the product and services." In this way it extends the technical process of processing digitised information on digital technologies to a socio-technical process. In order to explore the social technical phenomenon of digitalisation it serves to deconstruct and explore the components of the previous statement. This is done first in terms of the properties of digital information and digital artefacts, which enable digitalisation. Attention then turns to the unique properties digitalised artefacts with respect to non-digitalised objects.

Digital information is simply information that is represented in digital, or binary, non-material as bit strings (Faulkner and Runde 2010). The digitisation of information is the process of encoding of analogue information in in digital form. Once in this form it is open to storage and manipulation by digital technologies.

Digital artefacts have a variety of material properties which differentiate them from their non-digital counter parts and allow for the processing of digital information (Yoo et al. 2010a; Yoo et al. 2010b). They have memorisable components, which allow the recording, storage and recall of digital data. They are programmable as they contain microelectronics which can perform basic operations on binary data based on Boolean logic. When this capability is combined with the capacity of digital memory to store both data and instructions, the creation of stored program computers based on the von Neumann architecture is enabled. Using components such as processors they become programmable digital artefacts. The basic hardwired, micro-code operations of processors such as ability to store data in memory, access data from memory, compare binary data, and to perform logical and mathematical operations on binary data, can be combined and scaled up in such a way as to enable complex operations in sophisticated high level languages such as Java. Digitalised artefacts have a characteristic that they can sense the world around them, as they contain sensors which enable information from the external environment to be transcoded into digital data for storage and processing. This can come in the form of external actors entering data, data being read from external stored media, or generalised sensors providing information concerning the external environment. These artefacts have communicability, allowing them to share information with immediate users through transducers such as displays, or at a distance by interacting with other artefacts, infrastructures and actors through communications networks. Digital artefacts may be addressable, which means that they can be identified and addressed within their context over a network. They may be traceable, which emerges from their properties of memorisability and senseability, thus enabling them to be associated with events and entities in space and time. Finally, and as a consequence of their ability to process information and their traceability, they may be associable, so that inferences can be made about their future states and conditions given associated events and entities.

As a combination of data and programmed algorithms, digitised information is an immaterial artefact. And like other immaterial artefacts, such as standardised routines, it has a practical instantiation and significance. Similarly digitalised artefacts are imbued with meaning, on account of their ability to process digital information, and thereby contribute to the accomplishment of human goals through performing certain functions. In this way combinations of digitised artefacts, and the digital information that they process, become embedded within people's social material practices (Orlikowski and Scott 2008), such that they become digitalised. The process of digitalisation is facilitated by advances in technology, made over the past fifty years, in terms of the miniaturisation of hardware, increasingly powerful processors, inexpensive and reliable memory, broadband communication and efficient power management (Yoo et al. 2010c).

Recently scholars (Faulkner and Runde 2010; Kallinikos et al. forthcoming; Yoo et al. 2010a; Zittrain 2009) have begun to analyse the unique structural features of digitalised artefacts which have contributed to the reconfigurations of social material practices in terms of routines and relationships.

For example, in their discussion of the ontology of non-material technological objects and their relationship to the social world, Faulkner and Runde (2010) identify three simultaneous properties of non-material digital objects that sets them aside from material objects. First they claim that digital objects possess the quality of non-rivalry in use. For example the use of Wikipedia by one person is not affected by the simultaneous use by another. Second, digital objects are infinitely expansible. Given sufficient bearer capacity, the iPhone/Android game Angry Birds can be available to an arbitrarily large audience at no marginal cost, or without risk of depletion. Third, they are re-combinable, so that they can be combined with other digital objects to generate many permutations of new kinds of non-material objects. It is through this quality that Apple's software development kit (SDK), enables developers to stitch together modules of code, or application programmable interfaces (APIs) in order to produce iPhone apps.

(Kallinikos et al. forthcoming) identify a further, but related, set of generic characteristics which set digital non-material objects apart from physically material objects. First, they claim digital objects are editable, since they can be modified or updated with more ease than physical objects. In this way the content of a Facebook page can be easily changed. Second, they are interactive, which allows them to be accessed by numerous people simultaneously, which is similar to the notion of non-rivalry in use (Faulkner and Runde 2010). Third, they are open in the sense that they are not constrained to one device, program or agent. It is this openness that allows a Word document to be accessed within Libre Office on a PC or viewed within Google Docs on and Android handset. Finally, they are distributed, so that they are borderless and can be contained over a variety of physical locations. The same iPhone app can be stored locally on numerous iPhones in multiple locations.

Both (Yoo et al. 2010a) and (Zittrain 2006) identify characteristics of digitalised technologies that facilitate the social practice of innovation. Yoo et al. (2010a) identify four such characteristics. First, the homogenised nature of digital data leads to a loose coupling between data and device. This is in contrast to analogue data, where there is a tight coupling between analogue data and analogue devices, as analogue signals are stored in a form compatible with the physical characteristics of the device on which they are processed. The homogenous nature of digital data comes down to its binary form, which means that it can be stored or transmitted on any digital bearer. Second, they identify the re-programmability of digital technology, which allows for simple modification or change in the function of digital artefacts. This is enabled by the stored program architecture of digital technology, which allows for the temporary storage of data and instructions. In contrast, functionality is typically hardwired into the material hardware of analogue technology and is consequently immutable. Third, they claim digital information has an immaterial nature, so that it can be reproduced and replicated up until the capacity of the material bearers on which it is stored or transmitted. This in turn enables the fidelity of digital information to its original source to be improved by increasing its resolution. Finally, the put forward the notion of the self-referential nature of digital innovation, which implies that the diffusion of digital innovations enables yet further digital innovation. Their argument for this property is as follows. The diffusion of PCs leads not just to an increase in digital tools for the creation of content and applications, but also to an increase in consumer base for their consumption. Furthermore the increasing penetration of high speed internet provides not just an increase in distribution capacity, but a means of networking and enabling innovators. Ultimately the combination of these factors creates positive network externalities that further accelerate the creation and availability of digital devices, networks, services and contents (Benkler 2006; Hanseth and Lyytinen 2010).

Finally, Zittrain (2009) develops the concept that digitalised artefacts, such as digital platforms, are generative as the underlying technology has an "overall capacity to produce unprompted change driven by large, varied, and uncoordinated audiences". For digital platforms, this involves third-party developers independently contributing to innovation of platform complements. This ability allows a self-contained system to create, generate, or produce new content, structure, or behaviour without additional help or input from the original creators (Tilson et al. 2010). The conceptualisation of generativity has its origins in the work of cyberlaw scholars Lessig (2002) and Benkler (2006). Lessig (2002) was the first cyberlaw scholar to turn his attention to the innovative essence of the Internet. He provides a theoretical framework for analysis based on Saltzer's (1984) notion of an end-to-end architecture to explain why the Internet ecosystem allows for the development of innovative applications by those who are at the endpoints of the Internet network. Saltzer claims that functionality is most effectively developed by and placed close to those who use applications of that function, as this is where the knowledge surrounding the needs and development of an application lie. Benkler (2006), another cyberlaw scholar, extends Lessig's argument by claiming that it is the addition of easily programmable terminals in combination with the end-to-end architecture of the Internet that provides the basis for innovation in the Internet. Zittrain (2009) takes this argument further.

He proposes that the success of the Internet is as a result of its "generative" nature. Zittrain terms the PC and the Internet as a generative system that is open, flexible and an enabler of unthought-of innovative uses. Hence, the characteristics of a generative system are: leverage, or the extent to which tools make possible a set of activities which would be impossible or prohibitively expensive to achieve otherwise; adaptability, or the scope of uses tools can be put to and the ease to which they can be modified to extend this range of uses; sense of mastery, or how easily tools are adopted and adapted for use by a broad audience; accessibility, as the ease with which tools can be obtained, along with the information to use them; and finally, transferability, namely how transferable any change or new uses of tools are to others. Generativity is thus the capacity of a system to enable new or innovative use. Zittrain crafted his definition with the generative capacity of the internet in mind. However within the bounds of the innovation of complements on digital platforms, this definition can be narrowed for the purposes of this research to "The capacity of a digital platform to facilitate the unprompted creation of potentially novel platform complements by a varied and independent audience".

2.5.2 Digitalisation, new architectural forms and innovation

The process of digitalisation has enabled the evolution of architectural forms within digitalised technologies. Architectural form, or the arrangement of functional elements, the mapping of functional elements to physical components, and the specification of interfaces between these elements (Ulrich 1995) is of strategic importance as it drives the latitude to which third parties are able to contribute further innovation (Yoo et al. 2010a).

If the first wave of digitalization (Yoo et al. 2010b) involved the technical digitization of converting analogue contents and services into digital ones, the second wave enabled something altogether more complex. It enabled an evolution of architectural form in digitised artefacts beyond integral and modular architecture (Ulrich 1995; Yoo et al. 2010b). The second wave capitalises on the homogeneity of digital information, which allows for the separation of tightly coupled devices, networks, services, and contents. This is manifested in layered architectures within digital technology, which provide yet more architectural flexibility and independence (Adomavicius et al. 2008; Gao and Iyer 2006) and facilitate further innovation. This form of architecture is typified by the Open Systems Interconnection (OSI) seven layer model used in the network and telecommunications industries. A four layer architecture of device, network, service and contents (Benkler 2006) provides a simplified view of a layered architecture. In this way standardised interfaces and between the layers, provides designers complete independence with regards to how they innovate and create functionality within a layer. In this way, the layers represent different design hierarchies (Clark 1985), so that decisions concerning each layer can be made independently of its effects on other layers. The ability to separate between these layers is enabled by characteristics of digitalisation seen earlier (Yoo et al. 2010b), namely the reprogrammability of digital technology and the homogeneity of digital data. The separation of function, in the form of service, and material form, as physical device, in the layered architecture is allowed for by the reprogrammability of digital technology. The separation of network and contents is enabled by the homogenisation of data, and the ability for convergence that it allows. A layered architecture offers upwards and downwards flexibility (Tilson et al. 2010), as each layer is able to combine and capitalise upon any of the functional capabilities of the layer below it. Not only does this allow a myriad of potential applications, but it allows for a recursivity (Hanseth and Lyytinen 2010) allowing for the creation of boundless levels of infrastructures, platforms, applications and IT capabilities.

A later third wave of digitalisation (Yoo et al. 2010b) allows for the emergence of novel products and services through "mash-ups" over previously tightly coupled architectural boundaries. This third wave is enabled by a four layered modular architecture (Yoo et al. 2010a), a looser coupling between layers and modules, which facilitates the unleashing of generative practices. The layered modular architecture is conceived of as a hybrid of the modular architecture of a physical product and the layered architecture of digital technology. Product functionality of physical products is enhanced with software based capabilities. In a modular architecture, the relationship between the product and its components is nested and fixed. The components are product specific, which must be designed before the components are developed. The designers of a modular platform pre-specify all the modules and interfaces between them (Baldwin and Clark 2000; Ulrich 1995). However the four layered modular architecture draws on the standardisation of interfaces, and the modularity and granularity of digital artefacts (Kallinikos et al. forthcoming) to enable generativity. In a layered modular architecture, boundaries are not so fixed. The design of a component does not require product specific knowledge, components are product agnostic, and are often built without an end product in mind. The design of a component does require the knowledge of how the interfaces at the boundaries of a module function, so that it can be successfully combined with other components later. A product is created by combining groups of loosely coupled components though pre-specified interfaces over a set of heterogeneous layers, each belonging to a different design hierarchy (Clark 1985). In this way a layered modular architecture offers generativity within the constraints of physical form factor and the available functionality of the physical components of the product. Just as the concept of modularity (Baldwin and Clark 2000; Langlois 2003) has impacted industrial organisation, Yoo et al. (2010a) argue that the emergence of the layered modular architecture generates further changes to the organising logic of firms. The consequences of the four layer modular architecture has implications on the social nature of innovation in terms of who it is that innovates and what it is that is innovated. Yoo et al. (2010a) characterise the organising logic for a layered modular architecture as doubly distributed (Yoo et al. 2008). The organising logic is doubly distributed for two reasons: first, control over product components is distributed across multiple firms; and second, product knowledge is distributed across heterogeneous disciplines and communities.

2.5.3 Platforms as digitalised artefacts

One class of digital artefacts that are subject to a layered modular architecture are digital infrastructures (Tilson et al. 2010). These are defined as "the constitutive information technologies and organisational structures, along with the related services and facilities necessary for an enterprise or industry to function" (Tilson et al. 2010, p 748). The combination of the physical artefact that is the smartphone, the mobile OS, applications developers, content providers, and complex networks embody an example of a digital infrastructure (Tilson et al. 2012). Digital platforms, such as mobile platforms form a partition within a digital infrastructures configured as industry platforms, and are subject to all the characteristics of digitalised artefacts (Yoo et al. 2010a). As information infrastructures (Hanseth and Lyytinen 2010), digital infrastructures are generally open and free of centralised control, whereas the software platforms that they subsume, have the facility to be more closed and centrally controlled. As a consequence of adopting the perspective of digital infrastructures, I define digital platforms as “Digital infrastructures that are configured as industry platforms and that make available digitalised components (hardware, operating system, SDKs, APIs, and application delivery mechanisms) that act as a foundation upon which developers can build complementary services”.

The characteristics of digitalisation are such that the logic of innovation on digital platforms is potentially quite different from traditional product innovation (Svahn and Henfridsson 2012). For example when looking at the architectural design of digital platforms, a sense of instilling generative capacity for the reuse of ideas is as essential as enabling modularity. Consideration needs to be given to designing digital platform architectures that can scale to allow for a multiplicity of modules that digital materiality allows for. Another point to consider is that digitalised artefacts such as software platforms allow for a procrastinated binding (Svahn and Henfridsson 2012) of functionality to product. Whereas in physical assemblages, functionality is bound to a product at its point of manufacture, digitalised artefacts allow for functionality to be bound at the point of consumption. This is exemplified by the ability to download an application onto a smartphone, well after the point of manufacture and product purchase, which will allow additional functionality such as the ability to follow a Twitter stream.

Another characteristic of digital platform innovation concerns a tension between the platform owner's need to enable the innovation of platform complements in general and the need to curate or control innovation for desirable complements. As was seen in the example concerning Google Voice given in the introduction, this tension can sometimes lead to conflict between platform owners and developers. This conflict manifests itself in complex sequences of interactions between the two parties as they contest the boundary of permissible innovation. Furthermore, there appear to be occasions where developers are able to influence the boundary of permissible innovation, which brings into question the degree to which platform owners control the innovation of platform complements.

The notion that software platforms contain a tension between control and generativity is encountered repeatedly across the growing body of information systems research into software platforms. There is consequently an emerging consensus (Eaton et al. 2012; Ghazawneh and Henfridsson 2012; Tilson et al. 2010; Tilson et al. 2012; Tiwana et al. 2010) that digital platform innovation is often a dynamic process full of tension. In their research commentary concerning the interplay of architectural design, platform governance, and the environmental dynamics of ecosystems on the evolution of platforms, Tiwana et al. (2010) refer to a delicate balance of control by a platform owner and autonomy amongst developers. Tilson et al (2010;2012) discuss a tension which arises when platform owners seek to gain the advantages of generativity through third party innovation whilst attempting to control this innovation in their own interests. The same is the case in (Ghazawneh and Henfridsson 2012) when they present research which reports the need for platform boundary resources for resourcing and securing platform ecosystems through third party development. They refer to "a tension in the design of these boundary resources between maintaining platform control and stimulating 3rd party developers to build" (p. 2).

Furthermore, the idea that other ecosystem members can challenge platform owners' assumed control over innovation has also surfaced in IS platform research. Tilson et al. (2012) acknowledge this possibility, and, using the terminology of Eaton et al. (2011) suggest that platform owners control of innovation might be "influenced" by other ecosystem members. This possibility is advanced further by Eaton et al. (2012) who provide empirical examples of conflict over contested innovation between Apple, as a platform owner and developers. These empirical examples describe occasions where the platform owner has indeed been influenced with respect to decisions regarding what innovation of platform components can be permitted.

Clearly, there are relatively few studies into the tensions that occur on digital platform. There is a need to extend existing work with more empirical studies. In addition there is also a need to move beyond the description of examples of contested platform innovation to provide explanations as to how platform owners control over platform innovation can come to be challenged.

2.6 Summary

To date research into industry platforms appears to be largely the domain of scholars of product development, technology strategy and industrial economics. They have researched a number of problems concerning the management of platform strategy. They have, for example, studied how to profit from platforms by following open and closed strategies, as well as by harnessing network externalities. Approaches to platform governance using both price and non-price mechanisms have formed a basis for much of this research. Amongst the non-price mechanisms that are studied is that of platform architecture. This stream of research assumes that the platform owner has the power to control platform architecture, and hence platform innovation in the form of realised platform complements.

The field of information systems has only recently become interested in the phenomena of platforms and platform innovation. This is surprising given that much IS systems development is carried out on digital platforms. Whilst recent IS research into platforms has taken many conventions and concepts from previous research in reference disciplines, it has the potential to contribute new insights into platform phenomena. For example, the process of digitalisation and the characteristics of digitalised technologies are areas of interest to IS. These concepts have not been applied to platform research by the reference disciplines, and IS may consequently use them to bring new knowledge to studies of platforms. The concept of digitalisation is enabling a growing body of IS research that considers the tensions that exist between generativity and the control of innovation on digital platforms. There is however a need for empirical studies of the conflicts that arise between developers and platform owners over contested innovation on digital platforms. It is proposed that a description and explanation of the complex sequences of interactions that arise within such conflicts may generate two types of insights. First they may yield a deeper understanding of the tension between control and generativity. Second it may provide more clarity with regards to how the boundaries of permissible platform innovation are established.

3 Methodology

This thesis has the broad objective of developing a process theory to describe and explain sequences of action. To that end this chapter introduces a methodology, in the broad tradition of narrative analysis that enables sequence analysis of empirical data. The chapter then goes on to explain how the technique of narrative networks (Pentland and Feldman 2007) is used as a means of carrying out sequence analysis. Narrative networks are first described in a general sense, and then there is an extensive explanation at how they are used in practice in this research. This explanation takes into account how empirical data is sourced, manipulated, analysed and then presented in the form of narrative networks. Attention then turns to a different data set that is used to inform an analysis of mechanisms which explain the how sequences of actions unfold.

3.1 Sequence analysis as narrative analysis

The research question and sub questions, posed in this thesis, concern the description and explanation of common patterns of actions across different instances of sequence data. These questions are typical of those addressed in the literature concerning sequence methods (Abbott 1990; Corsaro and Heise 1990). Sequence analysis is considered (Abbott 1992; Abell 2004) to be a method that falls into the wider approach to data analysis that is termed narrative analysis.

Narrative analysis is broadly concerned with the study of texts concerning sequences of events and their consequences (Riessman 1994). As a genre of research, narrative analysis has its origins in literary analysis (e.g., Bal 1985; Barthes 1977; Propp 1968). However its applicability to social science research is apparent as social groups and organisations construct narratives concerning their shared and individual experiences of historical events (Riessman 1994).

Within the social sciences, and consequently organisational science and information systems, there are numerous approaches to narrative analysis, ranging from the study of oral narratives of personal experience (Riessman 1994) to organisational narratives (Boje 1991; Czarniawska-Joerges 1998). At one extreme approaches have focussed on deep contextual analysis of the way stories are told, as is reflected in the structural narrative analysis of Barthes (1977). At the other extreme studies have focussed purely on the content of a text, the "what" or as Abbott (1992) terms the narrative positivism of focussing on sequences of events in themselves. Management science approaches to narrative, generally concur that it represents a coherent sequence of events with a clear purpose or goal (Czarniawska-Joerges 1997; Czarniawska-Joerges 1998).

Within information systems research, narrative analysis has been used in studies concerning IT Implementation (Brown 1998; Sabherwal and Robey 1993) and IT failure (Bartis and Mitev 2008; Dalcher 2004) as well as in studies concerning best practice (Wagner et al. 2004). The approaches to narrative analysis in these studies have ranged from an interpretive approach to organisational narrative (Boje 1991) adopted by Brown (1998) to Sabherwal and Robey's (1993) adoption of Abbott's (1992) narrative positivism in applying optimal matching techniques to sequence analysis.

In this way, the sequence of events that I study are coherent, they are connected with each other. The participants involved in these interactions have goals which help direct, connect and make coherent the actions that they take. These are not disconnected or random events. Consequently these actions are part of larger narrative of events, and it is fitting that methods of sequence analysis, as part of narrative analysis, are applied in order that they be understood. The question is what approach to sequence analysis is best matched to the research problem?

3.2 Narrative networks: a methodological tool for sequence analysis

The first and fundamental requirement of this research is to identify patterns of actions carried out by members of a platform ecosystem. This needs to be done for two reasons. First, it is done in order to untangle their complex interactions into simpler patterns. Second, it is done to allow different instances of sequences of actions to be compared, to see if there exist fundamental patterns of actions. Only when the existence of sets of patterned sequences of actions is established, can the second part of the research question be addressed. This concerns the identification of underlying mechanisms that generate certain patterns of actions. The essential information that needs to be uncovered concerns who is doing what to whom and in what sequence.

A narrative network is a means of presenting sequential patterns of actions, which have been, or could be, observed in a narrative (Pentland and Feldman 2007 : p.787). A narrative network is visualised as graph. Pentland and Feldman (2008 : p.244) define a narrative network as "a collection of functional events related by their sequential occurrence in a story or set of stories"

By functional elements, Pentland is referring to a concept developed by (Hendricks 1972 : p.100), which refers to a basic building block of narrative (Pentland and Feldman 2008 : p.244). Hendricks argues that a narrative can be modelled as a sequence of "functional events", each one of which advances a plot (Pentland and Feldman 2007 : p.787). Pentland and Feldman go on to refer to these functional elements as narrative fragments (Pentland and Feldman 2008 : p.244). In order to advance a plot, each narrative fragment contains two or more actants and an action that links them. As such each fragment describes an event in terms of who is doing what to whom. Graphically a narrative network consists of nodes linked together by ties. Each node represents a narrative fragment, and each link, tying one narrative fragment to the next, represents sequential progression from one fragment to the next, in order to advance the plot.

Not only do narrative networks summarise individual patterns of action, they enable many sequences to be plotted on the same graph. This allows for the comparison of sequences and for pattern matching. Furthermore, they help prompt appropriate questions that facilitate the identification of mechanisms (Pentland et al. 2010 : p.932). For example, they may prompt the question why a particular narrative fragment generally follows another fragment.

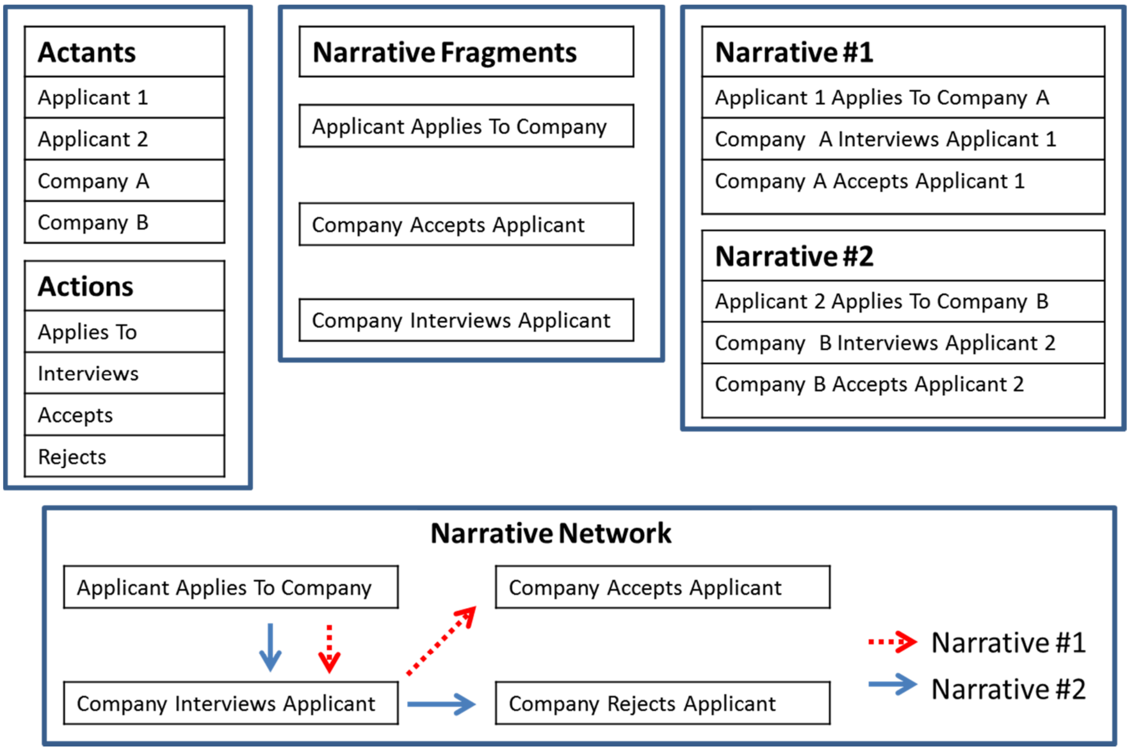

Actants
Applicant 1
Applicant 2
Company A
Company B
Actions
Applies To
Interviews
Accepts
Rejects
Narrative Fragments
Applicant Applies To Company
Company Accepts Applicant
Company Interviews Applicant
Narrative #1
Applicant 1 Applies To Company A
Company A Interviews Applicant 1
Company A Accepts Applicant 1
Narrative #2
Applicant 2 Applies To Company B
Company B Interviews Applicant 2
Company B Accepts Applicant 2
Narrative Network
Applicant Applies To Company
Company Accepts Applicant
Company Interviews Applicant
Company Rejects Applicant
Narrative #1
Narrative #2

Figure 3.1 An example narrative network.

(Own Diagram)

An example narrative network is illustrated in figure 3.1. The broad narrative that is considered here is a simplified hiring process. Figure 3.1 illustrates two scenarios of this narrative, the story of applicant 1 and their successful application to company A, and the story of applicant 2 and their unsuccessful application to company B. The illustration starts by indicating the various actors and actions involved in this narrative. It then displays a number of narrative fragments that can be built from these components. Following this it then replicates the two narrative instantiations using narrative fragments. Finally a generalised narrative network is displayed, showing the narrative fragments as nodes, and the paths of the two narrative instantiations between the nodes, displayed a directed ties. Before moving on, a final point to note is that it is possible to build scenarios within narrative networks, regardless of whether they have actually occurred. For example, it would easily be possible to visualise the scenario where an applicant applies to a company, and is accepted (or rejected) without an interview.

There are a number of alternatives to narrative networks as a method for describing and analysing patterns of actions. For example there are non-graphical methods such as the use of grammars (Pentland and Reuter 1994) and optimal matching analysis (Abbott and Hrycak 1990), as well as the rigorous formal framework of Abell (1987). Whilst they are rigorous more graphical methods, such as narrative networks, are simpler and easier to produce, analyse and communicate.

Whilst Abell's formal methods (1987) do allow for the representation of sequence in graphical form, nodes represent states rather than narrative fragments, such that one does not get the same sense of narrative flow. The graphical event structure analysis of Corsaro and Heise (1990) has been praised for its representation of narrative structure in events (Abbott 1992), however Pentland and Feldman (2007) claim that narrative networks offer more flexibility with regards to the relationships between events. This observation is certainly true in so far as it is possible to model both observed and possible sequences of events on a narrative network. It would, however, be mistaken to confuse narrative networks' simplicity with a lack of depth. Narrative networks can be represented as first order Markov models, which affords a whole host of rigorous statistical analysis, should there be sufficient data. It seems apparent, that for my purposes, narrative networks are the best option because of their simplicity, flexibility and power.

This is not to say that narrative networks, along with other approaches to sequence analysis, are not without their limitations. Pentland and Feldman (2007 : p.793) cite three short comings. First, unlike alternative approaches to narrative analysis, sequence methods abstract narrative to the extent that contextual meaning is lost. Second, the narrative fragments that make up the nodes of a narrative network are effectively black boxed, so that the level of analysis of actions does not become more fine grained. Last, narrative networks approach action in sequential event time. That is to say they handle events as discrete entities. The method does not extend to the complexities of managing overlapping events in clock time. Nevertheless, as long as the required focus is purely on actants acting on each other in time and space, that the coding scheme describing narrative fragments is defined at the appropriate level, and the constraints of event time are accepted, then the approach can be of great value.

I do not consider these limitations to be problematic to my proposed research question. In the first instance, the loss of contextual meaning is not serious, as the research question is focussed on clarifying the actions and analysing how they are patterned in isolation of context. The second concern regarding black boxing is a concern when analysis is required at a level of detail which is finer than the level of actions that are being analysed. As is stated in the section concerning coding, and in order to overcome this limitation, all efforts are made to ensure the level of analysis is a fine as is required. Finally, the fact that sequential analysis handles events as discrete entities plays very much to the demands of the research question. This research requires actions to be treated as discrete entities, so that they can be made sense of. There is, however, a concern that the stripping away of context may restrict the validity of the final part of the research question. This sub question concerns the identification and explanation of mechanisms that cause particular patterns of actions seen across different instances of sequences of events, which may well require a consideration of context. Whilst this concern is addressed in more detail further on in the thesis, it suffices to say at this point, that contextual information from the source data can later be reintroduced to facilitate this level of analysis.

3.3 The principles of building a narrative network

The flexibility in narrative networks means that there are a many different ways to approach their construction. Pentland and Feldman (2007 : p.790) identify a four stage approach.

The first task is to identify the focal phenomenon to be studied as an event sequence and to define boundaries around this phenomenon. The focal phenomenon not only describes the subject that is to be researched, but also the unit of analysis. The purpose for this is to focus both the collection of data and its analysis.

The second task is to establish a point of view, or perspective, from which the narrative sequence is unfolding. The perspective from which a narrative is being recounted, may affect where a sequence starts and finishes, as well as which actions the sequence emphasises and the granularity of the actions that are described.

The third task is collect stories, identify and code narrative fragments. Clearly this requires not only deciding upon an approach to data collection, but also the identification of a coding scheme that is consistent with the criteria identified in the first and second stages.

The fourth and final task concerns relating the narrative fragments in sequence and representing this by constructing a narrative network of nodes linked by ties.

3.4 The practice of building a narrative network

In practice it was found that Pentland's approach needed to be adapted in order to better suit the needs of the research. First of all the coding of data was split out from the collection of data. Second, the nature of my data source meant that it was possible to only collect disparate, unordered fragments of stories, rather than whole stories, as one might get from interviews. This meant that before sequences of actions, which were found in the data and that make up the rump of the analysis, could be coded, it was necessary to construct stories from the data that had been collected. Last, after Pentland's final stage, concerning the construction of narrative networks, a final stage was added focussing on the analysis of the narrative networks. In summary the six stages that were used are as follows:

The first task is to identify the focal phenomenon to be studied as an event sequence and to define boundaries around this phenomenon. The focal phenomenon not only describes the subject that is to be researched, but also the unit of analysis. The purpose for this is to focus both the collection of data and its analysis.

The second task is to establish a point of view, or perspective, from which the narrative sequence is unfolding. The perspective from which a narrative is being recounted, may affect where a sequence starts and finishes, as well as which actions the sequence emphasises and the granularity of the actions that are described.

The third task is to collect narrative data, which includes finding an appropriate source for the data, and deciding upon what data is to be collected and how it is to be obtained and stored.

The fourth task concerns identifying stories within the data. This involves parsing the data in order to find narrative fragments, which can then be appropriately processed into themed stories. Any gaps in the stories can be addressed if appropriate.

The fifth task concerns coding the narrative fragments, particularly the actions, which will make up the narrative network. This requires the identification of a coding scheme that is consistent with the criteria identified in the first and second stages.

The sixth task concerns the construction and analysis of narrative networks. Narrative networks were constructed by relating nodes, consisting of narrative fragments, in sequential order by means of ties, which represent transitions between fragments in the narrative. Analysis broadly consisted of comparing individual sequences to identify common patterns of action, in order to begin to identify generative mechanisms.

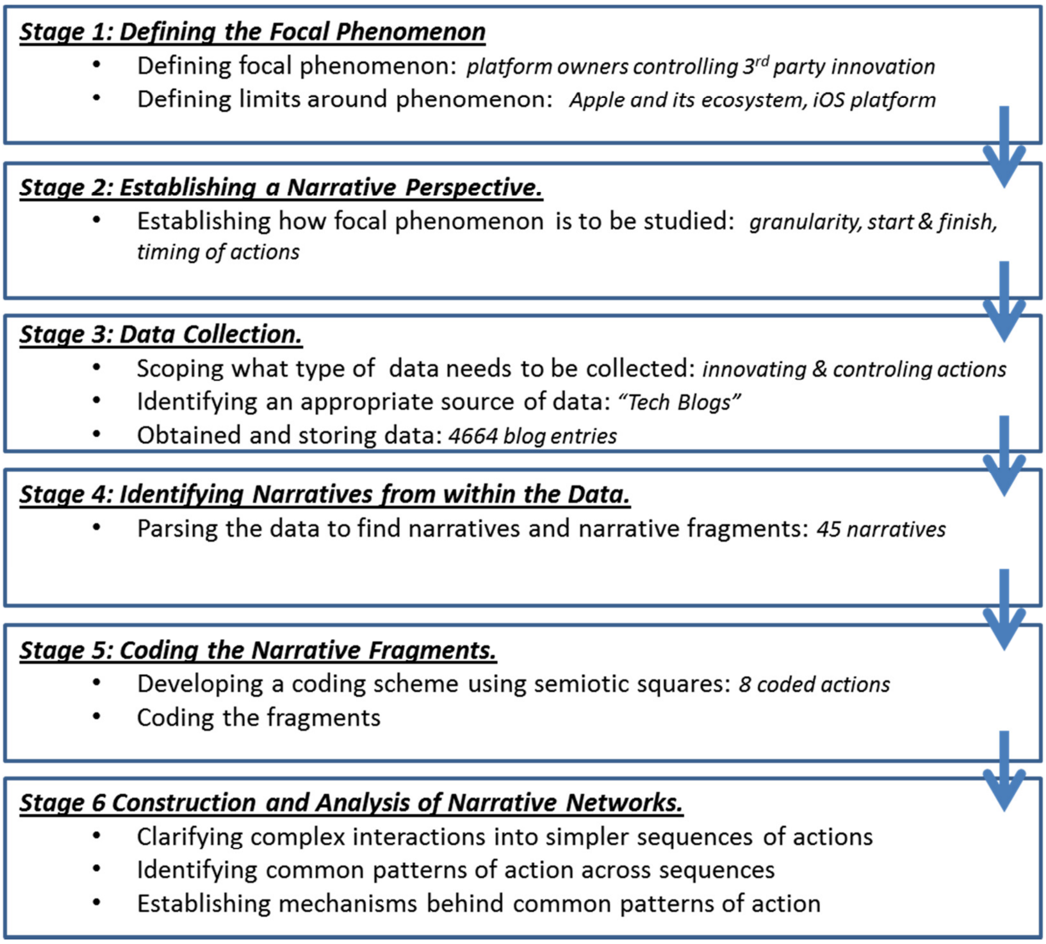

Stage 1: Defining the Focal Phenomenon
• Defining focal phenomenon: platform owners controlling 3rd party innovation
• Defining limits around phenomenon: Apple and its ecosystem, iOS platform
Stage 2: Establishing a Narrative Perspective.
• Establishing how focal phenomenon is to be studied: granularity, start & finish, timing of actions
Stage 3: Data Collection.
• Scoping what type of data needs to be collected: innovating & controling actions
• Identifying an appropriate source of data: "Tech Blogs"
• Obtained and storing data: 4664 blog entries
Stage 4: Identifying Narratives from within the Data.
• Parsing the data to find narratives and narrative fragments: 45 narratives
Stage 5: Coding the Narrative Fragments.
• Developing a coding scheme using semiotic squares: 8 coded actions
• Coding the fragments
Stage 6 Construction and Analysis of Narrative Networks.
• Clarifying complex interactions into simpler sequences of actions
• Identifying common patterns of action across sequences
• Establishing mechanisms behind common patterns of action

Figure 3.2 The six stages of building narrative networks.

(Own Figure)

Figure 3.2 provide a summary of the six stages that were used to build narrative networks in this study, and what follows, is a description about how these tasks were gone about in practice.

3.4.1 Establishing the focal phenomenon and the boundaries of research

The first task is to identify the focal phenomenon to be studied as an event sequence and to define boundaries around this phenomenon, in order to identify "a sphere of activity" (Pentland and Feldman 2007 : p.790). The focal phenomenon not only describes the subject that is to be researched, but also the unit of analysis. The purpose for this is to focus both the collection of data and its analysis.

The focal phenomena that are being researched in this study are the seemingly complex sequences of interactions that occur as platform owners and other members of the platform ecosystem negotiate the extent of innovation on mobile platforms. It would seem that platform owners use controlling actions in response to the innovating actions of other ecosystem members. In answering the research question, my interest is not only in these sequences of actions, but also in the events that trigger and the mechanisms that enable the platform owners controlling actions.

Following on from this, in event sequence analysis studies the unit of analysis varies between a focus on individual actions, events or moves (e.g., Pentland 1992) or on entire sequences or performances of actions (e.g., Pentland and Reuter 1994), depending on the goal of the study.

Since the overall goal of this research is to identify and explain common sequences of actions then the unit of analysis focusses on overall performances. With regards to defining the boundaries to research, Pentland and Feldman (2007 : p.790) cite Burke (1969) as a means to establishing the boundary of a narrative for analysis. Burke (1969) identifies five elements, namely: scene; act; agent; agency; and purpose.

With respect to scene, act and purpose this study focusses on the focal phenomenon described above. I specifically choose to focus on Apple's iPhone (iOS) platform ecosystem as a scene as it represents an example of a regime of platform control with the characteristics of interest to this study, namely: Apple monopolise the distribution channel of platform complements to consumers; Apple curate content within the iOS platform ecosystem; there are instances of contested innovation to study. In terms of the main agents to study, I decided to focus on Apple as the platform owner and application developers and platform enablers as those members from the platform ecosystem. These particular agents were chosen as it is their agency, or actions, which fundamentally make up the events that were see in these complex interactions. Whilst other agents may appear in these interactions, I treat them as secondary participants in the action. My interest in the actions or agency with which the agents engage, is limited to actions that either attempt to progress or control innovation on a platform.

3.4.2 Establishing a narrative perspective.

Establishing a narrative perspective not only provides additional focus to the data collection and coding effort, it also shapes the narrative that is presented for analysis. As Pentland and Feldman (2007 : p.790) point out, the stories collected in the field will be told from a particular point of view. This point of view may well be that of the informant's, or the informant's interpretation of another point of view. Ultimately, whatever the data, it will to some degree be further interpreted by the researchers involved in its analysis. Consequently a degree of bias may promote certain events in a sequence to the detriment of another. Ways to overcome this bias include, robust coding methods and by stating up front, the narrative perspective of this study. Whilst the narrative perspective of the informants is discussed in the section concerning data collection, it is possible to establish the perspective of the researcher. My perspective is informed by: first, a definition of the start and end points of the sequence that I am interested in; second, the granularity, or level of detail, of the events that I examine; and last, my treatment of time, or the passage of events, in the sequence. It goes without saying, that my focus on actors is impartial, and that I do not wish to promote any stage of the sequence of events over another. In combination these factors provide additional definition to the boundaries of a narrative.

Granularity

The decision concerning the granularity of events that are captured and coded, not only defines the level of detail and complexity of analysis afforded, but also the story that is told. A phenomenon expressed as a narrative consisting of micro events may seem quite different from a narrative consisting of macro events. Abell (1987) advises that there are diminishing returns in terms of gains in insight verses the level of detail sought in sequence analysis. Sabherwal and Robey (1993) suggest that too fine a level of granularity leads to a level of complexity of coding that becomes difficult to reliably analyse. Malone et al. (1999) recommend that in practice, a focus on more substantial actions leads to a more meaningful analysis. The granularity of actions that are analysed are also driven in part by previous decisions concerning the phenomenon under study, as well as decisions concerning the sources of narrative data and what they are able to tell us. For this last reason, I choose to limit the granularity of the actions I study, to the macro level of commercial entities acting on each other, as they affect the terms of innovation on a platform, as reported in public discourse.

Start and finish

The points at which a sequence of events starts and finishes can depend on the perspective of the observer. A structuralist perspective of narrative required a clear beginning, middle and end (Bal 1985). However, this can sometimes be difficult to identify, as Pentland (1999b : p.11) states "since everything is connected to everything else, where does one draw the line?". White (1980), as cited in Pentland (1999b : p.11) provides some guidance that is relevant to my case. White explains that the structure of a story has a clear moral or authoritative context, a challenge to the order at the start, and some resolution to that order at the end. In this way, I define the start of the sequences as the first attempt, in a particular story, by a member of the platform ecosystem to challenge the boundary of what can be innovated on a platform. I define the end of a sequence of events in the stories I analyse, by the final or last recorded action, which appears to achieve a steady state with regards to whether a new innovation is accepted or rejected.

Treatment of time

The treatment of time in narrative goes beyond when an event occurred, over what period of time and how long and what happened in the intervals between these steps. Consideration also needs to be given in terms of how events overlap, what events occur in parallel, and the difference between the sequence of events as reported by an informant in narrative form, and the actual playing out of events in fabula. Most approaches to sequence analysis (Abbott 1990; Abell 1987; Corsaro and Heise 1990) require a simple approach to the treatment of time in narrative. They are not yet able to cope with the complexities of overlapping or parallel events, and in that way narrative networks (Pentland and Feldman 2007) are no different. For that reason my approach to the treatment of time is linearize concurrent sequences, and to use place events in sequence according to the order of when they actually happened, otherwise known as event time.

3.4.3 Data collection

In order to analyse and compare patterns of action, I must first collect sequential data. With this in mind, the aim of my data collection is to build a corpus of stories as qualitative data (Bauer and Gaskell 2000) concerning contested innovation on platforms.

In fact the aim is to collect stories in the form of sequences of narrative fragments within the narrative boundaries that were previously identified. These narrative fragments concern actants performing actions on or to other actants. Within these fragments actants concern either Apple as the platform owner or other platform ecosystem members. The actions that are sought, concern actions by ecosystem members that push the boundaries of acceptable innovation on the iPhone platform, or actions by Apple, as the platform owner, that seek to control the extent of innovation on the platform. I also seek to identify when a particular event as narrative fragment takes place, in order to reconstruct the sequential order of a narrative. Finally, I aim to identify the events which trigger these actions and the mechanisms that enable them as means to explain these complex interactions. There is no need to collect additional contextual data concerning stories, as this does not play a role in sequence analysis.

Whilst narrative data is analysed as text, it can be originated from many different sources. Traditional sources include narrative interviews (Bauer and Gaskell 2000), literary texts (Bal 1985; Barthes 1977) and News Media (Van Dijk 1988), although narrative analysis has been applied to many other forms, such as music and images, both static and moving. Within organisational science and information systems narrative analysis is typically applied to interview data (e.g., Bartis and Mitev 2008; Sabherwal and Robey 1993), observational data (Corsaro and Heise 1990) and historical and archival records (Abbott and Hrycak 1990; Pentland and Reuter 1994). Intriguingly, Pentland in (Pentland and Reuter 1994; Pentland 1995; Pentland et al. 2010; Pentland et al. 2011) has used automatically generated electronic records, such as electronic invoices, as sources of narrative data for sequential analysis.

Given that the types of actants involved in my stories are so clearly defined, that is Apple as a platform owner, and apps developers and platform enablers as ecosystem members, it would seem reasonable that interviews with the respective parties would be an appropriate means of collecting stories concerning the phenomenon that I wish to research. However this did not come to pass for three reasons. First, access to the main actant, Apple was not possible, as the organisation is notoriously secretive. Second, many of the ecosystem members, who are involved in disputes over the extent of innovation on a platform, are small and hard to identify and approach through traditional sources. Finally, as I wish to compare a range of instances of the phenomenon, it would be extremely hard to identify enough stories given my constraints.

For this reason I examined alternative sources that would give ready access to a volume of stories concerning innovative acts on Apple's iPhone platform. For this, I turned to blogs.

Blogs as a source of narrative data

Davidson and Vaast (2009 : p.40) define blogs, formally known as Web Logs, as "web-based publications with reverse chronological order of dated entries, usually maintained and published with a blog authoring tool". They are of interest to me because certain types of blogs can provide regular commentaries on specific interest areas, such as news concerning mobile platforms including Apple's iPhone iOS.

The structure of the universe of blogs on the internet, otherwise known as the blogosphere, is quite complex and is dependent on the ownership of the blog site and the motivation for the blog. Blood (2004) notes that blogs may be used as: filters, or as reflections, on the observed world; as notebooks, or locations of longer, unfocussed essays; or as knowledge blogs for the creation and sharing of information. Mattson and Davidson (2009) describe the blogosphere as made up of: corporate bloggers, who use blogs to disseminate information about their company to the public and/or within the realm of the company; mainstream media bloggers, who use blogs as an alternative or supplementary channel for the information they disseminate; online news bloggers, who are independent of the mainstream and provide reportage and commentary on different aspects of the news; and independent bloggers, as individuals who wish to share their views and interests in the public sphere.

We are interested in those blogs reporting and commenting on developments amongst actors within the mobile services industry who have an interest in digital platforms. Fortunately, there is a well-defined, established group of bloggers reporting on these issues. Davidson and Vaast (2009) identify this group a "Tech Bloggers", who focus their commentary on technological innovation such as Web 2.0, digital music and multimedia apps, mobile technology and apps, and on the high tech companies, including Apple, Google and Microsoft, that facilitate this. Tech bloggers can be classified as a type of knowledge bloggers, as they create and diffuse knowledge about technological innovations, firms and products to a wide public. However as individuals they be in the employ of a corporation, an established mainstream media provider, an online news channel, or be acting independently. Following on from this, I define tech blogs, the type of blogs I use as a data source, as: "Web-based searchable publications sponsored by corporations, mainstream media providers, online news channels as well as independent actors that comment on technological innovation and the ICT industry".

The use of electronic sources such as blogs is becoming increasingly popular (Puri 2007) as means of data collection for social science research. The most popular use of this form of data collection is in virtual ethnography (Hine 2005), otherwise known as Netnography (Kozinets 2009). Within this genre of research methodology, blogs are becoming an increasingly important source of data, as can be seen in the burgeoning literature on methods (e.g., Hookway 2008; Wakeford and Cohen 2008; Ward 2006). This type of research is typified in organisational science by Schooneboom's (2007) ethnographies of anonymous work bloggers. Another genre of research methodology using online sources such as blogs is discourse analysis (Mautner 2005). This has had particular application in marketing research with respect to gathering customer intelligence, for example within the literature on tourism marketing (e.g., O'Connor 2008). There is increasing interest in the respected IS literature (Allen et al. 2006; Olson 2008) in the ethical aspects of data collection in electronic environments for academic research. However, it is surprising to note that there appears to be a paucity of actual research in major IS journal using these data sources (Kane and Fichman 2009). This all the more so in the case of research based on blogs as a source of data. Again, whilst there appears to be emerging research in information systems with regards to the ethical aspects of using blogs as a data source, e.g. Olson (2008), there seems to be little actual published research using data sourced from blogs. A notable exception is research led by Elisabeth Davidson, based on discourse analysis, regarding the reputation of technology bloggers (Davidson and Vaast 2009; Mattson and Davidson 2009; Vaast and Davidson 2008).

This hesitancy to use blogs as a source of data in IS research is curious, since they are a highly suitable secondary and tertiary source of information (Bar-Ilan 2005) for the reasons of relevance, quality and flexibility. First, regarding relevance, there are many blogs providing factual reports on corporate decisions and actions within the industries of interest to this research. Second, regarding quality, the quantity of different blog sources means that they can be crosschecked against each other and against online corporate sources of public information in order to assess the validity of the data that they contain. Third, with regards to flexibility, the online nature of blogs makes it easy for rapid data search, access and filtering. Blog entries are time stamped making it easy to assemble them into a sequential narrative thread. Finally, they are especially suited to the form of sequence analysis being conducted in this research. Whilst there is a risk in more contextually rich narrative studies that interpretation of blog data may be compromised by the fact that they are reports of reports of events, and potentially highly subjective in their own right, this concern is overcome in this study. This research is concerned purely with events in themselves, and all subjective contextual information is put to one side. The fact that the raw events used as data in this study are taken from different sources would add further credence to "the fact" that they happened.

In spite of its apparent suitability for narrative analysis, I am unaware, at this point in time, of any research using this methodological approach on data sourced as blogs in information systems or any other discipline. This is all the more surprising as narrative analysis has frequently been carried out across print equivalents of blogs, such as newspaper articles (Barnett 2005; Van Dijk 1988).

Collecting blog data

My task in collecting data was to find sequences of actions concerning interactions between Apple and its ecosystem members as they negotiate the boundaries of what can be innovated on Apple's iOS platform. I was looking for actions taken by ecosystem members as they attempt to innovate in such a way as to move the boundaries, and actions by Apple as it attempts to control these innovating actions. I aimed to collect this data as narrative fragments of one side performing some action on the other, which could later be stitched together in sequence to form whole stories concerning this phenomenon. I was hoping to find narrative fragments in the reportage of tech bloggers, who observe and publish electronically the goings on between members of Apple's iOS platform ecosystem. Prior investigation showed that typical blog entries would represent a single narrative fragment, often represented in the title of the blog entry, indicating an overall action being taken by one actant affecting another. If sufficient blog entries were collected, the narrative fragments could be assembled into a narrative. The task of data collection was simply to gather a sufficient corpus of blog entries concerning Apple's iPhone iOS platform over a period of time, so that stories could be identified and built.

The question arises as to which blogs to choose and then use as sources of narrative data. My concern was to be as neutral as possible in terms of the types of story I collected. In fact I wanted to collect as many stories as possible concerning the phenomenon. In order to do this I adopted the approach of Davidson and Vaast (2009) by using internet based technology aggregator Techmeme.com. Davidson and Vaast (2009) define an online aggregator as "web based functionality that selects webpages, such as blogs and online stories". As Davidson and Vaast (2009) explain, an aggregator "scrapes" information from blogs based on predetermined criteria, such as specified author or publisher. Whilst aggregators are biased in the sense that they search a predetermined list of blogs, the owners of aggregators attempt to keep databases of these lists as wide as possible. Furthermore the stories that it posts, are ordered by importance based on an algorithm that examines how many links there are to a story and how old it is. Naturally I choose aggregators that focus on tech bloggers.

Data was collected, as blog entries featured on Techmeme.com, over the five year period between January 2007 and December 2012 inclusive. This period of time provided a sufficient window in which to collect focussed stories for my analysis. The focus of data collection was on Apple's iOS platform, and the window of time from which data was sourced covers the platform’s genesis right up until the time of writing up this research. The decision as to whether to collect a blog entry as data was simply made on the basis of whether it discussed innovation with the iOS platform. References to relevant blog entries were stored in Microsoft Excel, a spreadsheet package which allows for the storage, tagging, referencing, searching and manipulation of textual data taken from blog entries. I continued to build and add to the corpus of data, until I exhausted my sources of information. In total I collected 4664 blog entries. At this stage I did not code the data.

Additional sources of data

Data extracted from blogs is intended to help build stories concerning the sequence of actions that Apple and its developers engage in as they contest permissible innovation on Apple’s iOS platform. In addition blogs will provide data to inform the explanation of these interactions in terms of the events which trigger tussles between Apple and its apps developers as well as the mechanisms which Apple employs in order to apply control to innovation occurring on its platform.

However additional data resources will be drawn upon in order to provide an understanding of the mechanisms that Apple uses to apply control. The content of blogs may well refer to the mechanisms that Apple is using without providing specific details of how those mechanisms work. To that end the specific Apple documentation, for example Developer Licence Agreements, which provide the rules by which apps developers must build apps, will be drawn upon as they are referred to in blog entries.

3.4.4 Identifying narrative themes within the data

In order to construct a narrative network and to carry out sequence analysis, the actions that make up different sequences need to be comparable. The actions that are observed in data need to be coded into a set of generic actions so that sequences can be made comparable. Whilst the boundary conditions, described earlier, help focus the development of describing specific types of actions, they are not sufficient. It is essential to develop coded actions that actually represent those that are observed in the stories that the data describes. The coded actions need to be formed in reference to actual stories or sequences of actions.

However the individual blog entries, which were collected as data in the previous stage, are not generally stories in their own right. Rather they are fragments of an overall story, as blog entries typically report the immediate events, at a particular point in time, concerning part of a lager narrative. Furthermore the data represents a set of unordered, unthemed fragments, only some of which are directly related to the phenomenon being studied.

Consequently, the data must be manipulated and combined into stories, or groups of coherent themed sequences of actions which represent the phenomenon to be researched, before developing a set of coded actions. It is not unusual to have to stitch together a set of stories from a set of fragmentary texts to enable narrative analysis, and this approach has precedence within the IS literature (Brown 1998).

This is done by carrying out three steps. First, relevant narrative fragments are identified and labelled from within the data. Second, these narrative fragments are assembled into themed stories. Last, attempts are made at filling any gaps found in the stories by searching for additional data. These steps are now described in more detail.

1) Identify and label fragments

The first step concerned going through the mass of collected data, identifying those blog entries that concerned the phenomenon to be researched, and labelling them, so that they could later be assembled into sequenced themed stories. The approach that was taken was to conduct content analysis (Bauer and Gaskell 2000) on the collection of blogs in order to interpret the qualitative textual data that they contained (Myers and Avison 2002; Silverman 2009).

Each blog entry was scanned, and matched against certain criteria to see if it contained a relevant narrative fragment. The criteria concerned the boundary conditions that were identified earlier on. The blog entry had to concern an action pertaining to altering or controlling the limits of innovation permitted on Apple's iOS platform. In addition this action must be carried out by one of the focal actors, either the platform owner, Apple, or a developer or platform enabler as a member of the platform ecosystem. If the blog entry fulfilled these criteria, it was then labelled or tagged in Excel with a named theme so that it could be identified later. The theme typically concerned the subject of innovation, so for example a particular application or a type of function.

2) Assemble fragments into themed stories

Once all the data had been scanned, and appropriate elements labelled, each set of themed blog entries were then extracted and summarised in tables as a sequence of events. Each table was built up of rows of narrative fragments, ordered sequentially in event time. The narrative fragments were extracted from the associated blog entry and summarised as short phrases identifying the focal actors and the action that was taking place.

3) Attempt to fill in narrative gaps

Finally where it was evident that stories in the themed tables were incomplete, additional data mining was carried out, by affecting a Google search for relevant blog entries for further information and narrative understanding to fill the gap. In this way, this approach is consistent with the concept of the hermeneutic circle (Myers 2004), whereby I continued collecting information for a narrative, until I was able to close discrepancies and gaps and I was able to make sense of the story.

Once tagged the blog entries could be manipulated enabling similar texts to be grouped, for the formation of stories. The decision was then taken as to whether a table containing a story consisting of a themed set of actions should be selected for coding. This was made on the basis as to whether it contained sufficient narrative detail, that is a beginning, a middle and an end (Barthes 1977), to merit further study. The stories that were identified and that contain sufficient narrative detail for further narrative network analysis are presented in the next chapter concerning empirical data.

3.4.5 Coding the data

The objective of coding the narrative fragments that form the themed stories, identified in the previous stage, is to facilitate the process of narrative analysis. By carrying out sequence analysis, I aim to: clarify the sequences of actions that I observe; facilitate the comparison and the identification of common sequences of actions; and to enable the discovery of common mechanisms that cause particular sequences of actions. In order to do this, the coding scheme needs to describe who is doing what to whom and when. The emphasis of the scheme will be on actions which push the boundaries of innovation, and those actions which constrain them.

Consequently it becomes clear that the respective elements of a narrative fragment (Hendricks 1972) need to be coded. These elements include: first and foremost the action; the actant carrying out the action; the actant upon which the action is directed; and finally the sequential order in which the narrative fragment occurs. Additional contextual factors that facilitate the interpretation of a narrative as it is constructed, such as motivation or setting, are not included. This is typical of sequence analysis which normally has a pure focus on events (Abbott 1992).

Before moving on to describing the coding scheme, it is necessary to consider a number of constraints that shaped the coding process. Five constraints were identified concerning: the boundaries on the events taking place; the granularity of the actions being studied; the start and stop points of the sequences; the perspective from which the sequences of actions were being narrated; and finally the ensuring of a narrative coherence to the coding scheme. These are now considered in turn.

First, regardless of a coding scheme, the possible codes are already constrained by the boundaries that were defined earlier for the sequences of actions that are being studied. For example the actants are prescribed as app developers and platform enablers who are attempting to extend the boundaries of permitted innovation on Apple's iOS platform, and Apple, as the platform owner who is attempting to control these actions. The actions to be studied, and hence action codes to be used, are limited to those that either attempt to extend or control the boundaries of innovation on Apple's iOS platform.

Second, the granularity of the actions to be studied, and codes to be used, is already determined to be at the macro level of commercial entities acting on each other, as they affect the terms of innovation on a platform, as reported in public discourse in the form of blog entries. This will result in a smaller lexicon of codes than would otherwise be the case if I was taking a more detailed view of action. However the benefit of a smaller lexicon of action codes is that it will limit the degree of variation in the sequence, making analysis of sequences easier (Sabherwal and Robey 1993). Caution must be exercised in selecting a level of granularity that on one hand guards against complexity, whilst avoiding oversimplifying the phenomenon that is being studied.

Third, the start and stop points in a sequence, during which actions that make up the phenomenon are observed, must be defined. This must be done in order to ensure consistency of data collection and analysis. In this study, the starting point is determined as when it is publically determined that a member of the ecosystem has attempted to move the boundary of permitted innovation. The finishing point is determined as the most recent publically recorded action that marks a steady state. These are reflected in the coding scheme.

Fourth, it is already determined that the perspective from which the stories will be built, will be as neutral as possible, so as not to bias a particular point of view, or certain section of the sequence. Consequently codes, representing actions, need to be as equally spaced across the potential space of actions as possible. The narrative perspective that is taken also effects the naming of the codes. For example one needs to consider whether to use the terminology used within the source data, i.e. the blogs, to label the codes, or whether to define the codes from first principles. Pentland (1995 : p.550) refers to informants' terminology as emic, and a research derived terminology as etic. Given the wide variety of terminology used across a range of different blogs and blog sources to describe similar events for different stories, an emic approach would be challenging and a less constraining etic approach was taken. An etic approach is also consistent with attempting to provide a narrative voice that is neutral.

Finally, the last constraint that was considered before coding the actions that make up the stories was the need to preserve a sense of coherence (Abbott 1992; Abell 2004; White 1980) across the sequences of actions. In this respect coherence refers to what Czaniawska (1997, 1998) terms a unity of action within a narrative. If coherence is maintained, then the risk that sequences of coded actions resemble a mere chronicle (Pentland and Feldman 2007), or just a set of events can be avoided. Coherence is maintained, if it is clear that the coded actions represent two sets of actors, negotiating the extent of the permissible boundaries of innovation within a platform and reaching some form of outcome.

The Greimas square as a semiotic tool to facilitate coding

A semiotic tool, known as the semiotic square, or the Greimas square, can help relieve a tension regarding the development of a coding scheme. This tension concerns the coding of actions seen in the actual narrative whilst respecting the constraints of the boundary conditions identified earlier.

According to the semiotician AJ Greimas (Greimas 1987), the semiotic square represents the elementary structure of signification. It emerged from the Paris School of Semiotics (Hébert 2007) and was developed by Greimas and Rastier (1968) as a tool to enable opposition analysis. The reason for this is that a primary analytical method employed by many semioticians, in decoding systems of signs to investigate social meaning, involves the identification of binary or polar semantic oppositions (for example us/them or public/private) in texts or signifying practices.

Along with other structuralists, Greimas (1987) claims that narrative progression, being composed of a system of signs, is organised by way of logical oppositions. The key contribution of the semiotic square is to enlarge the structural notion of binary opposition, by means of logic, such that the analytical classes arising from an opposition can be increased from two, to four and possibly eight and even ten (Hébert 2007). In so doing, Greimas shows that "the possibilities for signification are richer than the either/or of binary logic but that they are nevertheless subject to 'semiotic constraints'" (Chandler 2007). As a consequence of the expansion of terms that the semiotic square allows in analysing binary oppositions, it becomes possible to provide a richer analysis of narrative progression, whilst retaining semiotic rigour.

(Courtés et al. 1991) describe the semiotic square as visualising the logical structure of opposition. It contains a number of semantic terms (semes), which are logically related. Figure 3.3 illustrates a semiotic square, which does not contain any real opposing terms for analysis, but which contains labels, so that its underlying logic can be described.

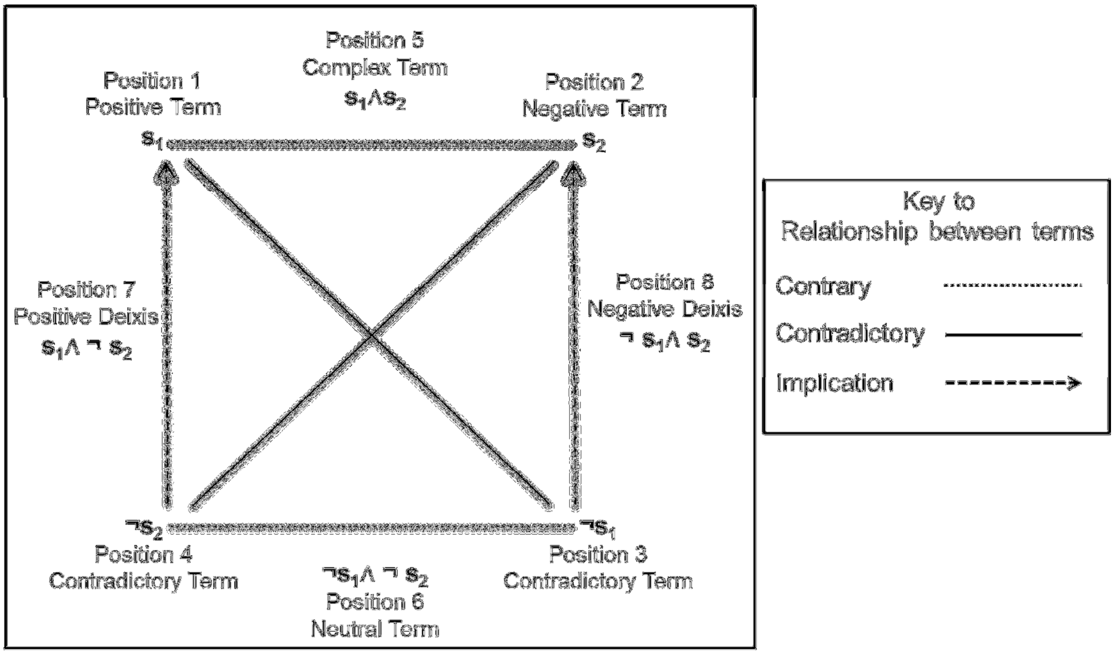


Figure 3.3 The relationship between terms in a semiotic square

(After Hébert (2007))

The semiotic square contains up to ten terms and meta terms in the ten labelled positions around the square. The most fundamental terms are s1 and s2, in positions one and two of the square. These terms form the dominant binary oppositions from which the rest of the square is constructed. The term s1 is typically thought of as being positive and its binary opposite, s2, is generally thought of as being the negative term.

However the semiotic square moves beyond two basic oppositions. It is possible to negate the two dominant opposing terms to produce contradictory terms. The third term, in position three, becomes not S1, and the fourth term, in position four, becomes not S2.

It then becomes possible to extend the four basic terms to eight or even ten, by considering metaterms. Metaterms are compound terms, which are synthesised from the four basic terms. The two key compound terms are known as the complex term S and the neutral term not S. The complex term S, in position 5, consists of a synthesis of not s1 and s2. The neutral term not S, in position 6, consists of a synthesis of not s1 and not s2.

In addition to S and not S, it becomes possible to synthesise additional compound terms, on the vertical sides of the square. Known as relationships of implication, they consist of s1 and not s2, in position 7, as well as s2 and not s1, in position 8.

Finally, some applications of the semiotic square attempt to synthesise the diagonal contradictory terms, s1 and not s1, position nine, as well as s2 and not s2, in position ten. These terms are not considered in this study, as they lead to metaterms that are impossible to realise in practice.

Following this outline of the logic of the semiotic square, it becomes possible to illustrate it with an example, provided in figure 3.4, based on terms identified by Hébert (2007 : p.43), concerning the binary opposition masculine and feminine.

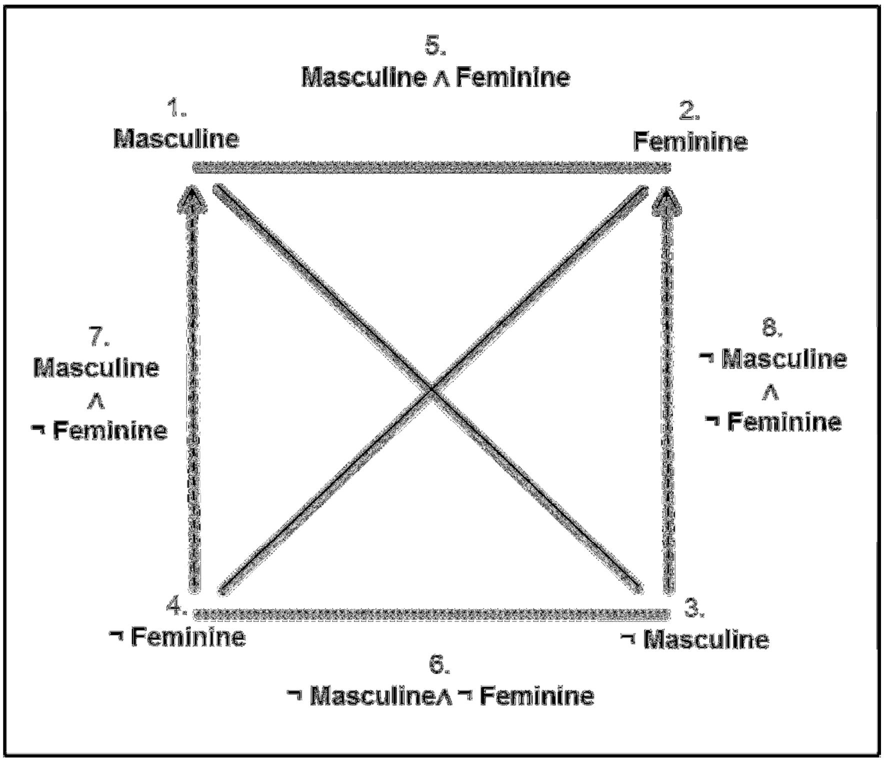


Figure 3.4 An example of a semiotic square with completed terms.

(After Hébert (2007))

The first terms that are defined are the dominant binary opposition, in positions one and two. The positive term is labelled Masculine and it's binary opposite is labelled Feminine. Following on from this, it become possible to define the contrary terms in positions three and four. These are simply Not-masculine and Not-feminine accordingly.

It is then possible to complete the labelling of the semiotic square by completing the metaterms. First, the complex term in position five is defined as Masculine and Feminine. Second, the neutral term in position six is defined as neither Masculine nor Feminine. Finally the relationships of implication, at the sides of the square, in positions seven and eight can be formed. These are defined as Masculine as well as Not-feminine, and Feminine as well as Not-masculine, respectively.

Now that the logic of an example binary opposition has been explored and expanded in a semiotic square, it becomes possible to illustrate it. Actual realisations of the terms and the metaterms are known as concepts, and are usually displayed within inverted commas. In semiotic analysis these concepts are found within actual texts, and are used to demonstrate the semiotic structure given by the terms within a semiotic square. It must also be noted that concepts are mere instantiations of terms and are not necessarily unique. It might be possible to label a term with several concepts. Figure 3.5 illustrates the terms indicated within the previous semiotic square with example concepts.

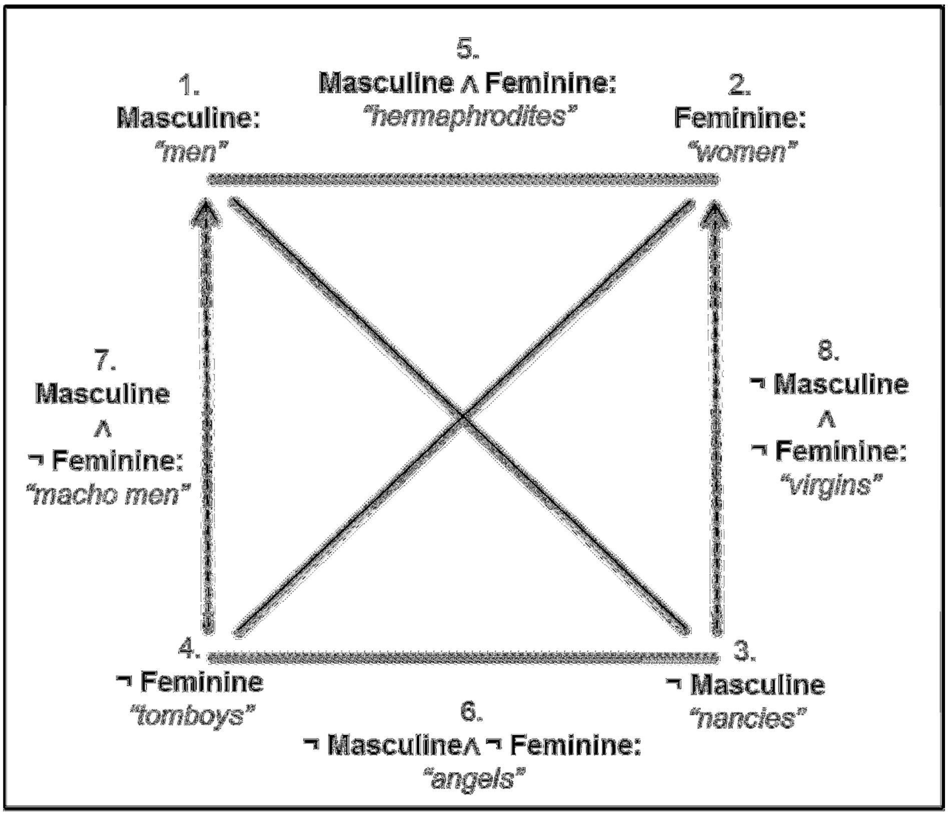


Figure 3.5 An example of a semiotic square with completed concepts.

(After Hébert (2007))

The terms Masculine and Feminine are straightforward to interpret and can simply be labelled with the concepts "man" and "woman". However the contradictory terms Not-masculine, in position three, and Not-feminine, in position four, require more thought. Concepts describing Not-masculine and Not-feminine are required to retain the basic characteristics of Masculine and Feminine respectively, but are contradictory such that they also have an absence of respective Masculine and Feminine traits. Following Hébert (2007) examples of these contradictory terms can be interpreted and labelled "effeminate" and "tomboy".

The complex term, in position five, must be realised by a concept that is both Masculine and Feminine at the same time. This is realised by the concept of "hermaphrodites". The neutral term, in position six, must be realised by a concept that is neither Masculine nor Feminine. Hébert (2007) interprets and conceptualises this as "angels", who are apparently androgynous.

Finally the terms of implication in positions seven and eight, Masculine as well as Not-feminine, and Feminine as well as Not-masculine, are sometimes conceptualised as the pure forms of the original binary opposition. Again, following Hébert (2007), the purest form of masculinity could be interpreted and labelled as "macho man" and the purest form of femininity could be interpreted and labelled as "virgin".

To conclude this example, it can be seen that the semiotic square enables the development and illustration of conceptual closure from a binary opposition that Frederic Jameson claims for it (Greimas 1987).

The use of the semiotic square as an analytical tool has spread far beyond literary studies (Greimas 1988) to encompass, for example, Political theory (Jameson 1983), Legal Theory (Jackson 1987), Understanding Corporate Language (Fiol 1989), the analysis of Computer games (Myers 1991), and the philosophy of science and technology (Haraway 1992).

In his forward to Greimas (1987), Frederic Jameson identifies two main uses of the semiotic square for analysis. The first is as a means of conceptualising, representing and visualising the logical structure of opposition in semiotic analysis. In this way it provides a clearer and more economical means to articulate relationships than through prose. The second is a heuristic tool. Rather than using the semiotic square as a means of representing what is already found, it can be used as a means of discovering what might be there, for example as a means of finding a logical coherent outline of a narrative within a text.

It is in this sense as a heuristic tool, facilitating the identification of representative events that make up narrative, that the semiotic square is used in this research. It was earlier claimed, that in order that a narrative is coherent and meaningful, it must involve a sense of transformation. Structuralists, such as Greimas (1987), claim that narrative progression is organised by way of transitions between oppositions. The semiotic square facilitates the identification and expression of the different semantic transformations and relationships between events as they undulate between binary opposites in a narrative. For example, the narrative of the life of

Christ, seen in terms of transitions from life, to death, and back to life (non death) again, through resurrection, is neatly illustrated by means of a semiotic square in Courtés (1991). In this sense, as Jameson suggests (Greimas 1987) the semiotic square offers "a virtual map of conceptual closure". In short it helps bring sense to narrative, by logically identifying a limited set of coherent terms and their relationship to each other, "an elementary structure of signification" (Greimas 1987), as narrative transitions within a text, which is precisely what is required of the coding scheme for this research.

However, as a semiotic tool, the Greimas square is not without critics. Greimas developed the semiotic square is part of the structuralist project (Greimas and Rastier 1968), in order to progress his work concerning structural narrative analysis

When the semiotic square is considered as part of structuralism, it exposes itself to at least two fundamental criticisms of structuralism. First, the approach does not allow for the contextualisation of meaning in terms of the social systems that construct the texts in the first place (Chandler 2007). Second, whilst structuralist approaches profess to be epistemologically objective in principle, they are often epistemologically subjective in practice. For example, any inexplicit oppositions that are "discovered", are a product of the mind of the interpreter rather than objectively identified from within the text (Chandler 2007).

An approach to circumvent this criticism is put forward by literary critic and political scientist, Frederic Jameson. In his forward to Greimas (1987), Jameson explains how, in his usage of Greimas' semiotic square (e.g., Jameson 1983), he brackets the structuralist ontology and simply uses it as an analytical device, which is well suited to analysing a system of meaning based on binary opposition.

A remaining criticism of the semiotic square, in that it leads to reductionist and programmatic decoding (Chandler 2007), is less important to this research, simply because it is being used a means to produce a compact formalised coding scheme, whilst remaining representative of and relevant to the text.

Developing the codes and coding in practice

The semiotic square provides a practical means by which a limited and coherent set of oppositional terms, and their relationships with each other, can be identified as codes in order to describe the events that occur in the stories found within the collected data. This helps overcome the tension, identified earlier, of seeking to express stories as sequences of generic actions, within the constraints of a limited lexicon whilst ensuring that the sequence remain faithful to the original stories.

The application of the semiotic square seems all the more suitable, when considering the fact that the actions both parties whose interactions I study, fall within binary oppositions. On the part of thirds party applications developers and platform enablers, it can be seen that they can either attempt to request that their innovations be accepted onto Apple's iOS platform, or they can attempt to force their innovation onto the platform. On the part of the platform owner, Apple can either allow an innovation onto its platform, or, in opposition to this, it can block an innovation from its platform.

The terms and concepts that were sought, by means of the semiotic square, in the collected texts were to represent narrative progression in the form of actions. In this way the semiotic square was being used for a syntactic or dynamic analysis for the sequencing of semantic values.

Two semiotic squares were constructed. One was built to represent the innovating actions carried out by third party developers and platform enablers as the came to terms with the binary opposition as to whether to request or to force their innovation on to a platform. The other was constructed to represent the controlling actions carried out by Apple, as the platform owner, as it navigated the binary opposition it faced in terms of allowing or blocking innovation.

In total two semiotic squares could potentially allow for a total of sixteen terms and associated concepts, which would lead to a large lexicon of coded actions. In order to keep the complexity of analysis within reasonable bounds, it was decided to restrict the lexicon to a total of eight terms, four for the controlling platform owner, and four for the parties attempting to innovate on a platform. This decision proved convenient, as (Hébert 2007) identifies the minimum number of terms for a complete semiotic square as four. Consequently, two dominant and two contrary terms were identified in each semiotic square based on the respective binary oppositions that were identified for each of the two players.

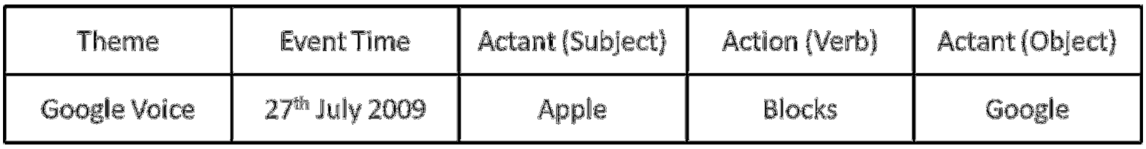
Following this, concepts were identified for all of the terms on both semiotic squares. Once the semiotic square was completed, the concepts would go on to become the codes used in sequence analysis. It was therefore essential that the concepts represented the key actions seen in the stories formed from the blog entries. The process of identifying these concepts was highly iterative involving a combination of the logic of the terms in the semiotic squares, familiarity of the narrative texts and a degree of intuition. The process continued until four concepts in each semiotic square were identified that represented a coherent narrative flow of the actions carried out by both parties as described in the texts. The thinking behind this effort was both interpretive and subjective. Interpretive because identifying concepts that fitted the logic of the semiotic square required some imagination. Subjective because, whilst the means of identifying concepts attempted to follow a rigorous logic, there was no objective measure as to whether they were truly representative of the key actions from within the text.

Nevertheless, two semiotic squares were constructed with a total of eight concepts, which enable a set of codes that are compact, coherent, judged to representative and logically explained.

The previous step had built up tables containing stories. Each table contained narrative fragments, extracted from blogs and put in sequential order. In addition each fragment contained a brief worded description, based on the key phrases from the blog entry, concerning the main actants and the action that was taken.

As such, each of these narrative fragments form the basis of an event which contribute to the sequence analysis using narrative networks that is to be performed. However, before the sequence analysis can be undertaken, the narrative fragments need to be coded using the coding scheme developed using semiotic squares.

In order to do this, each narrative fragment was analysed and coded in the following way. First, it was labelled according the theme of the story with which it takes place. Second, it was given a time stamp, based on the date of the originating blog entry, which broadly accorded to when the action took place. The remainder of the fragment is coded in the order of who is doing what to whom, in the order of subject, verb and object. The subject and object concern the actants involved, and are coded according to their given name, for example Apple or Adobe. The verb consists quite simply of the action that occurs in the narrative fragment, and is coded according to its closest match from the output of the semiotic squares. The structure of a coded narrative fragment is illustrated in figure 3.6.



Theme	Event Time	Actant (Subject)	Action (Verb)	Actant (Object)
Google Voice	27th July 2009	Apple	Blocks	Google

Figure 3.6 Structure of a coded narrative fragment

(Own Figure)

We then summarised all the coded stories by listing each as a column in a single table. Each column was composed of sequentially ordered coded narrative fragments relevant to a particular themed narrative.

Each column therefore represented a sequence of generic actions concerning the negotiation of the boundaries of innovation on Apple's iOS platform. This facilitates a dynamic syntactic analysis of actual sequences of actions, coded according to the semiotic squares. This analysis is played out by comparing different sequence of actions graphically on narrative networks. The construction of these narrative networks is the subject of the next step.

3.4.6 Constructing and analysing a narrative network

On first inspection, it might seem that the construction of narrative networks should be straightforward. After all, the basic building blocks, in the form of narrative fragments are all identified, linked to themes and sequenced. All it would take is to display them graphically and link the appropriate sequential relations amongst them with ties. However, there are subtleties that need to be observed in order that narrative networks are constructed in such a way as to maximise the ease of analysis. For this reason this sections combines both the construction and analysis of narrative networks.

Before constructing narrative networks, there is a need to understand what type of analysis is to be undertaken, which will in turn influence the type of narrative networks that are constructed. Three broad forms of analysis are to be undertaken. First, I intend to clarify the sequences of actions that I observe. Second, I seek to compare these sequences and attempt to identify common sequences and patterns of actions across them. In this way I hope to provide a base level of explanation (Pentland 1999a) to facilitate the third level of analysis. This third and final level concerns the discovery of common mechanisms that contribute to particular common patterns of actions that are observed. The three levels of analysis are all cumulative, such that the output of one, contributes to the development of the next

Narrative networks were chosen as a methodological and analytical tool for their simplicity and flexibility. This is in contrast to rigorous but constraining methods such as grammatical pattern matching techniques (Pentland and Reuter 1994) or optimal string matching and clustering (Sabherwal and Robey 1993). This is not to say that narrative networks lead to superficial analysis. The degree of formality in their application, and the constraints upon the level of analysis, depends upon the research design (Pentland and Feldman 2007). For example, if the sample size of sequences of events is sufficiently large, then sophisticated quantitative analytical methods can be applied. Pentland et al (2010) refer to rigorous statistical methods for pattern matching techniques (Theodoridis et al. 1999), such as n-grams (Manning et al. 1999), principle components (Dunteman 1989) and hidden Markov models (Ching and Ng 2006).

However, the number of relevant sequences collected for this research represents a relatively small sample size, such that these statistical methods are not useful. Nevertheless, the level of analysis using narrative networks will be extensive. Each of the three levels of analysis was approached as follows.

First I plotted narrative networks for each of the stories that were identified in the data. This was done for each separate narrative, by plotting the all the possible narrative fragments, that can be used, as nodes. Each narrative plot was taken individually, and the nodes as narrative fragments, which were used in the narrative, were linked in sequence using arcs in order to reveal a narrative network. The outcome of this stage was a set of narrative networks, representing each of the stories, plotted in such a way as to clarify the complex sequence of actions that were first observed in the blog data.

The second level of analysis examines narrative networks for common underlying patterns of actions across different sequences. Had there been a sufficiently large sample size, it would have been possible to carry out statistics to test the hypothesis as to whether sequences are similar or not, using chi-squared tests. However, this was not the case, and I carried out the analysis using the following steps. First, I counted the number of ties between each pair of nodes, across the different stories, in order to identify the frequency of transitions between certain events. Second, I constructed a first order Markov model summarised as a matrix. The matrix was made up of frequencies of transitions between events, which is then normalised, such that the transition probabilities sum to one. Finally, I constructed a "normalised" valued directed graph (Abell 1987) from the data in the matric in the form of a narrative network, where the nodes represent events rather than states. The thickness of the ties between the nodes represented the values in the normalised matrix, such that the most common narrative paths to stand out.

Whilst, as Pentland (1999b) notes, these directed graphs can be interpreted as finite state grammars (Chomsky and Miller 1958), analysis is kept at the level of visualisations (Langley 1999) as a means to identify and interpret commonalities and differences amongst sequences. The graphical form of the narrative network provides a means of interpreting commonalities and differences amongst sequences of events by inspection.

As well as simply identifying clusters of sequences which are broadly similar, a number of other traits can be found within the graphically represented sequences. For example it becomes possible to examine the density of the nodes, which is the inter-connectedness of nodes, both in terms of the degree of inward ties converging or going into a node and also of outward ties diverging or going out. The density of a network is an indication of the flexibility for the possibility of a range of different actions. When a network is not dense, that is when it is sparsely connected, it appears that the structure of potential action has become rigid and that it might be highly controlled. The analysis of density of ties can become more sophisticated in order to identify centrality, or bottlenecks of nodes, that are traversed across a range of different sequences. Another opportunity for analysis is to identify the reachability or distance of particular nodes. This leads to an understanding of how far apart certain actions elapse, and why this might be the case. This type of understanding follows on from an analysis of the actual sequencing of actions. A final area that can be considered is how the density of events is altered by the unfolding of time and distance. It is of interest to understand how the density of different types of action varies over these dimensions as this may give clues as to the underlying mechanisms that facilitate change.

The third and final stage of analysis concerns the identification of mechanisms that that contribute to the patterned sequences of actions that are observed. This involves the application of the theoretical frameworks to the analysis of the previous stages in order to provide explanations as to why the patterns of actions are occurring in the way that they do.

3.5 Description of narrative data

This section serves to present a brief description and provide examples of the narrative blog data that was captured.

The task of collecting data from web logs, or blogs, was simply to gather a sufficient corpus of blog entries concerning Apple's iPhone iOS platform over a period of time, so that stories could be identified and built. This corpus of data would provide the basis for identifying specific stories concerning tussles between Apple and its apps developers over the permitted boundary of innovation on the iOS platform.

Blog data was extracted from Techmeme.com, an aggregator of technology blogs. Blog entries concerning Apple's platform iOS, the iPhone, the iPad and iOS apps, and written during the period January 2007 until December 2011 were mined from Techmeme.com.

The details of relevant blog entries were stored in a Microsoft Excel file for further analysis. The basic fields that were recorded were: the date that the blog was written, the author and source (or blog name), the title of the blog entry, a summary of the entry, and finally the URL of the blog entry so that it could be returned to at a later time. Table 3.1 provides examples of summaries of the blog entries that were recorded. The first nine entries, a middle section (numbered 2241-2245) of five entries and the final five entries (numbered 4660-4664) are highlighted in the table.

#	Link Date	Author and Source	Headline
1	09/01/2007	longofest / MacRumors:	Apple Announces The iPhone
2	09/01/2007	Apple PR	Apple Reinvents the Phone with iPhone
3	10/01/2007	Mike Clendenin / EE Times:	Apple iPhone fuels speculation on design wins
4	10/01/2007	Gizmodo:	Gizmodo iPhone Hands On Part Deux: Why Isn't it White and Other Questions
5	10/01/2007	Mary E. Tyler / Infinite Loop:	iPhone is awesome (restrictions apply)
6	10/01/2007	Tristan Louis / The TNL.net weblog:	The iPhone is here [The TNL.net weblog]
7	10/01/2007	Ryan Block / Engadget:	The iPhone is not a smartphone
8	10/01/2007	Dan Lurie / The Unofficial Apple Weblog:	iPhone Will Not Allow User Installable Applications
9	11/01/2007	Arnold Kim / MacRumors:	Apple iPhone Apps Coming, but Limited
………			
2241	24/12/2009	Erick Schonfeld / TechCrunch:	Google Voice Is Coming Back To The iPhone Via The Browser, Thanks To VoiceCentral
2242	25/12/2009	TJ Luoma / TUAW:	Is Google Voice available for the iPhone?
2243	25/12/2009	Vladislav Savov / Engadget:	Want to connect your iPhone and Bluetooth keyboard? There's a (jailbroken) app for that (Update: video!)
2244	26/12/2009	MG Siegler / TechCrunch:	Flixup Brings Its Movie Tweet Aggregator To The Web For The Holiday Movie Season
2245	27/12/2009	Boy Genius Report:	GV Mobile 2.0 first look
……..			
4660	28/12/2011	Mike Butcher / TechCrunch Europe:	Will UK Prime Minister get an iPad app for work? Unlikely. Here's why.
4661	29/12/2011	Jack Purcher / Patently Apple:	Apple Envisions Face & Presence Detection Security for iOS Devices
4662	29/12/2011	Matt Brian / The Next Web:	Report: 300 million users now access Facebook via its mobile apps
4663	30/12/2011	Matt Brian / The Next Web:	“Official” Siri app hits the Android Market, highlights Google's marketplace issues
4664	31/12/2011	Electronista:	Apple sends takedowns to stop pirate-friendly iOS apps

Table 3.1 Example blog entries in the corpus of data.

Over 4664 blog entries were recorded chronologically in Excel covering the period running up to the launch of the first iPhone in early 2007 to the time that this thesis was written up in early 2012. A simple histogram is presented in figure 3.7 which provides a profile of the number of relevant blog entries per quarter year of the period from which data was sourced.

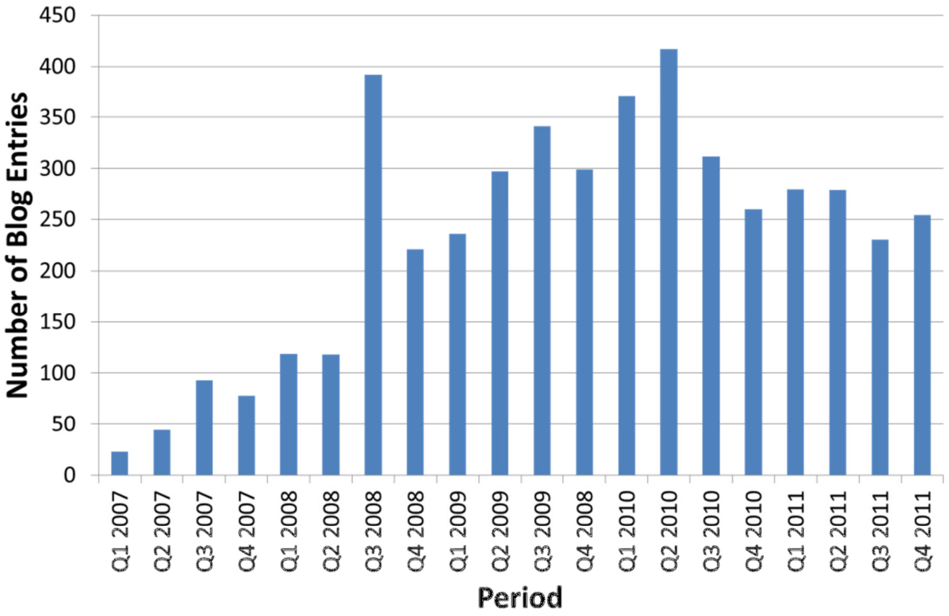


Figure 3.7 Distribution of blog entries over the period of analysis.

The number of blog entries concerning the iOS platform, the iPhone and the iPad, and iOS applications show a steady rise over the period from the first quarter of 2007 to the end of the second quarter of 2010, before falling away to what might be an approximate steady state over the rest of 2010 and 2011.

Within the corpus of data, blog entries were recorded from over 700 blog sites. However, 20 blog sites accounted for over 50% of the 4664 blog entries that were recorded. Table 3.2 lists these 20 blog sites in descending order of the number of blog entries that they contribute to the corpus of data. The table lists each blog site, the corresponding URL of the blog, the number of blog entries contributed to the corpus of data, the percentage of the total number of blog entries that the respective contribution makes, and finally the cumulative percentage of the blog sites to the total as the list descends. The list is dominated by the major tech blog sites that have an interest in Apple and consumer technology.

#	Blog	Associated Blog URL	Count	%	Cum %
1	TechCrunch	http://techcrunch.com	573	12.8%	12.8%
2	MacRumors	http://www.macrumors.com	373	8.3%	21.1%
3	VentureBeat	http://digital.venturebeat.com	152	3.4%	24.5%
4	AppleInsider	http://www.appleinsider.com	137	3.1%	27.5%
5	Engadget	http://www.engadget.com	133	3.0%	30.5%
6	Gizmodo	http://gizmodo.com	123	2.7%	33.2%
7	GigaOM	http://gigaom.com	109	2.4%	35.6%
8	ReadWriteWeb	http://www.readwriteweb.com	102	2.3%	37.9%
9	9 to 5 Mac	http://www.9to5mac.com	77	1.7%	39.6%
10	Silicon Alley Insider	http://www.alleyinsider.com	77	1.7%	41.3%
11	CNET News	http://news.cnet.com	76	1.7%	43.0%
12	Macworld	http://www.macworld.com	55	1.2%	44.3%
13	The Next Web	http://thenextweb.com	51	1.1%	45.4%
14	paidContent	http://paidcontent.org	46	1.0%	46.4%
15	Bits	http://bits.blogs.nytimes.com	43	1.0%	47.4%
16	Infinite Loop	http://arstechnica.com	41	0.9%	48.3%
17	New York Times	http://www.nytimes.com	40	0.9%	49.2%
18	TUAW	http://www.tuaw.com	40	0.9%	50.1%
19	Daring Fireball	http://daringfireball.net	38	0.8%	50.9%
20	Gadget Lab	http://www.wired.com	37	0.8%	51.7%

Table 3.2 Top contributing blog sites to the corpus of data.

Over 1200 individual authors were found to have contributed to the blog entries contained within the corpus of data. Just over 20% of the 4664 blog entries were written by just 11 authors. Table 3.3 lists these 11 authors in descending order of the number of blog entries that they contribute to the corpus of data. The table lists each author, the corresponding URL of the blog that they are associated with, the number of blog entries contributed to the corpus of data, the percentage of the total number of blog entries that the respective contribution makes, and finally the cumulative percentage of the blog sites to the total as the list descends. It is interesting to note that this list is dominated by authors from technology blog site TechCrunch.com and blog sites such as MacRumors.com and DaringFireball.net that specialise in commenting on Apple.

#	Author	Associated Blog URL	Count	%	Cum %
1	MG Siegler	http://techcrunch.com	200	4.5%	4.5%
2	Arnold Kim	http://www.macrumors.com	110	2.4%	6.9%
3	Leena Rao	http://www.techcrunchit.com	88	2.0%	8.9%
4	Jason Kincaid	http://techcrunch.com	85	1.9%	10.8%
5	Dan Frommer	http://www.eff.org	78	1.7%	12.5%
6	Eric Slivka	http://www.macrumors.com	78	1.7%	14.2%
7	Erick Schonfeld	http://techcrunch.com	77	1.7%	15.9%
8	Sarah Perez	http://techcrunch.com	55	1.2%	17.2%
9	Robin Wauters	http://techcrunch.com	53	1.2%	18.4%
10	Philip Elmer-DeWitt	http://tech.fortune.cnn.com	45	1.0%	19.4%
11	John Gruber	http://daringfireball.net	41	0.9%	20.3%

Table 3.3 Top contributing authors to the corpus of data.

3.6 Outline of other sources of data

Whilst data extracted from blogs is intended to help build stories concerning the sequence of actions between developers and Apple, additional sources of data need to be drawn upon in order to provide an understanding of the mechanisms that Apple uses to apply control. To that end two types of Apple documentation is draw upon to inform this later analysis. The first concerns the iOS Developer Program License Agreement that Apple puts in place between itself and third party developers seeking to innovate on iOS. The second of the two documents is the App Store Review Guidelines.

The iOS Developer Program License Agreement details the obligations and responsibilities of third party developers as members of the Apple Developer Program. In order to develop applications for iOS, developers must join the Apple Developer Program, and as part of this developers must agree to the terms and conditions of the license agreement. In return Apple awards developers a limited license to use Apple software to develop and test applications on the terms and conditions of the agreement. The document sets forth many of the rules concerning the nature of iOS applications and the means by which they are innovated to which a developer must adhere in order that their application is distributed from the App Store.

The App Store Review Guidelines is an additional document, which provides specific rules concerning the development iOS applications by which apps are judged in the App Store Review process. Apple recommends that developers make sure that their apps comply with these guidelines before they submit their innovations for review.

Unfortunately access to these documents presents the researcher with a dilemma. Apple does not make these documents publically available. It is possible to gain access to these documents by registering and joining the Apple Developer Program. However under section 10.4 of the iOS Developer Program License Agreement, Apple state that:

"You may not issue any press releases or make any other public statements regarding this Agreement, its terms and conditions, or the relationship of the parties without Apple's express prior written approval, which may be withheld at Apple's discretion."

Section 10.4 presents a condition which makes it difficult for the researcher to join the developer program, and then use Apple's documents for research and analysis in the public domain. This issue was circumvented by the presence of "leaked" versions of these documents made available on the internet.

A further property of both the iOS Developer Program License Agreement and the App Store Review Guidelines is that they are subject to change as Apple introduces new policies and changes existing policies. These changes are of interest because they precipitate changes in the ways that Apple's mechanisms of control are triggered. To that end different versions of these documents were sought on the internet. Table 3.4 shows the various different versions of the documents that were used and provides links to references where they can be accessed at the time of writing.

iOS Developer Program License Agreements	
1	iOS Developer Program License Agreement (Revision 20/10/08)
2	iOS Developer Program License Agreement (Revision 17/3/09)
3	iOS Developer Program License Agreement (Revision 22/1/10)
4	iOS Developer Program License Agreement (Revision 9/9/10)
5	iOS Developer Program License Agreement (Revision 15/2/11)
6	iOS Developer Program License Agreement (Revision 5/7/11)
App Store Review Guidelines	
1	App Store Review Guidelines (Revision 10/09/2010)
2	App Store Review Guidelines (Revision 15/02/2011)
3	App Store Review Guidelines (Revision 09/06/2011)

Table 3.4 Apple's documentation used as sources of data

3.7 Chapter summary

This chapter describes the methodology employed within this thesis. The task of methodology in this research is to enable the studying of series of events so that a process theory can be built. To that end, the overall methodology is introduced as sequence analysis, within the broader tradition of narrative analysis. The specific methodology is based on narrative networks (Pentland and Feldman 2007), and the means by which it is implemented in this research is explored in detail. First the focal phenomenon, sequences of action indicating instances of contested within the iOS platform ecosystem, are described. Second the narrative perspective, detailing the resolution of events that is studied is explained. Third, the sourcing and collecting of empirical examples of sequences of actions from blogs is both justified and detailed. Fourth, the method used for identifying narrative themes within the data is set forth. Fifth, the means by which a compact and logical coding scheme is identified using semiotic squares is explained. Finally the approach to constructing and analysing narrative networks from the empirical data is described. This is then followed by a brief description of the range and statistics concerning the narrative data that was collected for this research. The chapter closes by detailing the sources of data used to explain the mechanisms that Apple employs to control platform innovation.

4 Theoretical Framework

The overall objective of this dissertation is to develop a description and explanation of the dynamics of the control of innovation on digital platforms. To that end, this chapter presents a theory of organisational control which is used as basis to explain how the innovation of platform complements is controlled by platform owners. It starts with a brief and broad review of how control is theorised in the management literature before focussing on two broad theories. The first concerns Ouchi's theory of formal control (1977; 1978; 1979), and the second concerns Beniger's model of control (1986). The chapter finished by summarising a model of formal control that is applied to empirical data later in this thesis.

Amongst organisational scholars, there are numerous definitions of organisational control. For example Kling and Iacono (1984) define control as "actions by one actor or group to influence other actors or groups to comply with their preferences". Kirsch (1997) defines control as "all attempts to ensure individuals in organisations act in a manner that is consistent with meeting organizational goals and objectives". However I adopt the definition given by Beniger (1986) on account of its brevity and simplicity. His broad definition of control is "purposive influence toward a predetermined goal".

In its broadest sense, control has proved challenging to conceptualise using social theory (Gibbs 1989). Numerous interpretations have linked notions of control with that of power (Clegg 1989; Etzioni 1965; Foucault and Gordon 1980; Lukes 1981). Of relevance to this study is the concept of organisational control, which is traditionally associated with the work of Max Weber (1947). Weber explored control as a process of creating and monitoring rules enforced within hierarchical authority, thereby limiting employees' activities by regulating patterns of interaction. Of further relevance, is the work of organisational scholars towards the latter half of the 20th century such as Thompson (1969) who theorised organisational control as a cybernetic process of testing, measuring and providing feedback.

4.1 A theory of formal organisational control

Organisational scholars, such as William Ouchi, have developed these theories yet further in order to identify how organisations manage control. Ouchi (1979) theorises that organisations are able to assert control through market mechanisms, bureaucratic mechanism and social mechanisms. Market mechanisms deal with the control problem through their ability to precisely measure and reward individual contributions. Bureaucratic mechanisms rely upon the evaluation of workers performance combined with a socialised acceptance of common objectives. Finally, social mechanisms, termed clan control, rely upon socialisation processes in order to eliminate goal divergence between individuals. The actual type or mix of control mechanisms used depends on the context of the organisational setting (Kling and Iacono 1984). Ouchi (1979, 1980) theorises that the applicability of these broad control mechanisms hinges on performance ambiguity, or variety of ways that a task can be carried out, and goal incongruence, or lack of agreement in terms of what the output of an activity should be. In this way he suggests that: market mechanisms are suited for activities where there are few ways of carrying out a task, and that the output of that task is clear; bureaucratic mechanisms are appropriate for environments where there is a moderate level of performance ambiguity and goal incongruence; and that social controls, centring on establishing habitus (Bourdieu and Nice 1977) in terms of shared values, beliefs and rituals, are best suited to contexts when there are many way of achieving a goal, and there is little agreement as to the specifics of what the goal should be.

Activities concerning the development of complex information systems are generally thought of as having low performance ambiguity and goal incongruence. To that end most IS uses of Ouchi's theory of organisational control (e.g., Kirsch 1997) have focussed primarily on formal bureaucratically driven modes of control or on informal socialisation based forms of control. Formal modes of control involve a cybernetic process of monitoring, evaluating and rewarding performance and are highly dependent on access to information (Eisenhardt 1985). Informal modes of control are achieved by people strategies to ensure that individuals have understood and internalised organisational goals, thereby minimising the divergence of their preferences.

4.1.1 Formal modes of organisational control

In his historical treatment of how organisations responded to the crisis of control that was brought about by industrialisation in the 19th century, Beniger (1989) draws on Weber (1947). He identifies the introduction of bureaucratic and rational forms of organisation as the structural means by which organisations addressed this crisis of control. Bureaucracy is “the generalised means to control all large social systems, tending to develop whenever collective activities need to be coordinated toward some explicit and impersonal goal" (Beniger 1989 P390). Beniger goes on to state that the purpose of bureaucracy is capability and capacity, through organisational means, to process greater quantities of more complex information and consequently to control more effectively. Part and parcel of this process of bureaucratisation is the structural organisation of the division of labour, definition of responsibilities, hierarchical authority, and functional specialisation including specialised decision and communication functions. Weber identified the process of rationalisation as being complimentary to the organisational form of bureaucracy in terms of enabling organisational efficiency and effectiveness. The broad aim of rationalisation was to further increase the efficiency of bureaucracy by decreasing the amount and simplifying the nature of information to be processed. This process of rationalisation involves the abstraction of relevant information, and the destruction of unnecessary information, in order to facilitate its processing as is exemplified in the form of statistically driven scientific management (Taylor 1911).

Modern characterisations of formal management control (Eisenhardt 1985; Kling and Iacono 1984; Nidumolu and Subramani 2003; Ouchi 1979) draw on Weberian notions of the bureaucratic organisational form. In this way organisations are seen as formal, rationalised, and information based task systems. Organisations are structured hierarchically bestowing authority on managers. Authority enables managers to evaluate the effectiveness of subordinates through explicit reporting systems. Furthermore, managerial authority bestows the power to enforce discipline and punishment on one hand and rewards on the other, ensuring a uniform approach that enables coordination and for goals to be met. It is assumed that information flows up and control flows down the hierarchy, and that there is an information asymmetry such that actors higher up hierarchy have access to more information. This asymmetry is exaggerated by the assumption that workers are isolated in their activities and have little contact with their peers. The information required for their task completion is constrained in rules, which are enforced by managerial authority. Mechanisms of control require personal surveillance and direction by supervisors, which is enabled by information systems that can monitor employee behaviour. However, in many organisational environments, this has to be balanced with a degree of autonomy to enable creativity and resourcefulness of workers.

Theories describing formal modes of organisational control are employed in this thesis for two reasons. First, it facilitates theoretical focus. Second, formal rather than informal modes of control seem to predominate the control of innovation in digital platforms. The reason for platform owners' focus on formal modes of control comes down to challenges they face in terms of the speed, scale and scope of digital platform innovation. On large digital platforms, applications are developed in large volumes and at great speed. The range and variety of applications in terms of form, functionality and quality is considerable. Furthermore the number, range and variety of developers is also large. Given this tsunami of innovation that digital platforms invite, platform owners would face a crisis of control (Beniger 1986) without rationalised and efficient mechanisms of control. Finally, central to the application of formal strategies is the use of information systems in the exercise of control (Beniger 1986), and this provides a basis for discussion later in this thesis.

4.1.2 Control as a process of regulating behaviours and outcomes

Despite the structural basis of bureaucratic organisation, and despite the fact that one of the functions of formal organisation is that of control, the theory of organisational control is processual in nature (Ouchi 1977). In this way control is considered to be "the regulation of performance through the establishment of management processes to guide activities" (Nidumolu and Subramani 2003, P162). In this way it is the processes of control of platform innovation that are studied in this thesis. Ouchi (1977, P97) determines that "in controlling the work of people and technologies there are only two phenomenon that can be observed, monitored and counted: behaviour and output". It is the consideration of behaviour and output that form the basis of the formal aspects of his theory of control.

Ouchi's theory focusses on two formal control strategies, behavioural control and control of outcome. Each strategy is dyadic (Kirsch 2004) in nature and concerns control between two entities, the "controller", who is exercising control, and the "controlee", who is the target of control. Furthermore the two strategies serve different purposes and can occur independently of each other (Ouchi and Maguire 1975).

Behavioural control regulates activities by prescribing specific behaviours to be adopted by a controlee for the execution of a task such that a desired outcome is achieved (Nidumolu and Subramani 2003; Ouchi and Maguire 1975; Tiwana and Keil 2009). The ability to prescribe appropriate behaviours depends on whether the means-ends relations of an activity are known and whether appropriate instruction, or task programmability, is possible (Eisenhardt 1985; Ouchi and Maguire 1975). When this is the case specific rules can be documented which prescribe repeatable methods and procedures that the controlee must adopt for accomplishing outcomes. Behavioural control is based on surveillance, so that controllers can reward or discipline controlees based on the extent to which they follow established procedures.

Output control is based on the control of performances of tasks by judging outputs or outcomes, without regard to the process by which the outputs are achieved (Nidumolu and Subramani 2003; Ouchi and Maguire 1975; Tiwana and Keil 2009). This strategy of control requires the pre-specification (Tiwana and Keil 2009) of what the controlee should accomplish. It is against the achievement of this specification that the controlee is judged and ultimately rewarded. Clearly those outputs and outcomes that involve subjective judgement are less amenable for control, than those that can be determined objectively (Nidumolu and Subramani 2003).

4.1.3 Mechanisms for implementing behavioral and outcome control

Eisenhardt (1985) considers the processes of both behavioural and outcome control to consist of three basic elements: measurement; evaluation; and rewards and sanctions. Measurement requires that the desired execution or the results of tasks are known in advance and pre-specified by controllers, so that actual outcomes and behaviours are measurable. Evaluation refers to how aspects of the execution or results of tasks are assessed and judged. Both the elements of measurement and evaluation require the exchange of information between controller and controlee. Finally rewards and sanction are linked to the controlee achieving, or failing to achieve, the outcomes or behaviours that were initially specified by the controlling party.

The subdivision of the process of formal control into three basic elements is useful when considering the mechanisms that are required for implementing of both behavioural and outcome control (Kirsch 1997). Mechanisms are tailored to the differing ways that behavioural and outcome control makes use of these elements. The mechanisms used to invoke behavioural control clearly focus on the measurement and evaluation of how controlees carry out a task accompanied by appropriate reward mechanisms. Similarly the mechanisms used to invoke outcome control focus on the measurement and evaluation of the result of the task carried out by a controlee, which is also accompanied by appropriate reward mechanisms. In terms of standard information systems implementations then development methodology, direct observation through client personnel being present on vendor premises, weekly progress reports, periodic meetings and conference calls are typical behavioural control mechanism. Similarly functional specifications, target implementation dates and software testing are frequent outcome control mechanisms (Choudhury and Sabherwal 2003). The mechanism of control that is studied in this research is the process the platform owner uses to select which platform complements are admitted into a platform ecosystem, and which are not.

4.1.4 Informational requirements for the mechanisms of control

Each mode of formal control and the mechanisms through which they are implemented have informational requirements (Ouchi 1979). These informational requirements concern how the mechanism is communicated to controlees and how the results of these mechanisms are fed back to the controlling parties.

For behavioural control to be implemented effectively (Ouchi 1977) the requirements are as follows: first, the task must be able to be deconstructed or programmed as a set of activities and behaviours, also known as task program (Eisenhardt 1985); second, the controlling organisation must have knowledge and agreement of what that task program is and the mean-ends relationships that it entails, so that it can be defined clearly, and subsequently monitored effectively; third, the task program must be communicable and communicated to the controlees; and finally, the way the task program is performed by the controlee must be observable and measurable.

For outcome control to be properly implemented the requirements are as follows: first, the output must be specifiable in terms of a quality such as performance or features (Ouchi 1979). For this to happen the controlling party must have appropriate knowledge and ability; second, the specification must be communicable as clearly stated goals to the controlees (Eisenhardt 1985); third, a reliable and valid measure of outputs against the specification must be available, and the controlling party must have the ability to carry out the measurement and evaluation (Ouchi 1977).

4.1.5 The choice of control strategy

Along with informal modes of organisational control the ability of an organisation to effect and blend different modes of formal control depends upon their ability to manage the informational requirements required by these methods (Eisenhardt 1985; Kirsch 1997).

If an organisation is able to articulate and measure information concerning the programmability of a task then behavioural control can be used. Similarly if an organisation as able to articulate and measure information concerning the specification of the outcome of a task then outcome control can be used. Outcome control can be specified without any knowledge of how the task is to be performed. However for behavioural control to be managed there is some requirement for the outcome to known as this mode hinges on knowledge of the means used to produce a type of output. Furthermore, when the appropriate information is can be measured and articulated, both modes of formal control can be applied. When there is insufficient knowledge for these formal modes to be applied, Ouchi (1977) suggest the use of informal modes of control as an alternative.

4.1.6 The use of organisational control theory in Information Systems

Ouchi's Theory of control has been well received in the IS community, and is typically used in studies of control of information systems development. Of the more cited research, Kirsch (1997; 2004) is first significant proponent of the theory of control for this use. For example her use of the theory to examine the modes of control in IS project management (1997) revealed that in practice managers use a portfolio of formal and informal means of control. Her research into the dynamics of control in the deployment of global information systems revealed the mix and emphasis on different modes of control changed as the phases of such projects progressed. Choudhury et al (2003) examined portfolios of control in the specific IS context of outsourced software development projects, finding that outcome controls dominated at the start of projects, and that behavioural and informal modes became increasingly important as the project progressed. Harris et al (2009) examined how the development of flexible software was controlled under conditions of uncertainty, and developed the notion of emergent outcome control, as the goal of the project became clearer. Tiwana et al (2009) compared the types of control used in internal and outsourced software projects, determining that formal modes were used to a greater degree in outsource projects and informal modes more in internal projects. Finally Nidumolu et al (2003) attempt to extend Ouchi's theory with structural approaches to control, drawing upon concepts of standardisation of methods and decentralisation of authority.

4.1.7 Suitability of theory for studying processes of control of platform innovation

Whilst Ouchi's theories of formal control have proved useful for the study of control within IS development projects, the question remains as to whether they are suitable for studying processes of control in digital platform innovation. There are, after all, considerable differences between the innovation within large scale IS development projects and the development of platform complements for digital platforms. The most apparent differences are with regards to the scale and scope of development. The IS development projects that are examined in the IS literature tend to be concerned with small numbers of large, relatively slow moving projects where individual developers come from a small number of suppliers. As was outlined earlier, apps development on digital platforms concern large numbers of small, quick moving developments implemented by large numbers of independent developers. Furthermore, the objective of enabling third parties to develop complements on digital platforms is to effectively outsource innovation of small services that add value to a platform. Consequently developers of complements, such as apps, have a large degree of freedom with regards to what they develop in contrast to most IS developments which are generally well specified.

Such is the speed, scale and scope of the apps developments on digital platforms, that the information processing requirements for the formal control of these developments is significant. Without the application of highly rationalised and efficient methods of formal control, platform owners are likely to face their own crises of control (Beniger 1986). Ouchi's theory is well suited to studying formal control in these environments. Ouchi's theorisation is based on rational, efficiency oriented, information based conceptualisation of formal control (Eisenhardt 1985, P137). In terms of presenting a rationalised view, the theory of organisational control presents a mechanistic concept of formal control process (Kirsch 1997). The process of formal control appears to be simple: monitoring performances; comparing them with a standard; providing selective rewards and adjustments (Ouchi 1977). It provides a perspective of control as a standardising procedure, ensuring uniformity of approach to enable coordination and goals to be met (Nidumolu and Subramani 2003). In this way it requires the clear programmability of tasks and the specification of outcomes (Eisenhardt 1985). Ouchi's view of formal control is that it is information based (Eisenhardt 1985). Information is obtained from measuring behaviour or outputs (Ouchi 1977). It is then evaluated and used in decisions as to whether to apply control or not. Control requires the communication of information as mechanisms to inform controlees' performances. The process of formal control is therefore reliant on information systems (Eisenhardt 1985). Control is finally affected when the controlee adjusts his behaviour, motivated by selective rewards and sanctions, to act in accordance with these communicated mechanisms (Kirsch 1997).

4.2 Beniger's model of control

A complementary theory of control is provided in Beniger's (1986) historical treatment of how organisations over the past two centuries applied rational, bureaucratic and technical means of control in order to overcome crises in control develops a formal model of control which is also dependent on information. In his model organisational control is needed in order to coordinate and direct increasingly complex activities to restore order and to make them purposive, effective and efficient. Beniger develops a model of control, whereby effective control is dependent on information processing and on communication between controllers and controlees. Beniger (1986, P434) claims that information processing is necessary for purposeful goal directed activity, involving constant comparison of current states to future goals. This comparison is necessary in order to make sure that the activity being undertaken leads to progression towards achieving a goal. Beniger (1986, P434) also claims that two way communication between controllers and controlees is necessary in order that the former can communicate influence to the latter, and the latter provide feedback to the former. Thus an element of feedback is essential in control in order that control itself can be regulated.

Whilst Beniger does not clearly distinguish between behavioural and outcome control, his model is of interest as it provides finer granularity of the stages of control. Furthermore, whilst his book is largely a history of control within organisations and society at large using pre digital technologies of control, the vocabulary of control that he uses is consistent with technologies of information processing. For example, he refers to the pre-processing, or rationalisation, of information, in terms of its creation, storage, and coding (Beniger 1986, P395). He alludes to the functions whereby information is processed for determining decisions as being driven by programming. Finally he implies that control is communicated to and feedback obtained from the internal environment of an organisation and from the environment outside of it.

4.2.1 Technologies of control

At the heart of Beniger's thesis on control lie technologies for the communication of information and the technologies of information processing. Beniger's treatment of technologies of control divides broadly into practices and artefacts, which he considers have enabled and propelled each other over time.

Beniger identifies bureaucracy and rationalisation as technologies of practice and claims that as they emerged and became increasingly sophisticated, so did material technologies to enable and facilitate them. These artefacts facilitated the activities of bureaucratic control, namely information processing and communication, and mirrored their functions. His analysis of artefacts used as control technologies is initially historical. Information was generated on typewriters and was input into processing functions via stock ticker, press clipping services, opinion surveys, censuses and indices. Information was more effectively recorded and stored through the systematisation of record keeping and the advent of technologies such as dictating machines. The processing of information was increasingly automated through innovations such as the keyboard calculator and punch card systems, and communication was facilitated with the advent of the telegraph network, the telephone, and mass communications. Finally, Beniger documents how emergent bureaucratic practices, such as modern accounting techniques, professional management, statistical quality control and "scientific management" were programmed into managers in new institutions such as business schools.

However his analysis of artefacts as control technologies becomes more contemporary after the burgeoning volume and velocity of economic activity in the latter half of the 19th century. With this increase in activity, Beniger identifies the emergence of the "Information Society". The use of control technologies enabled a dramatic increase in the amount of information generated and processed, and an accompanying dependence on information for control. The increasing volume and importance of information meant that information processing and flows, needed to be controlled in their own right, leading to new control crises (Beniger P291). This in turn led to the introduction of new control mechanisms for the control of information in its own right, which was increasingly removed from the processing of matter and energy, resulting in layer upon layer of control. The control of information, which due to its immaterial nature, poses a significantly greater problem on account of the scale and speed of its production. However, as in previous control crises, innovations have enabled these challenges to be met. The control revolution has advanced in the past fifty years, to enable the bureaucratic control over vast quantities of digitalised information through the emergence of digital technology. As Beniger states, digitalised technology allows the creation, storage, coding, processing and communication of information on a scale and speed unimaginable at the start of the control revolution.

4.3 Summary: a model of formal organisational control

This chapter introduced theories of formal organisational control as a means to explain how platform owners control the innovation of digital platform complements by third parties. The following summary serves to restate a generalised theory of formal organisational control, based on the theories of Ouchi (1975; 1977; 1978) and Beniger (1986). It is used in this research to analyse empirical data concerning instances of contested innovation, in order to explain how platform owners attempt to control innovation. This generalised model is illustrated in figure 4.1.

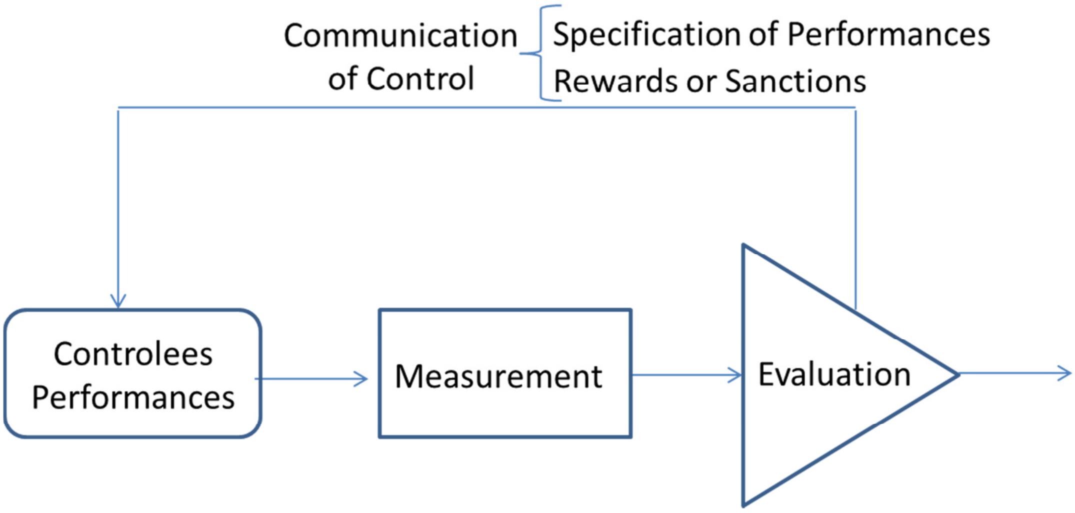


Figure 4.1 Generalised model of the process of formal control

(After Ouchi (1978))

It shows how the development of platform complements by third parties (or controlees) is controlled in terms outcomes and behavioural performances. It shows how they are measured and evaluated against specifications, after which the platform owner (the controller) issues a reward or sanction as appropriate. Control is communicated by controllers to controlees in the form of specifications, against which performances and measured and evaluated, and in terms of decisions, in terms of sanctions and rewards.

Specifications of control will be examined in order to identify both what innovation outcomes and behaviours are controlled and how they are controlled. The means by which platform owners measure and evaluate performances against specifications will be examined if sufficient data exists. The types of rewards and sanctions will be investigated and analysed against particular specifications. Furthermore, the dynamics of control, or changes over time in these components will be sought.

5 Results and Analysis

The objective of this chapter is to present and analyse the empirical data in order to describe and explain the dynamics of the control of innovation of platform complements on Apple's iOS platform. In doing so it addresses the research question namely, what is the nature of control in digital platform innovation? It does so by answering each of the two sub questions in turn: 1) how do the dynamics of the control of innovation change over time?; and 2) How is control of innovation asserted and lost on digital platforms? To that end, the chapter is broadly divided into two parts, description and explanation.

The first part concerns the description of the dynamics of the control of innovation in terms of sequences of action that occur when innovation is contested. Sub-sections 1.1 to 1.3 focus on addressing this description. 45 stories concerning instances of contested innovation that were identified within the empirical data extracted from blogs are first presented. A coding scheme, based on semiotic squares, is developed. This is used to code the stories into a set of 45 sequences of actions. Narrative networks (Pentland and Feldman 2007) are then constructed on the basis of this set of sequences of action. Finally, the narrative networks are combined and abstracted, in order to enable a generalised description of the dynamics of the control of innovation across the 45 observed instances of contested innovation.

The second part concerns an explanation of the dynamics of the control of innovation in terms of two opposing sets of mechanisms. The first is Apple's mechanism for control in the Apps Approval Process. The second concerns a set of mechanisms that developers use in responding to Apple's attempts at controlling innovation. Sub-sections 1.4 to 1.6 focus on addressing this explanation. Apple's Apps Approval process is first analysed as a mechanism for control using formal organisational control (Ouchi 1977; Ouchi 1978; Ouchi 1979) as a theoretical lens. The developers’ response to Apple's control is then analysed and explained as a set of mechanisms. Finally, the relationship between Apple's mechanism for control and the developers' mechanisms for responding to control, as well as the outcome of these instances of contested innovation, are described and explained on the basis of what is observed in the empirical data. Throughout the second part of the chapter insights are generated into the nature of the control of innovation and explanations are sought.

The chapter concludes with a summary of findings.

PART I: DESCRIBING THE DYNAMICS OF THE CONTROL OF PLATFORM INNOVATION

5.1 Presenting the stories

The corpus of data was processed in order to identify narrative themes and associated stories. Each story details innovating actions by an apps developer and controlling actions by Apple, as the apps developer attempts to get their innovation onto Apples iOS platform. As such this represents the implementation of stage four of the method outlined previously. The objective of identifying themed stories within the data is to facilitate steps four, five and six of my method. These final stages code the sequences of innovating and controlling actions within stories, so that narrative networks can be built and analysed.

The interactions detailed in the themed stories are concerned with tussles between Apple and its developers over what platform complements are permitted as innovations on the iOS platform. These stories contain a narrative where at some point Apple blocks an innovation, the developer contests Apple's decision, and some resolution is eventually arrived at. The stories do not contain a narrative where Apple simply accepts an innovation, or where a developer simply accepts Apple's decision to blocks an innovation, as both these story types are free from tussles.

45 complete stories were identified, which detailed individual instances of contested innovation on the iOS platform coming to some form of resolution. The 45 complete stories are detailed in table 5.1. Each narrative is described in the table in terms of: 1) the title of the innovation that was contested; 2) the developer of the innovation; an indication of the period over which the innovation was contested; 3) a brief description of the tussle; and 4) the narrative fragments concerning innovating and controlling actions of the developer and Apple which are listed sequentially. The title of the story is named after the innovation in question; furthermore it is enumerated to allow for easy referencing in this thesis. Last, the appropriate blog entry detailing each narrative fragment is referenced, and can be referenced in appendix A.

Case	Interaction Between Developers and Platform Owners	
1. 3G Skype **Skype vs. (Apple)** **From: 07/04/2009** **To: 07/10/2009**	Apple and Skype tussle over whether to enable voice over IP calls on 3G, rather than just Wi-Fi, using Skype's iPhone app	
	Action#1	Skype wish to launch a full blown version of their app on the iPhone including 3G access [5.1]
	Action#2	Apple only allows a cut back Wi-Fi only variant on its App Store, due to AT&T demands to prevent the 3G variant. [5.1]
	Action#3	Consumer groups lobby the FCC to force VoIP apps like Skype to be allowed on AT&T's 3G [5.2]
	Action#4	Apple and AT&T eventually give in to allow VoIP apps like Skype over 3G [5.3]
2. Admob **Google vs. (Apple)** **From: 31/03/2010** **To: 09/09/2010**	Apple and Google tussle following the introduction of Apple's new developer terms. Under these new terms app developers are prohibited from using Google's AdMob platform for in app advertising.	
	Action#1	Apple initially allows the AdMob platform [5.4]
	Action#2	Apple updates the terms of its Developer Program License Agreement, blocking the use of the AdMob platform [5.4]
	Action#3	Following complaints, the FTC investigates the market for mobile advertising [5.5]
	Action#4	Apple updates its terms, making clear that alternative advertising platforms are allowed. [5.6]
3. Adobe Flash **Adobe vs. (Apple)** **From: 17/06/2010** **To: 09/11/2011**	Concerns the tussle between Apple and Adobe concerning Apple's ban on the use of Adobe Flash on the iPhone.	
	Action#1	Adobe has frequently expressed its interest in enabling Flash on the iPhone. http://blog.admob.com/2010/06/09/mobile-advertising-and-the-iphone [5.8]
	Action#2	Apple continues to disallow Flash citing security and battery consumption issues [5.9]
	Action#3	Adobe contends Apple's claims, and places public advert stating how it wishes to work with Apple [5.10]
	Action#4	Applications are released (e.g. Skyfire), which enable the viewing of Flash content on the iPhone, without Flash actually being present on the handset [5.11]
	Action#5	Adobe announces that it will cease with development of its mobile flash browser plugin and focus on developing HTML 5 capabilities [5.12]
4. Adobe Developer Tools **Adobe vs. (Apple)** **From: 05/10/2009** **To: 09/09/2010**	Concerns the impact on Adobe of the policy restriction that Apple placed on the choice of developer tools used by third parties to create iPhone applications.	
	Action#1	Adobe create tools to enable designers to build Flash apps to run on the iPhone [5.13]
	Action#2	Apple originally allows the use of Adobe tools for the development of iPhone apps. [5.13]
	Action#3	Apple later introduces policy to restrict the use of the Adobe tools [5.14]
	Action#4	Adobe tries to influence Apples decision through adverts [5.15]
	Action#5	Apple eventually relents under pressure from developers [5.16]
5. Ari David **Ari David vs. (Apple)** **From: 22/05/3020** To: 26/05/2010	Concerns Apple's rejection of politician Ari David's election campaign app	
	Action#1	Republican Candidate Ari David has iPhone App submitted to cover his election campaign [5.17]
	Action#2	App rejected on the grounds that it is defamatory of his opponent, incumbent Henry Waxman. [5.17]
	Action#3	The decision is widely reported in the media [5.18]
	Action#4	Apple changes its mind and allows the app [5.18]
6. Baby Shaker **Sikalosoft vs. (Apple)** From: 22/04/2009 To: 23/04/2009	Apple rejects and then removes an app, developed by Sikalosoft, deemed "poor taste" as it "simulates" violence to minors.	
	Action#1	Developer submits app which enables a caricature baby to be silenced when handset is shaken [5.19]
	Action#2	The app is initially allowed on to the App Store by Apple [5.19]
	Action#3	Following public outrage, Apple remove the app from the App Store [5.20]

Table 5.1 Themed stories identified within the corpus of data

Case	Interaction Between Developers and Platform Owners	
7. Big Brother Security **Big Brother Security vs. (Apple)** **From: 15/06/2011** **To: 15/06/2011**	Concerns the tussle between Apple and Big Brother Security over an app which takes photos of intruders trying to gain access to an iPhone	
	Action#1	Developer submits app which takes photos of "intruders" entering incorrect passwords on iPhones [5.21]
	Action#2	Apple initially admit the app onto the App Store [5.21]
	Action#3	When Apple find out that the app is collecting commonly used passwords the app is banned [5.21]
	Action#4	Developer removes the offending functionality and resubmits [5.21]
	Action#5	The decision is widely reported [5.21]
	Action#6	Apple ignore him and the app is not readmitted [5.21]
8. Bobble Rep **Tom Richmond vs. (Apple)** **From: 09/11/2009** **To: 14/11/2009**	Apple and Tom Richmond tussle over an app which contains caricatures of US politicians	
	Action#1	Tom Richmond submits app enabling US voters to find contact details of their senators and member of congress [5.22]
	Action#2	Apple reject app on account of the cartoon caricatures of members being "obscene" and "objectionable" [5.22]
	Action#3	Richmond publicizes the rejection and accompanying rational [5.23]
	Action#4	Apple reverse their decision and admit the app onto the App Store [5.23]
9. Box Office **Metasyntactic vs. (Apple)** **From:04/08/2008** **To: 14/08/2008**	Concerns the tussle between Apple and Metasyntactic concerning Apple's removal of their movie listings app from the App Store	
	Action#1	Metasyntactic submits movie listings app Box Office [5.24]
	Action#2	Apple approves the app and it is put onto the App Store [5.24]
	Action#3	Apple remove the app for no apparent reason [5.24]
	Action#4	The developer publicizes this decision in the media [5.25]
	Action#5	Apple reverses its decision and allows the app back into the App Store [5.26]
10. C64 Emulator **Manomio vs. (Apple)** **From: 08/09/2009** **To: 18/09/2010**	Concerns the tussle between Apple and Manomio concerning their Commodore 64 app, which contains a BASIC emulator, enabling users to “program” their iPhones	
	Action#1	Manomio submits their Commodore 64 emulator app to Apple [5.27]
	Action#2	Apple reject the app because it contains a BASIC emulator [5.27]
	Action#3	This news is widely reported [5.27]
	Action#4	The developers apparently remove the BASIC emulator and resubmit the application [5.27]
	Action#5	Apple then admit the App on to the App Store [5.27]
	Action#6	Apple then remove the app when they find out that the emulator can be accessed by shortcut keys, this functionality being hidden within the app as an “Easter egg” [5.27]
	Action#7	Apple relax their policy with regards to interpreters and the app is allowed again [5.28]
11. CastCatcher **Amro Mousa vs. (Apple)** **From: 26/08/2009** **To: 26/08/2009**	Concerns the tussle between Apple and Amro Mousa concerning his radio streaming app, which is removed after Apple become concerned that it uses too much bandwidth	
	Action#1	Developer Amro Mousa submits CastCatcher, an internet radio streaming app [5.29]
	Action#2	Apple admit the app onto the App Store [5.29]
	Action#3	Apple later withdraws the app claiming that it transferred "excessive volumes of data over the cellular network." [5.29]
	Action#4	Developer attempts to influence Apple in online media [5.29]
	Action#5	The developer reconfigures the app [5.29]
	Action#6	Apple then readmit the app onto the App Store [5.29]
12. Convertbot **Convertbot vs. (Apple)** **From: 28/08/2009** **To: 28/08/2009**	Apple and Convertbot tussle over the UI of an app which converts measurements between different scales.	
	Action#1	Developer submits Convertbot [5.30]
	Action#2	The app is not allowed onto the App Store because an icon is too similar "to one of Apple’s default resource icons" [5.30]
	Action#3	Developer attempts to influence Apple in online media [5.30]
	Action#4	Developer resubmits new corrected version [5.30]
	Action#5	New corrected version is accepted [5.30]

Table 5.1. Themed stories identified within the corpus of data (continued)

Case	Interaction Between Developers and Platform Owners	
13. EFF Updates App **Exact Magic vs. (Apple)** **From: 02/06/2009** **To: 02/06/2009**	Apple and Exact Magic tussle over whether an iPhone app which enables an RSS feed from the Electronic Frontier Foundation (EFF) should be allowed onto the App Store	
	Action#1	Exact Magic submits app to allow Electronic Frontier Foundation RSS feed [5.31]
	Action#2	Apple blocks claiming "iTune's App Store's policy against "objectionable" content" [5.31]
	Action#3	EFF publicize the case on their blog [5.31]
	Action#4	Apple reverse their decision and admit the app on to their App Store [5.31]
14. Eucalyptus **James Montgomerie vs. (Apple)** From: 21/05/2009 To: 24/05/2009	Apple and James Montgomerie tussle regarding his eBook reader app which accesses free open source eBooks from Project Gutenberg	
	Action#1	Developer submits Project Gutenberg eBook reader [5.32]
	Action#2	Apple rejects on basis that it can access "objectionable" content - i.e. Kama Sutra [5.32]
	Action#3	The news of the rejection is spread on the blogosphere [5.33]
	Action#4	Apple changes its mind and allows the application [5.34]
15. EyeTV **Elgato vs. (Apple)** **From: 16/10/2009** **To: 19/10/2009**	Concerns the tussle between Apple and Elgato Systems concerning whether their app which streams live TV to an iPhone or an iPad, should be allowed on to the App Store	
	Action#1	Developer submits 3G based TV streaming app [5.35]
	Action#2	App is initially allowed onto App Store [5.35]
	Action#3	Apple then pull app because it use of 3G is something that AT&T has objected to in the past [5.36]
	Action#4	The removal of the app brings widespread media coverage [5.36]
	Action#5	Apple then restore the app on the App Store [5.37]
16. Financial Times **Financial Times vs. (Apple)** **From: 14/07/2009** **To: 31/08/2011**	Apple and the Financial Times tussle over how revenues from an iPhone/iPad version of the newspaper should be shared.	
	Action#1	The Financial Times submits an iPhone app version of its newspaper [5.38]
	Action#2	The app is accepted by Apple onto the App Store [5.38]
	Action#3	The FT publicly dispute Apple's cut of the FT's iPhone app revenues [5.39]
	Action#4	The FT launch an HTML 5 based version of its newspaper for the iPhone and iPad [5.39]
	Action#5	The FT then withdraw its application from the App Store [5.40]
17. Google Books **Google vs. (Apple)** **From: 06/02/2009** **To: 26/06/2011**	Concerns the tussle between Apple and Google over in app subscriptions for the Google Books app on the iPhone	
	Action#1	Google submit their eBook reader Google Books [5.41]
	Action#2	Apple accepts the app into their App Store [5.41]
	Action#3	Apple reject Google Books because it does not conform to their new rules on in app purchases [5.42]
	Action#4	The news of the rejection receives media coverage [5.42]
	Action#5	Google resubmit their eBook reader Google Books conforming to the new rules [5.43]
	Action#6	Apple readmit Google Books back into the App Store [5.43]
18. Google Voice **Google vs. (Apple)** From: 27/07/2009 To: 16/11/2010	Concerns the tussle between Google and Apple over the availability of Google Voice as a native application on the Apple App Store	
	Action#1	Google submit Google Voice app to Apple. [5.44]
	Action#2	Apple initially accepts Google Voice, and derived applications, onto the App Store [5.44]
	Action#3	Apple bans Google Voice, and derived apps, claiming an overlap with iPhone functionality [5.44]
	Action#4	Google appeals to Federal Communications Commission, who investigate, but Apple retain the ban [5.45]
	Action#5	Google launches a Web App version of Google Voice circumventing Apple's control [5.46]
	Action#6	Apple eventually allows Google Voice in the iPhone [5.47]

Table 5.1. Themed stories identified within the corpus of data (continued)

Case	Interaction Between Developers and Platform Owners	
19. GV Mobile **SK Dev Solutions** **vs. (Apple)** **From: 27/07/2009** **To: 19/09/2010**	Concerns the tussle between Apple and SK Dev Solutions concerning their app which enables Google Voice on the iPhone, independent of Google's own app.	
	Action#1	SK Dev solutions submitted their Google Voice based app to Apple [5.48]
	Action#2	The application is accepted by Apple onto the App Store [5.48]
	Action#3	Apple then block application, along with other GV variants [5.48]
	Action#4	This news receives media coverage [5.48]
	Action#5	Developer releases GV Mobile as an HTML5 web app [5.49]
	Action#6	Apple changes its mind and allows GV Mobile, along with other Google Voice variants, back onto the App Store [5.50]
20. Hottest Girls **Allen Leung vs. (Apple)** **From: 25/06/2009** **To: 25/06/2009**	Apple and developer Allen Leung tussle over whether his soft porn app should be allowed onto the App Store	
	Action#1	Hottest Girls soft pornography application submitted [5.51]
	Action#2	The application is accepted by Apple [5.51]
	Action#3	Apple then withdraw the application shortly after as it was deemed inappropriate content according to the iPhone Developer Program [5.52]
	Action#4	The news receives coverage in the media [5.52]
21. I Am Rich **Armin Heinrich vs. (Apple)** **From: 06/08/2008** **To: 06/08/2008**	Apple and developer Armin Heinrich tussle over whether a functionless app costing $1000 should be on the App Store	
	Action#1	I Am Rich application - functionless app costing $1000 submitted [5.53]
	Action#2	I Am Rich application accepted onto the App Store [5.53]
	Action#3	Apple rapidly pull I Am Rich, but not before the application sells 6 times [5.54]
	Action#4	News of this application being pulled receives coverage in the media [5.54]
22. Jailbreakme **Comex vs. (Apple)** **From: 02/08/2010** **To: 07/07/2011**	Concerns the tussle between C and Apple over the right to "hack" the iPhone, taking the control away from Apple concerning the software that runs on the device, giving it to consumers**.**	
	Action#1	Developer Comex releases iPhone hack based on code hidden in an Adobe PDF file to simply jailbreak an iPhone with iOS 4 [5.55]
	Action#2	Apple quickly release an OS patch (iOS 4.0.2) to block this hack [5.56]
	Action#3	Later the developer release a new hack to jailbreak iOS, based on a similar method [5.57]
	Action#4	Apple promptly react to this by releasing a new OS patch [5.57]
23. Mark Fiore **Mark Fiore vs. (Apple)** **From: 15/04/2010** **To: 16/04/2010**	Apple and Mark Fiore tussle over his iPhone app, featuring his satirical cartoons of public figures, being allowed on the App Store	
	Action#1	Satirical cartoonist Mark Fiore submits iPhone app in December 2009 [5.58]
	Action#2	The app is rejected because it "ridicules public figures," a violation of the iPhone Developer Program License Agreement [5.58]
	Action#3	When Fiore is awarded Pulitzer prize for journalism, the news of the app rejection is publicized widely [5.59]
	Action#4	Apple relent and allow the app onto the App Store [5.60]
24. Me So Holy **Lil' Shark vs. (Apple)** **From: 11/05/2009** **To: 11/05/2009**	Apple and Lil' Shark tussle over whether to allow the Me So Holy app, which enables people to superimpose their faces onto religious figures, onto the App Store	
	Action#1	Developer submits app which enables people to superimpose their faces onto religious figures [5.61]
	Action#2	Apple admits app onto the App Store [5.62]
	Action#3	Apple then removes app on the grounds that it is contains objectionable content and is in violation of Section 3.3.12 from the iPhone SDK Agreement [5.62]
	Action#4	This news receives media coverage [5.62]
	Action#5	Developer then relaunches app on alternative App Store Cydia for jail broken iPhones [5.62]

Table 5.1. Themed stories identified within the corpus of data (continued)

Case	Interaction Between Developers and Platform Owners	
25. Nescaline **Jonathan Zdziarski vs. (Apple)** **From: 22/12/2009** **To: 22/12/2009**	Apple tussles with Jonathan Zdziarski concerning whether to allow his Nintendo emulator app, Nescaline, on the iPhone.	
	Action#1	Developer submits Nintendo emulator for iPhone [5.63]
	Action#2	App gets approved - which is unusual as emulators generally get rejected at this time [5.63]
	Action#3	Apple withdraw app from App Store apparently on account of it being an emulator. [5.63]
	Action#4	This news receives media coverage [5.63]
	Action#5	Alternative Nintendo Emulators become available on alternative App Store Cydia for jail broken iPhones [5.63]
26. Netshare **Nullriver vs. (Apple)** **From: 31/07/2008** **To: 01/08/2008**	Concerns Nullriver's tussle with Apple over the whether their iPhone tethering app should be allowed onto the App Store	
	Action#1	Nullriver submit Netshare an iPhone tethering application [5.64]
	Action#2	Apple allow the application onto the App Store [5.64]
	Action#3	Apple then pull the application from the App Store on the same day [5.64]
	Action#4	This is widely reported in the media [5.64]
	Action#5	Apple reallow the application onto the App Store the next day [5.65]
	Action#6	Apple then pull the application from the App Store that same day [5.65]
27. NinjaWords **Matchstick Software vs. (Apple)** **From: 04/08/2009** **To: 06/08/2009**	Apple and Matchstick Software tussle over an iPhone dictionary app, sourcing words from open source Wiktionary.org	
	Action#1	Developer submits iPhone dictionary app, sourcing words from open source Wiktionary.org [5.66]
	Action#2	Apple rejects app on account of offensive "urban slang" terms that are not found in popular dictionaries [5.66]
	Action#3	News of this rejection receives media coverage [5.66]
	Action#4	Developer decides to filter offensive content and resubmit app [5.67]
	Action#5	In the meantime, Apple introduce rule allowing adult rated (17+) content and allow the app [5.67]
28. Opera **Opera Software ASA vs. (Apple)** **From: 31/10/2008** **To: 12/04/2010**	Concerns Opera Software's tussle with Apple over the whether their internet browser app for the iPhone should be admitted onto the App Store.	
	Action#1	Developers at Opera submit their browser app for iPhone [5.68]
	Action#2	Apple rejects it as it competes with core Apple functionality, namely the Safari browser [5.68]
	Action#3	Opera start a long campaign of influence [5.68]
	Action#4	As Apple relaxes its policies, Opera rebuild their browser for the iPhone and submit [5.69]
	Action#5	Apple approves the browser [5.69]
29. Podcaster **Alex Sokirynsky vs. (Apple)** **From: 4/09/2008** **To: 27/01/2009**	**Concerns the strategies used by a developer in order to have his podcast download application listed on the App Store.**	
	Action#1	Developers submit Podcaster an application for downloading Podcasts [5.70]
	Action#2	Apple rejects Podcaster App claiming that it duplicates Apple's own functionality [5.70]
	Action#3	This news receives media coverage [5.70]
	Action#4	Podcaster Developer Uses Little-Known "Ad Hoc" Mode to distribute banned iPhone app [5.71]
	Action#5	The developer tweaks the functionality of his application and resubmits to the App Store [5.72]
	Action#6	Apple then lists the adjusted application. [5.72]
30. Pull My Finger **Eric Ogren vs. (Apple)** **From: 04/09/2008** **To: 12/12/2008**	Apple and Eric Ogren tussle over whether his "Pull My Finger" novelty app should be allowed onto the App Store	
	Action#1	Developers submit "Pull My Finger" [5.73]
	Action#2	Apple rejects the application on the basis of being limited utility [5.73]
	Action#3	Developer publicizes rejection, claiming there are many applications of limited utility that have been accepted [5.74]
	Action#4	Apple finally relents and admits the application [5.74]

Table 5.1. Themed stories identified within the corpus of data (continued)

Case	Interaction Between Developers and Platform Owners	
31. Pulse News Reader **Alphonso Labs Inc vs. (Apple)** **From: 08/06/2010** **To: 09/06/2010**	Concerns the tussle between Apple and Alphonso Labs Inc over admitting their news reader app onto the App Store	
	Action#1	Developers send Pulse News Reader, a news-reading iPad app, in for review [5.75]
	Action#2	Apple accept the app, Steve Jobs praises it in the 2010 Worldwide Developers Conference [5.75]
	Action#3	The app is then pulled from the App Store, after the New York Times claims it infringes their rights [5.75]
	Action#4	The pulling of the app receives media focus [5.76]
	Action#5	The developers adjust the application so that New York Times stories are omitted [5.76]
	Action#6	Apple reinstate the app [5.76]
32. Readability **Readability vs. (Apple)** **From: 21/02/2011** **To: 10/03/2011**	Apple and Readability tussle over whether to allow an eBook reader app onto the App Store	
	Action#1	Readability submit their eBook reader, which passes revenues on to original publishers and authors [5.77]
	Action#2	Rejection of readability gets widespread media coverage [5.77]
	Action#3	Apple reject Readability because it does not conform to their new rules on in app purchases [5.77]
	Action#4	Readability launch HTML 5 version of their app [5.78]
33. Routesy **Steven Peterson vs. (Apple)** **From: 27/06/2009** **To: 27/06/2009**	Concerns the tussle between Apple and developer Steven Peterson over his iPhone app which provides timetable information for San Francisco public transport	
	Action#1	Developer submits Routesy, a GPS driven iPhone app which provides bus timetables in San Francisco [5.79]
	Action#2	Apple accept the application into the App Store [5.79]
	Action#3	After complaints from bus operator NextBus, that it owned timetabling information, and Routesy had no rights to use it, Apple pulls the application [5.79]
	Action#4	Routsey appeal to the San Francisco Municipal Transportation Agency who actually run the buses and trains in SF, as well as to Apple [5.79]
	Action#5	The application is reinstated into the App Store [5.79]
34. Sekai Camera **Tonchidot vs. (Apple)** **From: 04/03/2010** **To: 04/03/2010**	Concerns the tussle between Apple and Tonchidot concerning their augmented reality app and whether it should be allowed into the App Store	
	Action#1	Develop Tonchidot submits Sekai Camera, an augmented reality app that uses GPS and WiFi for location finding [5.80]
	Action#2	Apple accept the App, and it wins acclamation within the industry [5.80]
	Action#3	Apple pull the app, claiming it accessed non-public iOS functionality [5.80]
	Action#4	News of the pulling of Sekai Camera receives coverage in the media [5.80]
	Action#5	Tonchidot re-engineer the app [5.80]
	Action#6	The app is allowed back onto the App Store [5.80]
35. Simply Beach **Simply Beach vs. (Apple)** **From: 23/02/2010** **To: 23/02/2010**	Concerns the tussle between Apple and Simply Beach over a beachwear retail app that banned from the App Store for containing indecent content	
	Action#1	Simply Beach, a shopping app from a beachwear retailer that sells bikinis, is submitted [5.81]
	Action#2	Apple accept the app into the App Store [5.81]
	Action#3	The app is blocked, as part of Apple's purge of pornographic content [5.81]
	Action#4	Simply Beach express their annoyance online [5.81]
	Action#5	Apple quietly reinstate the app without contacting Simply Beach [5.81]

Table 5.1. Themed stories identified within the corpus of data (continued)

Case	Interaction Between Developers and Platform Owners	
36. Someecards **Duncan Mitchell vs. (Apple)** **From: 07/10/2009** **To: 08/11/2009**	Concerns the tussle between Apple and Duncan Mitchell concerning his ecard iPhone App and its placement on the App Store	
	Action#1	Someecards develops and submits its ecard app for iPhone [5.82]
	Action#2	Apple accept the application onto the App Store [5.82]
	Action#3	Apple pull the application when they find satirical cards they find objectionable [5.82]
	Action#4	News of Apple's pulling this app reaches the media [5.82]
	Action#5	Someecards remove the offending ecards and resubmit [5.83]
	Action#6	Apple accept the application a second time [5.83]
	Action#7	Apple pull the application when they find satirical cards they find objectionable for a second time [5.83]
	Action#8	Someecards remove the offending ecards and resubmit [5.83]
	Action#9	Apple readmit the application onto the App Store [5.83]
37. Stanza **Lexcycle vs. (Apple)** **From: 02/02/2010** **To: 02/02/2010**	Concerns the tussle between Apple and Lexcycle concerning their ebook iPhone app	
	Action#1	Lexcycle submit their ebook application Stanza [5.84]
	Action#2	Apple admit Stanza onto the App Store [5.84]
	Action#3	Apple pulls Stanza, after an update which enables users to transfer books in the ePub or eReader format to their mobile devices using a USB cable. [5.84]
	Action#4	News of the pulling of Stanza is publicized in the media [5.84]
	Action#5	Lexcycle remove the offending functionality, which used a non-public API to enable sync via the USB cable [5.84]
	Action#6	Apple readmit Stanza onto the App Store [5.84]
38. Tawkon **Tawkon Ltd vs. (Apple)** **From: 17/05/2010** **To: 23/03/2011**	Apple and Tawkon Ltd tussle over the admission into the App Store of an iPhone app which measures levels of cellular radiation.	
	Action#1	Developers Tawkon submit an iPhone app which measures levels of cellular radiation [5.85]
	Action#2	Apple reject it, claiming that it uses undocumented APIs [5.85]
	Action#3	News of Apple's rejection of Tawkon reaches the media [5.85]
	Action#4	Tawkon release the application on alternative App Store, Cydia, for jailbroken iPhones [5.86]
39. TrapCall **Tel Tech Systems Inc vs. (Apple)** **From: 04/04/2011** **To: 04/04/2011**	Apple and Tel Tech Systems Inc tussle following their application to have their call blocking app admitted on to the App Store	
	Action#1	Tel Tech Systems Inc submit their Trapcall app which unmasks harassing blocked and private phone calls [5.87]
	Action#2	Apple ignore responding to the submission for 201 days [5.87]
	Action#3	News of the treatment of TrapCall is reported in the media [5.87]
	Action#4	Finally Apple admit the app into the App Store [5.87]
40. Trillian **Cerulean Studios vs. (Apple)** **From: 13/10/2009** **To: 19/11/2009**	Apple and Cerulean Studios tussle following their application to have their instant messaging app admitted into the App Store	
	Action#1	Cerulean Studios submit Trillian an instant messaging app for the iPhone [5.88]
	Action#2	Apple ignore responding to the submission for 3 months [5.88]
	Action#3	News of the treatment of Trillian is reported in the media [5.88]
	Action#4	Finally Apple admit the app into the App Store [5.89]
41. Tweetie **Loren Brichter vs. (Apple)** **From: 10/03/2009** **To: 10/03/2009**	Concerns the tussle between Apple and Lauren Brichter concerning whether an iPhone Twitter client should be allowed into the App Store	
	Action#1	Developer Loren Brichter submits iPhone Twitter client Twitter [5.90]
	Action#2	Tweeties is allowed onto the App Store by Apple [5.90]
	Action#3	Apple pull Tweetie as it allows access to Twitter feeds containing obscene language [5.90]
	Action#4	The developer publicises the decision over Twitter [5.91]
	Action#5	Apple relent and readmit Tweetie [5.91]

Table 5.1. Themed stories identified within the corpus of data (continued)

Case	Interaction Between Developers and Platform Owners	
42. Voice Central **Riverturn vs. (Apple)** **From: 29/07/2009** **To: 24/12/2009**	Apple and Riverturn tussle following the submission of their Google Voice based app Voice Central	
	Action#1	Developer Riverturn submit their Google Voice based app VoiceCentral [5.92]
	Action#2	Apple admits the app in to the App Store [5.92]
	Action#3	Apple pull VoiceCentral when they reject Google's own Google Voice App [5.92]
	Action#4	News of the pulling of VoiceCentral is reported in the media [5.92]
	Action#5	Riverturn relaunch the app on alternative app store Cydia, for jailbroken iPhones, thus circumventing Apple's App Store. [5.93]
43. Wallpaper Gallery **Start Mobile vs. (Apple)** **From: 21/08/2009** **To: 21/08/2009**	Apple and Start Mobile tussle over an app which contains artwork of President Obama	
	Action#1	Start Mobile Wallpaper Gallery submit an app concerning artwork of President Obama, including Shepard Fairey's famous "HOPE" image [5.94]
	Action#2	Apple claim the application is inappropriate stating that it "ridicules public figures" and do not allow it onto the App store [5.94]
	Action#3	This action generates much discussion in the media [5.94]
	Action#4	Apple acknowledge that the rejection was unjust and admit the app into the App Store [5.94]
44. Wallpaper Universe **FunMobility vs. (Apple)** **From: 24/11/2008** **To: 24/11/2008**	Apple and FunMobility tussle following the introduction of their wall paper app which features nudes	
	Action#1	Develop FunMobility submit Wallpaper Universe, a wall paper app featuring nudes [5.95]
	Action#2	Apple admit the application into the App Store (contrary to its rules, and by mistake) [5.95]
	Action#3	News of the pulling of Wallpaper Universe is reported in the media [5.95]
	Action#4	Apple remove the app from the App Store [5.95]
45. Wi-Fi Sync **Greg Hughes vs. (Apple)** **From: 14/05/2010** **To: 14/05/2010**	Apple and Greg Hughes tussle over whether his iPhone app, which enables cable free WiFi based synching for the device, should be allowed into the App Store	
	Action#1	Developer Greg Hughes submits Wi-Fi Sync enabling an iPhone to wirelessly sync with a computer without having to use a USB cable [5.96]
	Action#2	Apple reject the app as a result of security issues and "technical issues including reading and writing outside of the app's container." [5.96]
	Action#3	News of the rejection of Wi-Fi Sync reaches the media [5.96]
	Action#4	The developer launches the app on alternative App Store, Cydia, for jailbroken iPhones [5.96]

Table 5.1. Themed stories identified within the corpus of data (continued)

The positioning of these 45 stories within the window of time in which the blog entries were written is detailed in figure 5.1. Figure 5.1 details when Apple attempted to exert control on each of the attempted innovations detailed in the stories.

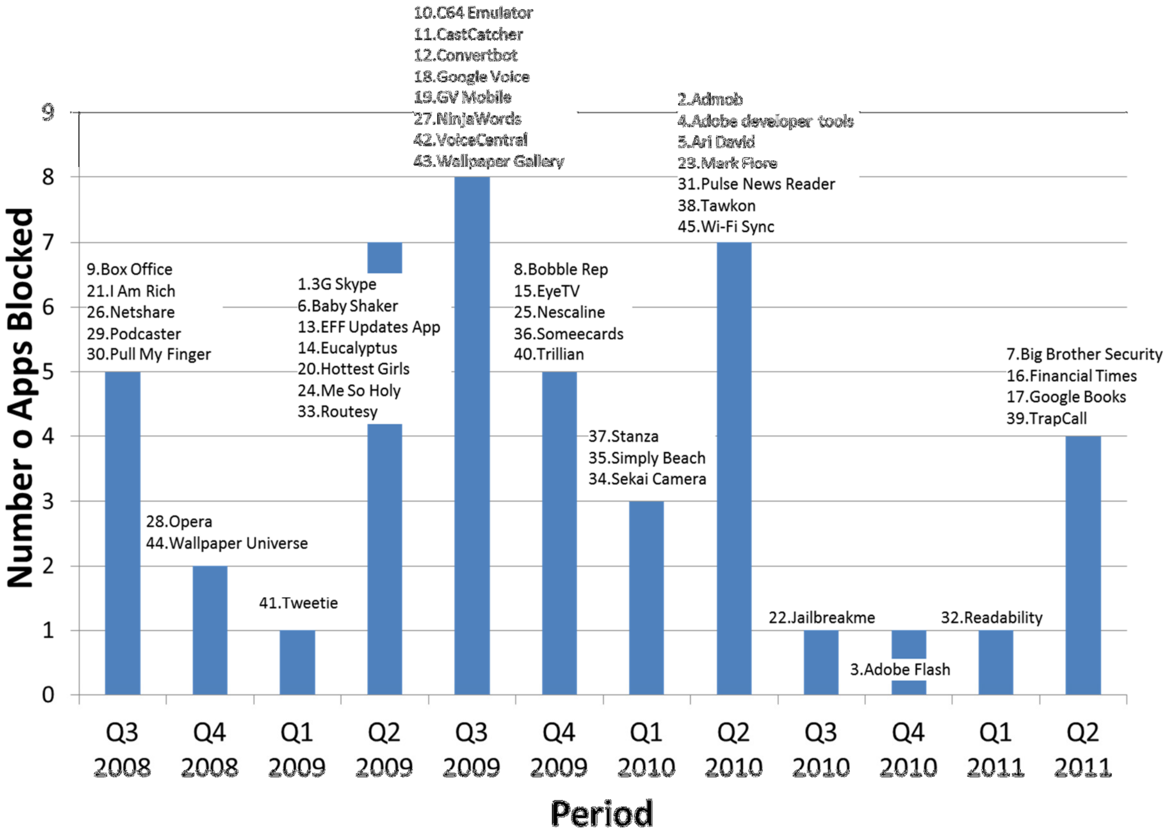


Figure 5.1 Distribution in time of when stories occur.

5.2 Coding and constructing narrative networks

Steps five and six of the approach adapted from Pentland and Feldman (2007) are used to construct narrative networks. Step five concerns the identification of a coding scheme and applying it to the themed narrative fragments. Step six involves the construction and analysis of narrative networks from the coded themed narrative fragments.

5.2.1 Coding the narrative fragments

Narrative fragments are coded for the purpose of sequence analysis in the form of narrative networks. The most essential element of each narrative fragment is the type of action that it contains. The treatment of the coding of this element makes up the bulk of this analytical stage. The results of this analysis are presented in two steps. First, a static semantic analysis, using semiotic squares, was conducted of the actions identified within the stories summarized in table 5.1. The results of this step are shown in order to identify common generic actions across the stories which can be used as codes. Second, a dynamic syntactic analysis was applied, and the results are presented in the form of coded sequences of narrative fragments, as they unfold in each themed story.

Identifying generic actions

The logic of semiotic squares, presented in chapter 3, is used in order to provide a cohesive but simplified range of generic actions that describe the innovating and controlling acts of the apps developers and Apple, as seen in the collected stories. These are illustrated in Figure 5.2, and are explained thereafter.

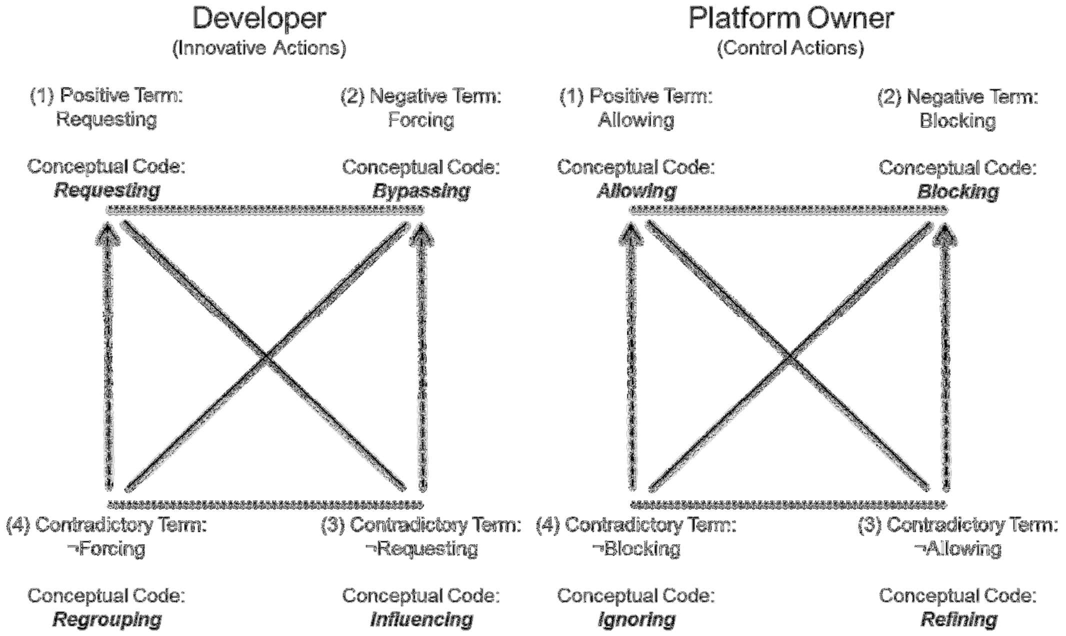


Figure 5.2 Semiotic squares identifying generic actions.

Starting with the developer, the basic oppositional terms, both positive and negative, were identified. A developer can broadly either attempt to "request" that an innovation is accepted, or, in opposition, attempt to "force" it on to a platform. These oppositional terms provide the basis of the logic that will help the identification of a basic set of coded actions for the developer. Since a compact lexicon of actions is required, the coding scheme is restricted to two additional logical terms.

First, the contradictory term of the positive term is identified as simply not "request". Second, the contradictory term of the negative term is identified as not "force". These logical terms are then used to identify "concepts" or actual realizations from within the data. A realization of the positive term of "request", in position 1, is simply the "requesting" of a new app, which is frequently seen in the data as the first action in the sequence. This “requesting” of a new app may or may not be within Apple’s boundary of permissible service innovation.

A realization of the negative term "force", in position 2, is simply the "bypassing" of Apple's platform by the developer. This is typically seen after Apple has blocked the developers’ attempts at gaining approval. This was the case when rival iPhone Application Store providers, in response to Apple's blocks, bypassed Apple's zone of control and opened up application stores distributing web apps, rather than native apps, for the iPhone. In this way the “bypassing” of Apple’s platform indicates the forcing of something outside of Apple’s boundary of permissible service innovation. Finding concepts to illustrate the contradictory terms in position 3 and 4, is more sophisticated than simply negating existing concepts. Concepts describing each contradictory term retain some basic characteristics of the original term, whilst losing others, and acquiring some similarities to the original term's binary opposite. In this way, developers are neither "requesting" nor "forcing" platform owners, but "influencing" them on the one hand, or simply "regrouping" and unilaterally abandoning or modifying their original plans on the other, without seeking to “bypass” the platform. “Influencing” is seen for example, in the case of Google appealing to the regulatory powers of the FCC to influence Apple to approve the Google Voice application on the iPhone App Store. Influencing may or may not be a direct appeal by the developer to the platform owner. Simply having users protest that a service innovation has been disallowed is an indirect act of a developer “influencing” a platform owner. "Regrouping" typically happens in the example of jailbreaking, where Apple identifies and blocks methods of jailbreaking, forcing developers to give up and try a new method or something completely different. “Regrouping” is interpreted as a backing down on the developers’ part to the challenge that they initially made to the platform owners boundary of permissible service innovation.

Applying the same process to the part of the platform owner, the fundamental oppositional terms "allow" an innovation or "block" an innovation are identified. The contradictory terms are simply "not allow" and "not block". A realization of the positive term of "allow", in position 1, is simply the "allowing" of a new app. The act of “allowing” a service innovation entails that the innovation falls within the platform owner’s boundary of permissible innovation, or, more interestingly, the platform owner has shifted the boundary of platform innovation in order to accommodate the new innovation. Similarly, a realization of the negative term of "block", in position 2, is simply the "blocking" of a new platform innovation. “Blocking” indicates that the attempted service innovation falls outside of the platform owner boundary of permissible innovation, and that the platform owner is therefore refusing to admit it on to the platform. Using the same logic as in the case of synthesizing concepts for the contradictory terms for developer actions, the same is done for the platform owner. A coded action is identified for the contradictory term in position 3 through the act of “refining”, which signifies the platform owner realizing that the issue is more complex than can be realized through a simple act of blocking or allowing. This can, for example, imply the reversal of previous decisions to either block or allow, and instead applying discretionary decisions to narrow the class of applications being blocked, or similarly expanding those allowed. In this case “refining” represents a targeted shift in the boundaries of permissible platform innovation. This is the case with Apple’s decision to refine the initial block of all developer environments other than its own, for later to realize that this meant excluding a large amount of games, which were constructed using alternative developer tools. Finally a coded action is identified for the last contradictory term in position 4 by synthesizing the act of the platform owner "ignoring" the requests of the developer. This is seen in the case of the traditional publishers’ failing to respond to Apple’s generative act of Apple distributing eBooks. By “ignoring” an attempted platform innovation the platform owner effectively and passively blocks a potential challenge to the boundary of permissible innovation, without outwardly committing itself one way or the other.

Following the application of the semiotic square to identify generic actions within the data a compact lexicon of eight coded actions emerges. On the developers' side the coded actions of requesting, bypassing, influencing and regrouping are identified. On the platform owner's side the coded actions of allowing, blocking, refining and ignoring emerge.

Identifying sequences of generic actions

Following the identification of a set of generic actions, it becomes possible to code the sequences of narrative fragments clustered into themed stories in table 5.1. The opportunity is also taken to broadly code Apple and the actors in the observed tension simply as platform owner (P) and developer (D) respectively. The next step is to generate a table that shows themed sequences of generic actions. This analysis is presented in table 5.2, and through these results a set of narrative networks can be constructed.

	1	2	3	4	5
Story	**3G Skype**	**Admob**	**Adobe Flash**	**Adobe Developer Tools**	**Ari David**
Developer (D)	Skype	Google	Adobe	Adobe	Ari David
Action#1	*D Requests P*	D Request P	*D Request P*	*D Requests P*	*D Requests P*
Action#2	*P Blocks D*	*P Allows D*	*P Blocks D*	*P Allows D*	*P Blocks D*
Action#3	*D Influences P*	*P Blocks D*	*D Influences P*	*P Blocks D*	*D Influences P*
Action#4	*P Refines D*	*D Influences P*	*D Bypasses P*	*D Influences P*	*P Refines D*
Action#5		*P Refines D*	*D Regroups*	*P Refines D*	

	6	7	8	9	10
Story	**Baby Shaker**	**Big Brother Security**	**Bobble Rep**	**Box Office**	**C64 Emulator**
Developer (D)	Sikalosoft	Big Brother Security	Tom Richmond	Metasyntactic	Manamio
Action#1	*D Requests P*	*D Requests P*	*D Requests P*	*D Requests P*	*D Requests P*
Action#2	*P Allows D*	*P Allows D*	*P Blocks D*	*P Allows D*	*P Blocks D*
Action#3	*P Blocks D*	*P Blocks D*	*D Influences P*	*P Blocks D*	*P Influences D*
Action#4	*D Influences P*	*D Influences P*	*P Refines D*	*D Influences P*	*D Regroups*
Action#5		*D Regroups*		*P Refines D*	*P Allows D*
Action#6		*P Ignores D*			*P Blocks D*
Action#7					*P Refines D*

	11	12	13	14	15
Story	**CastCatcher**	**Convertbot**	**EFF Updates App**	**Eucalyptus**	**EyeTV**
Developer (D)	Amro Mousa	Convertbot	Exact Magic	James Montgomerie	Elgato
Action#1	*D Requests P*	*D Requests P*	*D Requests P*	*D Requests P*	*D Requests P*
Action#2	*P Allows D*	*P Blocks D*	*P Blocks D*	*P Blocks D*	*P Allows D*
Action#3	*P Blocks D*	*D Influences P*	*D Influences P*	*D Influences P*	*P Blocks D*
Action#4	*P Influences D*	*D Regroups*	*P Refines D*	*P Refines D*	*D Influences P*
Action#5	*D Regroups*	*P Allows D*			*P Refines D*
Action#6	*P Allows D*				

Table 5.2 Themed stories coded as sequences of generic actions.

	16	17	18	19	20
Story	**Financial Times**	**Google Books**	**Google Voice**	**GV Mobile**	**Hottest Girls**
Developer (D)	Financial Times	Google	Google	SK Dev Solutions	Allen Leung
Action#1	*D Requests P*	*D Requests P*	*D Request P*	*D Requests P*	*D Requests P*
Action#2	*P Allows D*	*P Allows D*	*P Allows D*	*P Allows D*	*P Allows D*
Action#3	*D Influences P*	*P Blocks D*	*P Blocks D*	*P Blocks D*	*P Blocks D*
Action#4	*D Bypasses P*	*D Influences P*	*D Influences P*	*P Influences D*	*D Influences P*
Action#5	*D Regroups*	*D Regroups*	*D Bypasses P*	*D Bypasses P*	
Action#6		*P Allows D*	*P Refines D*	*P Refines D*	

	21	22	23	24	25
Story	**I Am Rich**	**Jailbreakme**	**Mark Fiore**	**Me So Holy**	**Nescaline**
Developer (D)	Armin Heinrich	Comex	Mark Fiore	Lil' Shark	Jonathan Zdziarski
Action#1	*D Requests P*	*D Requests P*	*D Requests P*	*D Requests P*	*D Requests P*
Action#2	*P Allows D*	*P Blocks D*	*P Blocks D*	*P Allows D*	*P Allows D*
Action#3	*P Blocks D*	*D Regroups*	*D Influences P*	*P Blocks D*	*P Blocks D*
Action#4	*D Influences P*	*P Blocks D*	*P Refine D*	*D Influences P*	*D Influences P*
Action#5		*D Influences P*		*D Bypasses P*	*D Bypasses P*

	26	27	28	29	30
Story	**Netshare**	**NinjaWords**	**Opera**	**Podcaster**	**Pull My Finger**
Developer (D)	Nullriver	Matchstick Software	Opera Software ASA	Alex Sokirynsky	Eric Ogren
Action#1	*D Requests P*	*D Requests P*	*D Requests P*	*D Requests P*	*D Requests P*
Action#2	*P Allows D*	*P Blocks D*	*P Blocks D*	*P Blocks D*	*P Blocks D*
Action#3	*P Blocks D*	*D Influences P*	*D Influences P*	*D Influences P*	*D Influences P*
Action#4	*D Influences P*	*D Regroups*	*D Regroups*	*D Bypasses P*	*P Refines D*
Action#5	*P Allows D*	*P Refines D*	*P Allows D*	*D Regroups*	
Action#6	*P Blocks D*			*P Allows D*	

Table 5.2 Themed stories coded as sequences of generic actions (continued).

	31	32	33	34	35
Story	**Pulse News Reader**	**Readability**	**Routesy**	**Sekai Camera**	**Simply Beach**
Developer (D)	Alphonso Labs Inc	Readability	Steven Peterson	Tonchidot	Simply Beach
Action#1	*D Requests P*	*D Requests P*	*D Requests P*	*D Requests P*	*D Requests P*
Action#2	*P Allows D*	*P Blocks D*	*P Allows D*	*P Allows D*	*P Allows D*
Action#3	*P Blocks D*	*D Influences P*	*P Blocks D*	*P Blocks D*	*P Blocks D*
Action#4	*D Influences P*	*D Bypasses P*	*D Influences P*	*D Influences P*	*D Influences P*
Action#5	*D Regroups*		*P Refine D*	*D Regroups*	*P Refines D*
Action#6	*P Allows D*			*P Allows D*	

	36	37	38	39	40
Story	**Someecards**	**Stanza**	**Tawkon**	**TrapCall**	**Trillian**
Developer (D)	Duncan Mitchell	Lexcycle	Tawkon Ltd	Tel Tech Systems Inc	Cerulean Studios
Action#1	*D Requests P*	*D Requests P*	*D Requests P*	*D Requests P*	*D Requests P*
Action#2	*P Allows D*	*P Allows D*	*P Blocks D*	*P Ignores D*	*P Ignores D*
Action#3	*P Blocks D*	*P Blocks D*	*P Influences D*	*D Influences P*	*D Influences P*
Action#4	*D Influences P*	*D Influences P*	*D Bypasses P*	*P Allows D*	*P Allows D*
Action#5	*D Regroups*	*D Regroups*			
Action#6	*P Allows D*	*P Allows D*			
Action#7	*P Blocks D*				
Action#8	*D Regroups*				
Action#9	*P Allows D*				

	41	42	43	44	45
Story	**Tweetie**	**Voice Central**	**Wallpaper Gallery**	**Wallpaper Universe**	**Wi-Fi Sync**
Developer (D)	Loren Brichter	Riverturn	Start Mobile	Funmobility	Greg Hughes
Action#1	*D Requests P*	*D Requests P*	*D Requests P*	*D Requests P*	*D Requests P*
Action#2	*P Allows D*	*P Allows D*	*P Blocks D*	*P Allows D*	*P Blocks D*
Action#3	*P Blocks D*	*P Blocks D*	*D Influences P*	*P Blocks D*	*D Influences P*
Action#4	*D Influences P*	*D Influences P*	*P Refines D*	*D Influence P*	*D Bypasses P*
Action#5	*P Refines D*	*D Bypasses P*			

Table 5.2 Themed stories coded as sequences of generic actions (continued).

5.2.2 Constructing narrative networks

The purpose of constructing narrative networks is not just to visualize the sequence of actions in one particular story, but to allow a comparison of different stories and sequences of actions. The expectation is that by doing this then patterns of actions can be identified and interpreted.

Constructing individual narrative networks

The first step was to plot narrative networks for each of the stories that were identified. Figure 5.3 illustrates the narrative network that describes the sequence of actions that occurred in the story concerning the tussle between Google and Apple over Google Voice. By way of reference, the sequence of actions described in this narrative network is summarised as story #18 in table 5.2.

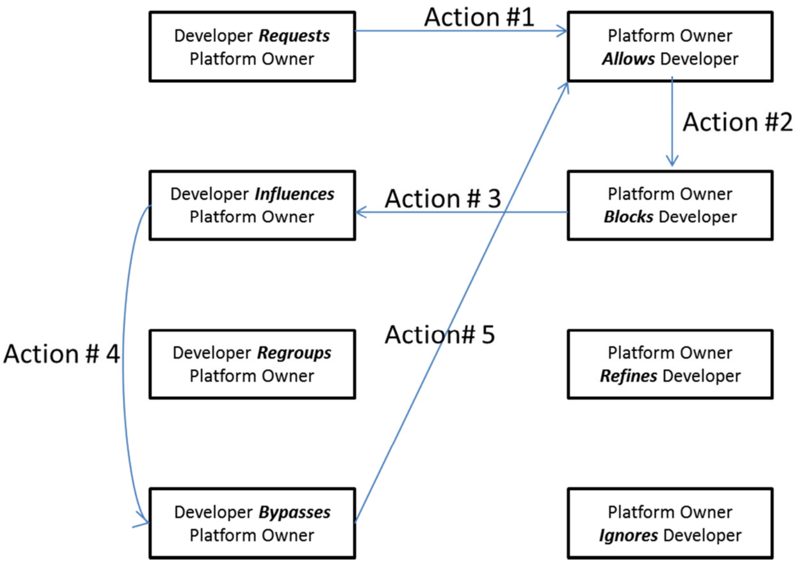

Developer *Requests* Platform Owner
Action #1
Platform Owner *Allows* Developer
Action #2
Developer *Influences* Platform Owner
Action # 3
Platform Owner *Blocks* Developer
Action # 4
Developer *Regroups* Platform Owner
Action# 5
Platform Owner *Refines* Developer
Developer *Bypasses* Platform Owner
Platform Owner *Ignores* Developer

Figure 5.3 Illustrative narrative network of Google Voice

As in all the narrative networks that were built, the illustration contains eight nodes, each representing one of the possible narrative fragments that were coded. These nodes are linked with arcs in the sequence in which events unfold. In the case of Google Voice, which is illustrated, the nodes are linked as follows. Google attempted to launch ("request") its application officially on the iPhone on the 14th July 2009 (Action #1). In parallel, a number of third party developers had already launched ("allow") applications based on Google Voice on the App Store.

Following Apple's initial acceptance ("allow") of Google Voice onto the App Store (Action #2), Apple then reversed its decision and ejected ("block") the app from the App Store on the 27th

July 2009 (Action #2). Google’s initial response was to wait – and lobby alternative channels - whilst other actors investigated the matter on its behalf (“influence”) (Action #3), the Federal Communications Commission wrote to Apple on the 31st July 2009 to inquire why Google Voice had not been approved. Apple’s response on the 21st August 2009 was that Google Voice was rejected from the App Store because it interferes with the iPhone's "distinctive user experience".

Apple contended by declaring that Google Voice "appears to replace the iPhone's core mobile telephone functionality and Apple user interface with its own user interface for telephone calls, text messaging and voicemail." As such Apple continued to block this application and Google appeared to continue doing nothing. Then on the 26th January 2010 Google launch Google Voice for the iPhone as a web application (Action #4), this circumventing Apple’s control of native applications on the App Store through its approval process. In this case Google’s action was to “bypass” Apple. It could have continued doing nothing, or as in the case of other stories it could have tried to “regroup” by attempting to launch other applications, not related to Google Voice, on the iPhone. The risk for Apple by carrying out a “blocking” action is that the developer will carry out some form of “bypass” action, with the net result of being beyond Apple’s zone of control. This is what happened with Google Voice. Ultimately Apple relented (Action #5), and on the 27th September 2010 Apple approved (“allowed”) Google Voice for the App Store. These narrative networks, plotted for individual stories, succeed in simplifying and making sense of what would appear to be complex interactions that result as a consequence of the tensions between Apple and its developers.

Combined narrative networks

Once narrative networks were plotted for all the stories uncovered in the data concerning tensions on Apples platform, it was possible to combine them on one narrative network in order to facilitate comparison and analysis. In order to do this, the technique detailed in chapter 4 was applied. First, the number of ties, between each pair of nodes across the different value networks, was summated. These were then summarised in a first order Markov model as a matrix. The matrix, which was made up of frequencies of transitions between events, was then normalised. Finally, a “normalised” narrative network was constructed, where the thickness of the ties is proportional to the number of instances of transiting from one node to another. This combined narrative network is illustrated in figure 5.4.

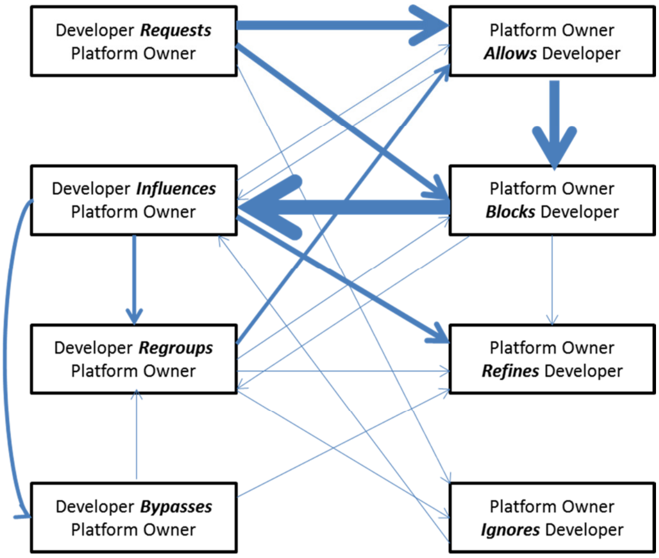

Developer *Requests* Platform Owner
Developer *Influences* Platform Owner
Developer *Regroups* Platform Owner
Developer *Bypasses* Platform Owner
Platform Owner *Allows* Developer
Platform Owner *Blocks* Developer
Platform Owner *Refines* Developer
Platform Owner *Ignores* Developer

Figure 5.4 Combined narrative network.

5.3 Describing a generalised sequence of actions

The analysis of narrative networks represents the sixth and final step of adaptation of the approach by Pentland and Feldman (2007) to constructing and analysing narrative networks. The objective here then is to first simplify and then interpret the combined narrative network, in order to provide a generalised description of contested platform innovation as a phased patterned sequence of actions.

The combined narrative network in figure 5.4 may look chaotic. So it should, as there are many possible ways to go about contesting innovation. However, it can be simplified, broken down and interpreted in order to reveal a structured pattern of actions. These structured patterns of actions reveal: 1) challenges to the boundary of permissible service innovation on a digital platform; 2) tensions that occur between platform owners and third party developers; and 3) resolutions to contested platform innovation.

5.3.1 Abstracting the combined narrative network

By revisiting the Markov matrix that forms the basis of the combined narrative network in figure 5.4, it becomes possible to produce a simplified network by filtering out links between nodes that transition in less than five stories. The resulting simplified narrative network is plotted in figure 5.5, and represents the most common transitions between nodes that are made. The thickness of the ties between nodes continues to be proportional to the total number of transitions made across all 45 stories.

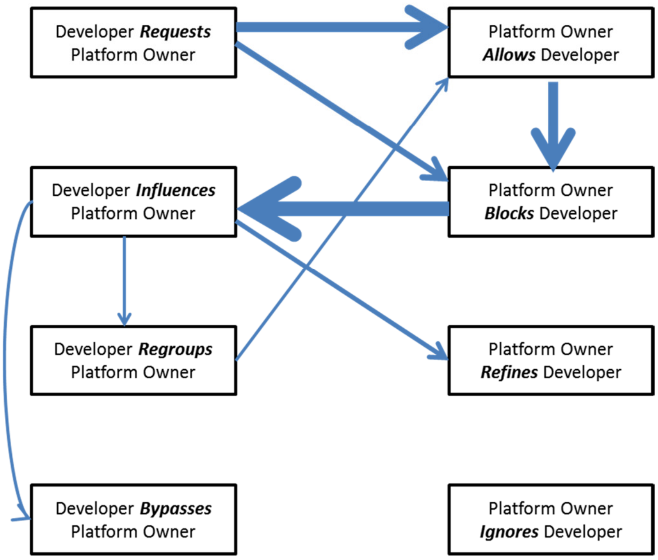
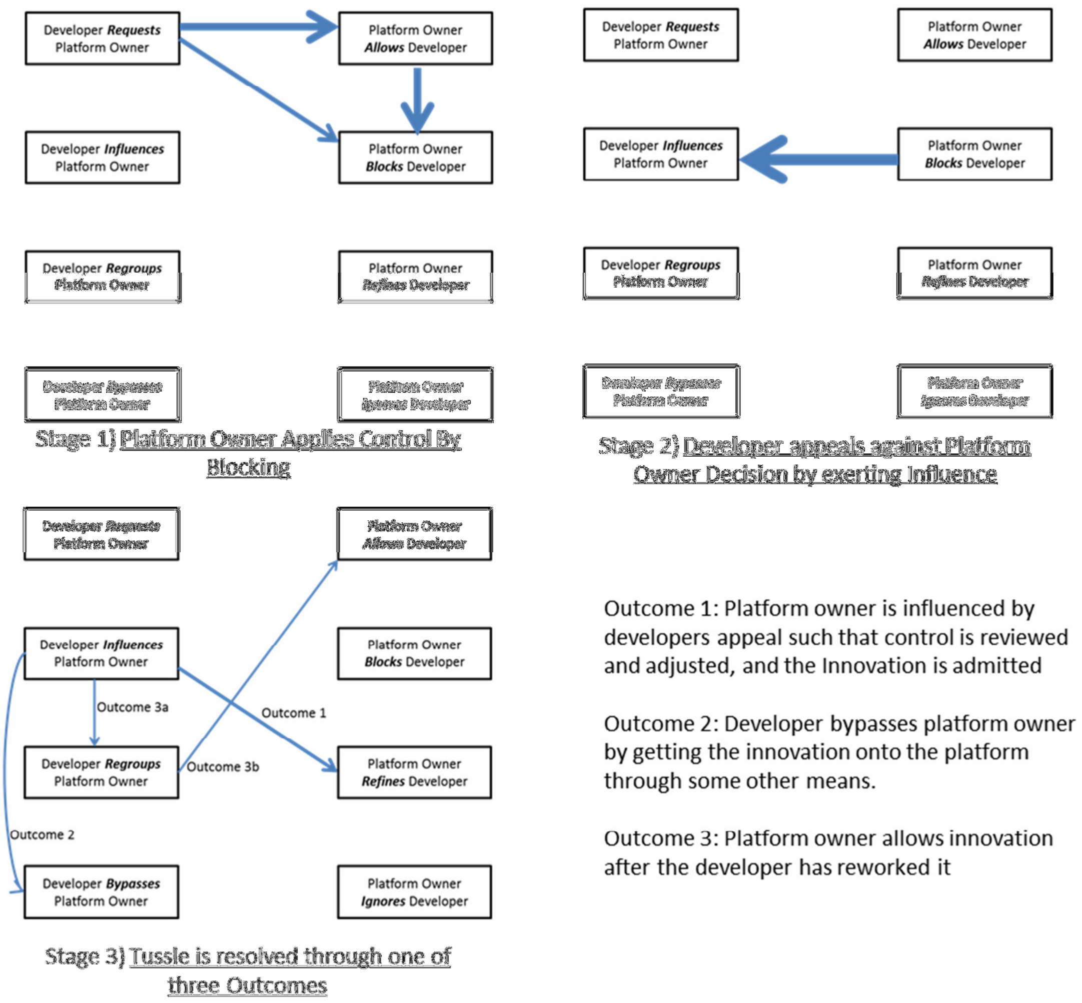

Developer *Requests* Platform Owner
Platform Owner *Allows* Developer
Developer *Influences* Platform Owner
Platform Owner *Blocks* Developer
Developer *Regroups* Platform Owner
Platform Owner *Refines* Developer
Developer *Bypasses* Platform Owner
Platform Owner *Ignores* Developer

Figure 5.5 Simplified narrative network showing common transitions.

5.3.2 Interpreting and describing the combined narrative network

The simplified narrative network shown in figure 5.5 can be broken down further to reveal three broad stages of tussle that occur across the 45 examples of contested innovation. These three stages are illustrated in figure 5.6.


Developer Requests Platform Owner
Platform Owner Allows Developer
Developer Influences Platform Owner
Platform Owner Blocks Developer
Developer Regroups Platform Owner
Platform Owner Refines Developer
Developer Bypasses Platform Owner
Platform Owner Ignores Developer
Stage 1) Platform Owner Applies Control By Blocking
Stage 2) Developer appeals against Platform Owner Decision by exerting Influence
Outcome 3a
Outcome 1
Outcome 3b
Outcome 2
Stage 3) Tussle is resolved through one of three Outcomes

Outcome 1: Platform owner is influenced by developers appeal such that control is reviewed and adjusted, and the Innovation is admitted

Outcome 2: Developer bypasses platform owner by getting the innovation onto the platform through some other means.

Outcome 3: Platform owner allows innovation after the developer has reworked it

Figure 5.6 Three stages of interaction in contested innovation

Stage 1: Build up

All of the examples of contested innovation identified in the data show attempts by third party developers to challenge the boundary of innovation permitted by Apple. Each example starts with the respective developer requesting that code in the form of an app or a platform enabler be admitted into the App Store. It is then generally either blocked, or allowed and then blocked by Apple. This is illustrated as stage 1 in figure 5.6, where Apple applies control by ultimately blocking the developer's innovation. Many attempts at innovation, such as Adobe Flash (#3), Jailbreakme (#22) or Podcaster (#29) were blocked outright, as they contravened Apple's commercial interests and hence the boundary for permissible service innovation. The majority of our examples, such as Google's Admob (#2), Adobe Developer Tools (#4), I Am Rich (#21), and Netshare (#26), were initially allowed and later blocked, after Apple determined that these innovations were no longer desirable. In this latter case of control, these innovations can be thought of as being on the cusp of the boundary of permissible innovation.

For the sake of clarity figure 5.6 stage 1 does not show the two exceptions to the rule, TrapCall (#39) and Trillian (#40), where Apple’s response was simply to ignore the respective developers requesting for their apps to be admitted on to the App Store. In these circumstances Apples ignoring action is very similar to a blocking action.

Stage 2: Tension

Apple’s blocking (and ignoring) actions produced a response that leads to tensions in all of the example stories extracted from the data. Apple’s response effectively marks the beginning of an instance of contested innovation. Stage 2 of figure 5.6 shows the developer contesting Apple's decision by attempting to influence Apple to change its mind. Developers attempt to influence Apple directly, for example by complaining, or indirectly via a regulator, the press or their user base. The most extreme example of a developer attempting to influence Apple directly, and indirectly, was Adobe (#3 and #4) who initiated an advertising campaign communicating their “love for Apple” during the conflicts between the two concerning the use of Flash. Google attempted to influence Apple using regulators in the form of the Federal Communications Commission (FCC) in the example of Google Voice (#18) and the Federal Trade Commission (FTC) in the example of AdMob (#2). By implication of the method used (sourcing data from blogs), all of the examples used the press, or at least blogs, to put forward their cases, and in many cases these reports gave disgruntled users a voice.

Stage 3: Resolution

The third and final stage of the simplified narrative network, illustrated in figure 5.6, marks the resolution of the tussle between the developer and Apple. The figure shows three typical outcomes following the developer's initial attempt at contesting Apple's blocking decision through influence. This third stage typically represents the end of a tussle with some kind of resolution as its end product.

The first outcome shows the platform owner sufficiently influenced by the developer's appeal such that control is reviewed and adjusted, and the innovation is admitted on to the platform. This first outcome was seen for example in the stories of Adobe Developer Tools (#4), Box Office (#9) and Mark Fiore (#23).

The second outcome shows the developer adopting a strategy of bypassing the platform owner by distributing the innovation to iOS users through some other means. This second outcome is exemplified for example by the Financial Times (#16), which ended up distributing their iOS version of their newspaper as a Web 2.0 app rather than as a native app on the App Store. Another example of this outcome is demonstrated in the story of the PlayStation emulator app Nescaline (#25), which after it was removed from the App Store, saw variants placed on alternative app stores for jailbroken iOS devices.

The third and final type of observed outcome follows on from Apple not responding to attempts at influence by the developer. In this outcome, the developer uses a regrouping strategy. The act of regrouping might entail simply giving up and trying something different, or in the case of many of the stories found in the data, for example CastCatcher (#11) or Convertbot (#12), the developer reworking the innovation so that it is acceptable to Apple, who then allow the reworked innovation.

PART II: EXPLAINING THE DYNAMICS OF THE CONTROL OF PLATFORM INNOVATION

The construction and analysis of narrative networks addresses the first sub-question by describing the dynamics of control within instances of contested platform innovation. The second sub-question is concerned with explaining the dynamics of the control of innovation, and this is now addressed in the three sub sections that follow. The first examines the mechanism of control that Apple uses, in order to assert control. The second explains how the developers respond to control through the use of their own mechanism. The third and final sub section attempts to find relationships between the mechanisms of control and the mechanisms which respond to control in order to explain the outcomes that are achieved across the stories.

5.4 Analysis of Apple's app approval process as a control mechanism

To examine how Apple attempts to assert control on its platform I use the theory of formal organisational control (Ouchi 1977; Ouchi 1978; Ouchi 1979). In particular I examine Apple's Apps Approval Process, which is the formal mechanism used to control whether a developer's innovation is admitted into the iOS platform ecosystem. To do this I examine the use of this control mechanism across the 45 instances of contested innovation found within the empirical data. This analysis attempts to understand which modes of control, behavioural and outcome control, are used as well as how the four elements of a formal control mechanism, identified by (Eisenhardt 1985), are applied within the Apps Approval Process. The four elements of formal control concern: Specification; Measurement; Evaluation; and Rewards/Sanctions.

The analysis is structured in three parts. First, an overview of the four elements of the mechanism is discussed. It is found that the element of specification varies the most across the stories. Second, the stories are analysed, grouped and categorised in terms of the criteria for control that is applied to each. Specifications for control, modes of control and the type of sanction or reward are then mapped to each of the criteria for control. Finally, the results of this categorisation are discussed in order to summarise insights made into the Apps Approval Process as a mechanism of control.

5.4.1 Overview of the app approval process as a control mechanism

The analysis is simplified by the fact that some of the elements are either readily apparent, or constrained by a lack of data. Each element of the control is now briefly taken in turn.

First, the element of reward or sanction used within this mechanism is simply the admittance or non-admittance of the developer's innovation into Apple's App Store and the iOS platform ecosystem. These rewards and sanctions remain the same across all 45 instances of contested innovation that were found.

Second, no substantial information concerning the structure or functioning of the measurement and evaluation elements of the Apps Approval Process could be found within the data. For example, it was not possible to ascertain whether the process is completely manual or partly automated or how many people Apple employs to carry out these functions. This apparent lack of public data is not surprising, given the secretive nature of Apple as an organisation. Consequently, no insight was generated with respect to these two elements of the mechanism.

Third, Apple uses a wide range of specifications, against which it measures new applications, and evaluates whether they should be allowed into the App Store, or rejected. Apple presents these specifications in two documents, the iOS Developer Program License Agreement, and the App Store Review Guidelines. Whilst Apple attempts to keep these documents confidential to Apple and developers only, copies can be found on the internet.

Finally, the mode of formal control for a particular specification can be ascertained by the wording of the corresponding entry within the Apple documents. Specifications referring to how apps are to be produced correspond to a behavioural mode of control. Specifications referring to the characteristics a submitted app should, or should not have, corresponded to outcome control.

Consequently my analysis of the Apps Approval Process is limited to examining the specifications against which Apple decides whether platform innovation should occur, and matching them to instances of contested innovation. Nevertheless a range of insights can be made from this analysis.

5.4.2 Analysis of criteria and specifications of control

The tables, which follow, describe how the Apps Approval Process was applied, as a control mechanism, to the 45 instances of contested innovation.

The first table categorises the criteria, or reasons, for Apple rejecting the apps described in the stories. The blog data was examined once more and the reasons Apple gave for rejecting particular apps were extracted. These were interpreted, such that common criteria for rejection were found amongst the stories. Table 5.3 lists these common criteria and the stories that were correspondingly affected.

Motivation	Criteria	Apple Rationale for blocking App	Narrative
Competing Platforms	Executable Code from other Platforms	Risk of losing control of platform if alternative platforms are permitted to function as an overlay to iOS	3 Adobe Flash
	Cross-Compilers	Desire to control the iOS development environment to ensure quality of apps	4 Adobe developer tools
	Platform Emulators	Risk of losing control of platform if autonomous platforms are permitted to function as an overlay to iOS	10 C64 Emulator
			25 Nescaline
	Jailbreaking	Risk of losing control of platform if unregulated apps are permitted on iOS	22 Jailbreakme
	Alternative Advertising Platforms	Desire to block competing mobile advertising platforms from platform ecosystem	2 Admob
	Alternative Payment Platforms	Desire to maintain control over profitable revenues from in app purchasing	16 Financial Times
			17 Google Books
			32 Readability
Legal Action	Data Collection and Privacy Abuse	Risk of legal action by not protecting customers personal data	7 Big Brother Security
	Threat to Apple's Legal Position	Apple protects itself against legal action taken by third parties concerning the content or nature of developers apps	31 Pulse News Reader
			33 Routesy
Brand Image	Dubious Value	Desire to protect its brand by providing apps that are useful or have lasting entertainment value	21 I Am Rich
	Duplication of Functionality	Desire to maintain an advantageous position for its own content and apps	18 Google Voice
			19 GV Mobile
			28 Opera
			29 Podcaster
			42 VoiceCentral
	Appropriate Look and Feel	Desire to maintain control of its branded look and feel	12 Convertbot
	Objectionable Content	Apple’s desire to prevent content and apps from entering its platform ecosystem which might be objectionable to its customer base, and hence damage its brand image	5 Ari David
			6 Baby Shaker
			13 EFF Updates App
			14 Eucalyptus
			23 Mark Fiore
			24 Me So Holy
			27 NinjaWords
			35 Simply Beach
			36 Someecards
			44 Wallpaper Universe
	Use of Private APIs	Desire to prevent developers using undocumented and unstable APIs which might damage platform’s reputation	34 Sekai Camera
			38 Tawkon
			45 Wi-Fi Sync
Network Partners	Excessive Cellular Data Usage	Risk of upsetting valuable commercial partners by overloading their networks with data	11 CastCatcher
			15 EyeTV
			26 Netshare
	Enabling VoIP over 3G	Risk of upsetting valuable commercial partners by promoting products that cannibalise their revenues	1 3G Skype
Unknown	Unknown	Apple provided no reason for blocking app	9 Box Office
			39 TrapCall
			40 Trillian

Table 5.3 Apple’s motivations and criteria for applying control

Table 5.3 also shows how these criteria for rejection can be further categorised by underlying themes, which were interpreted as Apple's underlying motivation for applying control. Furthermore these categories of motivations for Apple applying control loosely correspond with the headings Apple uses for groups of specifications in its iOS Developer Program License Agreement. The criteria, that were identified and that fall under these motivations, correspond with individual specifications outlined in the iOS Developer Program License Agreement and the App Store Review Guidelines.

Apple's motivations for control are summarised as follows. The first motivation concerns Apple’s desire to maintain sole control over the content and applications that reside on its platform. In this way it seeks to protect iOS by preventing other platforms, whose content it cannot control, from functioning on iOS. The second motivation concerns Apple seeking to protect itself from legal action taken by third parties as a result of the content or functionality of the apps it distributes on behalf of developers. The third motivation concerns Apple seeking to control the content, functionality and look and feel of the apps that it distributes in order to maintain the quality of its platform and its brand image and reputation. The fourth motivation concerns Apple’s need to maintain good commercial relationships with its network partners, who it relies upon in order to distribute its handsets to consumers. As a consequence of this it seeks to ensure that the functionality of the apps that it distributes is not damaging to its partners’ network performance and network revenues. The fifth and final motivation is a catch all for unknown criteria for control. This motivation exists as there were occasions, based on the collected data, where the reason for Apple blocking an innovation from the App Store was unknown.

The next level of analysis of the Apps Approval Process concerned mapping of Apple's specifications for control onto the criteria and motivations for control that were previously found. As this was done, the corresponding specification was analysed in terms of which mode of control it was facilitating and whether compliance would be met with a sanction or reward. This analysis was carried out for each of the sixteen criteria for control that were identified amongst the stories. Table 5.4 provides a summary of the analysis which was carried out for the criteria concerning the rejection of apps that contain objectionable content. It can be seen from the table that the corresponding references to Apple's specifications are given, and that it forms an outcome mode of control. Furthermore the table lists a brief history and background to the criteria and specification as found amongst the blog data.

<table>
<tr><td colspan="2"><b>Criteria:</b> Objectionable Content</td></tr>
<tr><td colspan="2"><b>Specification:</b> 3.3.18 of iPhone Developer Program License Agreement
<b>Rev 20-10-08: 3.3.12</b> Applications may be rejected if they contain content or materials of any kind (text, graphics, images, photographs, sounds, etc.) that in Apple’s reasonable judgment may be found objectionable, for example, materials that may be considered obscene, pornographic, or defamatory.
<b>Revs 1-22-10 & 17-3-09: 3.3.14</b> Applications may be rejected if they contain content or materials of any kind (text, graphics, images, photographs, sounds, etc.) that in Apple’s reasonable judgment may be found objectionable, for example, materials that may be considered obscene, pornographic, or defamatory.
<b>Rev 9-9-10 onwards: 3.3.18</b> Applications may be rejected if they contain content or materials of any kind (text, graphics, images, photographs, sounds, etc.) that in Apple’s reasonable judgment may be found objectionable, for example, materials that may be considered obscene, pornographic, or defamatory.

14, 15, 16, 18, 19 of the App Store Review Guidelines (Rev 9-9-10 onwards)
<b>14.1</b> Any app that is defamatory, offensive, mean-spirited, or likely to place the targeted individual or group in harm’s way will be rejected
<b>14.2</b> Professional political satirists and humorists are exempt from the ban on offensive or mean-spirited commentary
<b>15.1</b> Apps portraying realistic images of people or animals being killed or maimed, shot, stabbed, tortured or injured will be rejected
<b>15.2</b> Apps that depict violence or abuse of children will be rejected
<b>15.3</b> "Enemies" within the context of a game cannot solely target a specific race, culture, a real government or corporation, or any other real entity
<b>15.4</b> Apps involving realistic depictions of weapons in such a way as to encourage illegal or reckless use of such weapons will be rejected
<b>15.5</b> Apps that include games of Russian roulette will be rejected
<b>16.1</b> Apps that present excessively objectionable or crude content will be rejected
<b>16.2</b> Apps that are primarily designed to upset or disgust users will be rejected
<b>18.1</b> Apps containing pornographic material, defined by Webster's Dictionary as "explicit descriptions or displays of sexual organs or activities intended to stimulate erotic rather than aesthetic or emotional feelings", will be rejected
<b>18.2</b> Apps that contain user generated content that is frequently pornographic (ex "Chat Roulette" apps) will be rejected
<b>19.1</b> Apps containing references or commentary about a religious, cultural or ethnic group that are defamatory, offensive, mean-spirited or likely to expose the targeted group to harm or violence will be rejected
<b>19.2</b> Apps may contain or quote religious text provided the quotes or translations are accurate and not misleading. Commentary should be educational or informative rather than inflammatory</td></tr>
<tr><td colspan="2"><b>Mode:</b> Outcome Control</td></tr>
<tr><td><b>Date</b></td><td><b>History</b></td></tr>
<tr><td>20/02/2010</td><td>Apple aims to uphold the quality of its App Store by providing applications that are "within good taste". Consequently it attempts to weed out applications that do not fulfill these criteria. However in the early days of the App Store these rules were not made explicit. Indeed when they were enforced, they often seemed to be applied subjectively, thoughtlessly and arbitrarily.</td></tr>
<tr><td>25/02/2010</td><td>Commentators rail against Apple's hypocritical behaviour, which appears to banish risqué offerings the small developers whilst allowing those from established brands</td></tr>
<tr><td>16/04/2010</td><td>Developers then begin to complain about Apple's "rules", particularly section 3.3 within the iOS Developer Program License concerning the criteria by which apps are to be developed , as being too vague</td></tr>
<tr><td>09/09/2010</td><td>In response Apple releases a clear set of App Store Review Guidelines, to supplement the iOS Develop Program License Agreement, in order to clarify what type of apps are allowed and what apps are not allowed into the App Store</td></tr>
</table>

Table 5.4 Example of mapping of specifications to a criteria of control.

A summary of all the individual tables, that were created but not illustrated in this dissertation, is given in table 5.5. It shows the five different motivations identified for control, along with their corresponding criteria for control and references to where this control is specified in Apple's documentation. Furthermore it indicates what mode of control is associated with the specification, whether compliance will be sanctioned or rewarded, and finally whether developers have had any influence in changing the specification. As will be discussed shortly, it was identified in the histories of certain criteria, that developers have on occasion been able to influence Apple's specifications.

Motivation	Criteria	Specification	Mode	Sanction/Reward	Developer Influence
Competing Platforms	Executable Code from other Platforms	iOS Developer Program License Agreement. Section 3.3.2	Outcome	Sanction	No
	Cross-Compilers	iOS Developer Program License Agreement. Section 3.3.1	Behavioural	Reward	Yes
	Platform Emulators	iOS Developer Program License Agreement. Section 3.3.2	Outcome	Sanction	Yes
	Jailbreaking	iOS Developer Program License Agreement. Section 3.2e	Outcome	Sanction	No
	Alternative Advertising Platforms	iOS Developer Program License Agreement. Section 3.3.9	Outcome	Sanction	Yes
	Alternative Payment Platforms	App Store Review Guidelines. Sections 11.12, 11.13 and 11.14	Outcome	Sanction	Yes
Legal Action	Data Collection and Privacy Abuse	iOS Developer Program License Agreement. Sections 3.3.8, 3.3.9, 3.3.10, 3.3.11 and 3.3.12	Outcome	Sanction	No
	Threat to Apple's Legal Position	iOS Developer Program License Agreement. Section 11	Outcome	Sanction	No
Brand Image	Dubious Value	App Store Review Guidelines. Section 2.12	Outcome	Sanction	No
	Duplication of Functionality	App Store Review Guidelines. Sections 2.11 & 10.2	Outcome	Sanction	No
	Appropriate Look and Feel	iOS Developer Program License Agreement. Section 3.3.7	Both	Reward	No
	Objectionable Content	iOS Developer Program License Agreement . Section 3.3.18. and App Store Review Guidelines Sections 14, 15, 16, 18, 19	Outcome	Sanction	Yes
	Use of Private APIs	iOS Developer Program License Agreement. Section 3.3.1	Behavioural	Sanction	No
Network Partners	Excessive Cellular Data Usage	iOS Developer Program License Agreement. Section 3.3.19	Outcome	Sanction	No
	Enabling VoIP over 3G	iOS Developer Program License Agreement. Section 3.3.28	Outcome	Sanction	Yes
Unknown	Unknown	Unknown		Sanction	No

Table 5.5 Summary of the characteristics of control by criteria and specification.

5.4.3 Insights into Apple's app approval process as a control mechanism

Three observations can be made on the basis of the previous analysis of the Apps Approval Process as a control mechanism.

Emphasising sanctions rather than rewards

The first concerns the way that the control mechanism is specified and the predominant mode of control that is applied. Most of the specifications within Apple's Apps Review Process concern what outcomes or behaviours will be sanctioned rather than rewarded.

For example the specification 3.2e of the License Agreement concerns implicit sanctions with respect to the criteria of Jailbreaking:

"You will not, through use of the Apple Software, services or otherwise, create any Application or other program that would disable, hack or otherwise interfere with the Security Solution, or any security, digital signing, digital rights management, verification or authentication mechanisms implemented in or by the iOS, this Apple Software, any services or other Apple software or technology, or enable others to do so;"

Specification 3.3.7 of the License Agreement is related to the criteria concerning the appropriate look and feel of apps. It is one of only two criteria that implicitly reward compliance:

"Applications must comply with the Human Interface Guidelines and other Documentation provided by Apple."

The reason for this may come about as result of Apple's opening up innovation of iOS apps to third party developers. In order that it maximises the value of its iOS platform, Apple attempts to foster a wide a range of innovative apps. Rather than constrain developers by specifying apps that it perceives as desirable, which may possibly limit the scope for breadth of innovation, it is easier to specify features that it does not want, and thus maintain scope for apps that it has not conceived of.

Emphasising outcome rather than behavioural control

The second is related to the predominant mode of control. Most of the specifications that were applied across the tussles identified in the data concern outcome modes of control rather than behavioural modes of control.

For example, specification 15.5 of the App Store Review Guidelines is related to the criteria of objectionable content and specifies an undesirable outcome:

"Apps that include games of Russian roulette will be rejected"

On the other hand, specification 3.3.1 of Developer License is related to the criteria prohibiting the use of private APIs, and details desired design behaviour:

"Applications may only use Documented APIs in the manner prescribed by Apple and must not use or call any private APIs."

The application of behavioural control requires knowledge by the controlee of the task program, or the programmed set of activities and behaviours required to complete a task (Eisenhardt 1985). It is challenging to specify a task program, when the specified outcome of a task, in the form of an innovative app, is unknown. On the few occasions that it does specify behavioural control, its preference is for prohibiting rather than encouraging behaviours. This is seen in the prohibition of the use of cross-compilers and the prohibition of the use of private APIs. The third example of behavioural control concerns the look and feel of applications, whose according specification mandates the use of Apple's Human Interface Guidelines. It is presumed that Apple has sufficient knowledge and understanding of the look and feel of its desired applications, so that it can mandate how these features are designed.

APPLE'S SPECIFICATIONS CAN BE INFLUENCED AND DO CHANGE

The final observation is about the ability of developers to influence Apple's specification of the control mechanisms over time. This occurred across five criteria, namely in the prohibition of: cross-compilers; platform emulators; alternative advertising platforms; alternative payment platforms; and enabling VoIP over 3G. Developers were able to contest these rules and the subsequent blocking of their innovations. The apps were later allowed into the App Store, following these challenges, and the corresponding specifications in the iOS Developer Program License Agreement and App Store Review Guidelines were adjusted accordingly.

For example, specification 11.14 of the License Agreement is related to the criteria concerning alternative payment platforms. This specification changed between the version of the agreement released on 15th February 2011 and the version released on the 9th June 2011. The reason for this change was that publishers who distributed content, such as magazines, on iOS apps complained about the new terms that Apple introduced in early 2011 regarding in app purchases of subscriptions. The first version is as follows:

***Rev 02-15-2011:** "Apps can read or play approved content (magazines, newspapers, books, audio, music, video) that is sold outside of the app, for which Apple will not receive any portion of the revenues, provided that the same content is also offered in the app using IAP at the same price or less than it is offered outside the app. This applies to both purchased content and subscriptions."*

The revised version reads as follows:

***Rev 06-09-2011:** "Apps can read or play approved content (specifically magazines, newspapers, books, audio, music, and video) that is subscribed to or purchased outside of the app, as long as there is no button or external link in the app to purchase the approved content. Apple will not receive any portion of the revenues for approved content that is subscribed to or purchased outside of the app"*

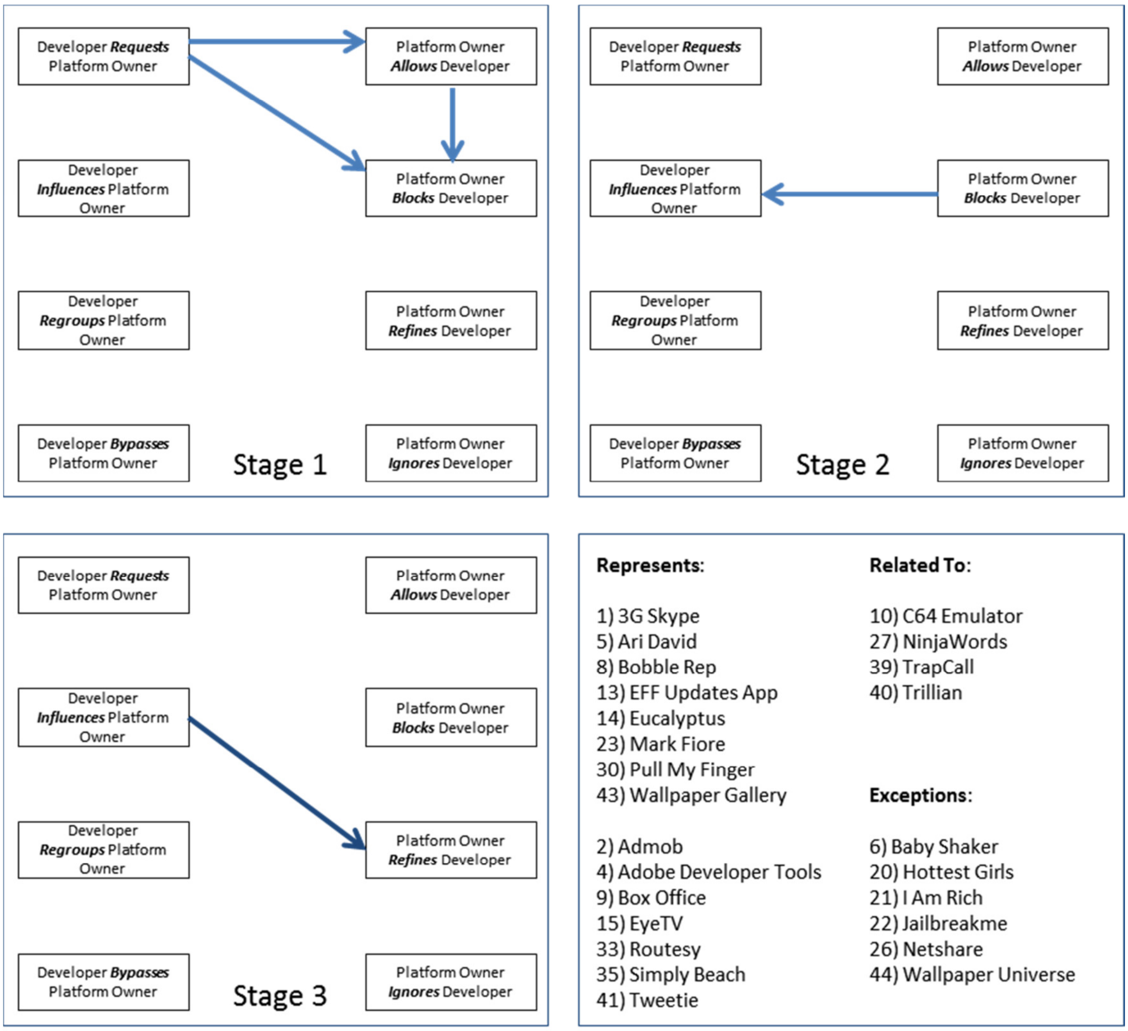
5.5 Analysis of developers' responses to the control of innovation

The abstracted sequence of events, uncovered by the narrative networks, indicates that developers typically respond Apple's attempts at controlling their attempting innovations. Following on from the analysis of Apple's Apps Review process as a control mechanism, attention is now turned to examining how developers attempt to overcome Apple's controlling actions.

Each of the three mechanisms of response used by developers is now considered and described in turn. Following this, there is a brief discussion in terms of how the developers' responses generate insights into Apple's control of platform innovation.

5.5.1 Mechanism of response #1: Influence

This mechanism involves the developer attempting to influence Apple directly and indirectly, in order to persuade them to allow the innovation after all. Indirect influence takes the form of appealing to third parties such as regulators (as in stories #3 Admob and #18 Google Voice) and the media (as in all the stories covered in this research and whose details were found in public blogs).


Developer *Requests* Platform Owner
Platform Owner *Allows* Developer
Developer *Influences* Platform Owner
Platform Owner *Blocks* Developer
Developer *Regroups* Platform Owner
Platform Owner *Refines* Developer
Developer *Bypasses* Platform Owner
Platform Owner *Ignores* Developer
Stage 1
Stage 2
Stage 3

Represents:

1) 3G Skype
5) Ari David
8) Bobble Rep
13) EFF Updates App
14) Eucalyptus
23) Mark Fiore
30) Pull My Finger
43) Wallpaper Gallery

2) Admob
4) Adobe Developer Tools
9) Box Office
15) EyeTV
33) Routesy
35) Simply Beach
41) Tweetie

Related To:

10) C64 Emulator
27) NinjaWords
39) TrapCall
40) Trillian

Exceptions:

6) Baby Shaker
20) Hottest Girls
21) I Am Rich
22) Jailbreakme
26) Netshare
44) Wallpaper Universe

Figure 5.7 Implementation of the first mechanism of response: Influencing

The typical sequence of events in this mechanism is detailed in figure 5.7. Typically at the end of this sequence, and when the strategy is successful, Apple makes an exception for the app and similar innovations, and refines the specification of its control so that similar occurrences are not repeated.

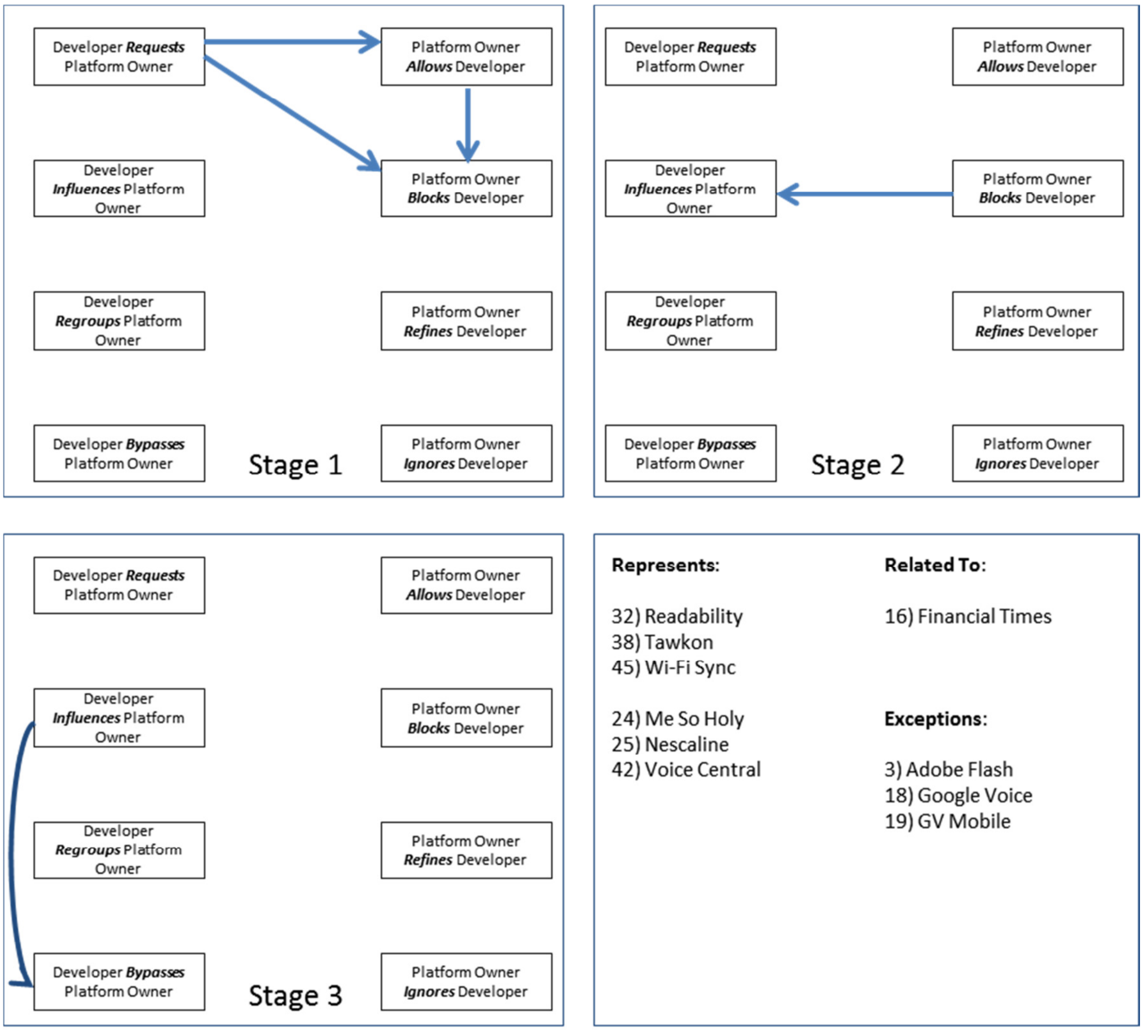
The sequence of actions in figure 5.7 directly represents a number of stories. For example, in stories #8 Bobble Rep, #23 Mark Fiore, and #43 Wallpaper Gallery in which satirical representations of political figures were made. These Apps were initially blocked on account of Apple interpreting them as containing objectionable content. However when Apple was made aware that these apps were created by professional, well known, and in the case of Mark Fiore, Pulitzer Prize winning journalists or artists they were allowed onto the App Store. Furthermore Apple changed its rules regarding objectionable content to make exceptions for professional satirists.

This sequence of events is also related to applications such as C64 Emulator (#10) and Triallian (#40). These stories followed a more elaborate version of the influence strategy involving a different or slightly larger sequence of actions. For example Trillian (#40) was ignored by Apple when it first requested presence on the App Store. After its developer influenced Apple by complaining, it was subsequently allowed.

Finally, there are a number of stories where developers tried and failed to apply an influence (and refine strategy). These are listed in the figure as exceptions, and each one of them blatantly breaks Apple's agreements and guidelines such that their block was non-negotiable.

5.5.2 Mechanism of response #2: Bypass

The bypass mechanism involves the developer distributing their innovation on iOS through some other channel, other than the official Apple App Store, once the app has been blocked. Within the data set used in this research, this strategy is carried out after the developer has attempted to influence Apple, since all the stories are reported on blogs, which is an influencing move. The typical sequence of events in this mechanism is detailed in figure 5.8.


Developer **Requests** Platform Owner
Platform Owner **Allows** Developer
Developer **Influences** Platform Owner
Platform Owner **Blocks** Developer
Developer **Regroups** Platform Owner
Platform Owner **Refines** Developer
Developer **Bypasses** Platform Owner
Platform Owner **Ignores** Developer
Stage 1
Stage 2
Stage 3

Represents:

32) Readability
38) Tawkon
45) Wi-Fi Sync

24) Me So Holy
25) Nescaline
42) Voice Central

Related To:

16) Financial Times

Exceptions:

3) Adobe Flash
18) Google Voice
19) GV Mobile

Figure 5.8 Implementation of the second mechanism of response: Bypassing

Developers who attempt to bypass Apple typically follow one of two distribution channels. The first means is by redeveloping the native iOS application as a web app based on HTML 5 that can simply be accessed via an HTML 5 iOS compatible web browser. This is the route taken by Google Voice (#18), GV Mobile (#19), VoiceCentral (#42), Readability (#32) and the Financial Times (#16). The second means is by distributing the native iOS app via an alternative iOS app store, such as Cydia. However, this distribution channel is only accessible by those users who jailbreak their phone, which is an activity that Apple strongly discourages. Nevertheless this is the route taken by four apps, Me So Holy (#24), Nescaline (#25), TawkOn (#38), and Wi-Fi Sync (#45).

The Financial Times (#16) is related to this strategy, simply because it deviates slightly from the pattern in figure 5.8, by effectively withdrawing itself (regrouping) from the App Store. Adobe

Flash (#3), Google Voice (#18), and GV Mobile (#19) are listed as exceptions to the strategy, as they were all eventually admitted onto the official Apple App Store. In the case of Adobe Flash (#3) it was partially enabled on the App Store by another app (Skyfire) which translates Flash content into HTML 5 content in real time and via proxy servers rather than on the iOS device itself.

5.5.3 Mechanism of response #3: Regroup

This mechanism involves the developer repurposing a blocked application until it satisfies Apples criteria, and is subsequently allowed onto the App Store. In this way the developer capitulates to Apple's will, but ends up with their modified innovation on the App Store. The typical sequence of events in this mechanism is detailed in figure 5.9.

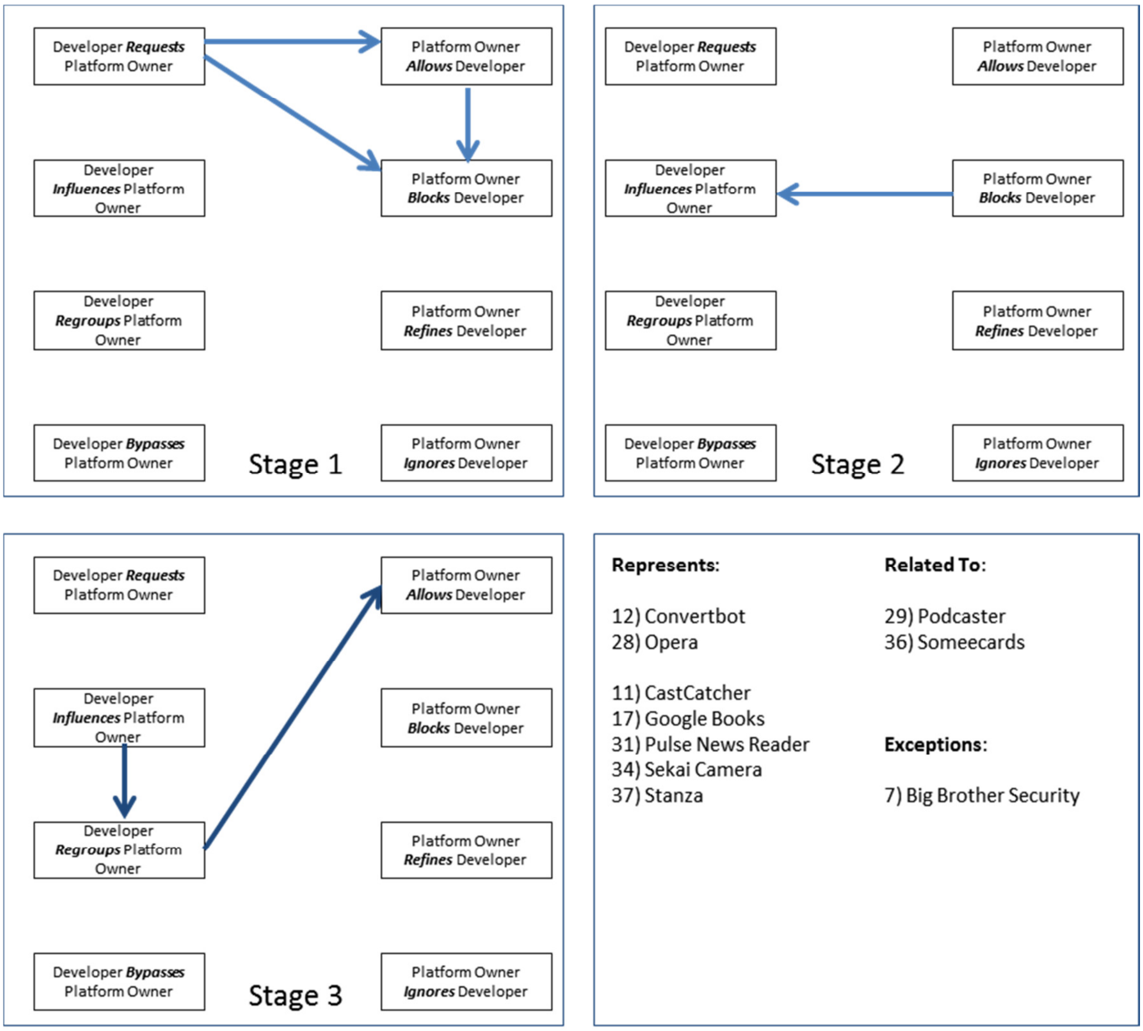

Developer *Requests* Platform Owner
Platform Owner *Allows* Developer
Developer *Influences* Platform Owner
Platform Owner *Blocks* Developer
Developer *Regroups* Platform Owner
Platform Owner *Refines* Developer
Developer *Bypasses* Platform Owner
Platform Owner *Ignores* Developer
Stage 1
Stage 2
Stage 3
Represents:
12) Convertbot
28) Opera
11) CastCatcher
17) Google Books
31) Pulse News Reader
34) Sekai Camera
37) Stanza
Related To:
29) Podcaster
36) Someecards
Exceptions:
7) Big Brother Security

Figure 5.9 Implementation of the third mechanism of response: Regrouping

This mechanism was followed in the sequence illustrated by Google Books (#17), which was initially blocked on account of Google not following Apple's new terms with regards to its in app purchasing policies. These new terms were introduced in 2011 for in app purchases such as book and magazine subscriptions for eBook reader apps. Following an attempted influence strategy, Google decided to adjust its apps and accept Apple's new terms. Google Books was duly readmitted back onto the App Store. Applications such as Podcaster (#29) and Someecards (#36) effectively followed a more elaborate version of the regroup strategy involving a larger sequence of actions.

Big Brother Security (#7) is listed as an exception because it attempted to regroup. The developer changed his application, after it was blocked, to remove the code that Apple objected to. However Apple simply ignored him, a de facto block, after he submitted the revised application.

5.5.4 Insights into Apple's control of innovation as a result of developer responses

A developer's application of any one of the three mechanism of response provides insights into Apple's control of platform innovation. Each mechanism is taken in turn, and the implications it holds for Apple's control of innovation discussed.

Successful influencing marks a loss of control

If a developer successfully uses an influencing mechanism, Apple will reverse its decision to block an app and admit it into the App Store. Two apparent reasons come to mind as to why Apple might reverse its blocking decision.

First, it might be persuaded that it incorrectly interpreted its own rule. This is exemplified by the story of Simply Beach (#35), in which the content of a beachwear retailers App was mistaken for soft porn, and blocked on the basis that it contained objectionable content. Once Apple has been alerted to its mistake, then it reversed its decision and allowed the app onto the App Store. Using the terms of formal control (Ouchi 1977; Ouchi 1978; Ouchi 1979), the specification of Apple's control was in order, however it made an error in terms of how it measured and evaluated whether control should be asserted. As a consequence of this type of error, the specification does not change, whilst the way that it is applied does.

Second, Apple might be persuaded that its rule is inappropriate. This was exemplified by the story of Adobe Developer Tools (#4). The use of these tools was banned on account of Apple enforcing its criteria prohibiting the use of cross compilers. This decision was fiercely contested both by Adobe and by many developers. Eventually Apple relented and changed its rules so that cross compilers were allowed. Using the terms of formal control once more, Apple's measurement and evaluation processes correct, however the specification itself was inappropriate. Consequently the specification changes, but the processes of measurement and evaluation remain intact.

Within the empirical data, the first error, concerning a misinterpretation of a rule, appears to happen when applying more subjective criteria, such as objectionable content specifications, to smaller apps. In addition to Simply Beach (#35), Tweetie (#41), Ninjawords (#27) and EFF Updates (#13) are examples of this type of error. The second error, concerning the changing of an inappropriate rule, appears to happen when applying more objective criteria to larger entities, such as whether a platform enabler like Google Admob(#2) is a competitive threat, or whether a VoIP app like Skype(#1) poses a threat to network partners.

In either case, Apple temporarily loses control of the boundary of permissible innovation, as the developer has forced it to be recast.

Successful bypassing marks an impasse

If a developer successfully uses a bypassing mechanism, Apple's App Store is bypassed for an alternative distribution channel. This was seen in the case of Google Voice (#18), which bypassed Apple by means of an HTML5 web app, and in the case of Nescaline (#25), which bypassed Apple by distributing its app from the Cydia store for jailbroken iPhones.

Since the developer's app does not appear on the App Store, Apple maintains control of innovation in its platform ecosystem. The boundary of permissible innovation remains unchanged as a result of a bypassing strategy.

However, Apple does not maintain control over alternative distribution channels outside of its platform ecosystem. Apple cannot control its iOS users from using these alternative channels to access an innovative app.

Consequently the successful application of a bypassing mechanism marks an impasse between the developer and Apple.

Successful regrouping marks a retention of control

If a developer successfully employs a regrouping mechanism, an app is eventually admitted into the App Store, albeit in a form which is acceptable to Apple. This was seen for example in the story of Google Books (#17).

Whilst the revised app might appear on the iOS as a platform innovation, Apple has retained control over the boundary of innovation, which remains unchanged.

Finally, the failure of a developer to use a regrouping mechanism, or any of the alternatives, indicates Apples continued retention of control over innovation in its platform ecosystem.

5.6 Examining the outcomes of contested platform innovation

Apple's attempts at applying control and the developers' attempts at resisting control have been analysed across all 45 instances of contested platform innovation. A final analysis is now carried out with respect to the overall outcomes of these tussles between Apple and its developers. This analysis attempts to understand the outcomes of these instances of contested innovation based on two factors. The first is based on outcomes according to the types of criteria, or five motivations, for control that Apple employs. The second concerns the outcomes according to the type of strategy that the developers employ to resist control. The purpose of this final analysis of results is twofold. First, it seeks to identify relationships between Apple's motivation for control, the developers' responses to control, and the outcomes of the 45 instances of tension in terms of whether platform innovation is achieved. Second, it is intended to generate further insights into the dynamics of the control of platform innovation.

The analysis is broken down into two parts. The first part simply summarises and presents the outcomes of this tension in tabular form. The second part is concerned with the detailed analysis of these results.

5.6.1 Presenting the outcomes of contested platform innovation

Table 5.6 lists outcomes by each of Apple's motivations for applying control and the strategies used by developers in response. Outcomes fall into two categories, either an app is brought into the App Store as a platform innovation, or it is left out. These outcomes are listed for each of the 45 instances of contested innovation.

The table progresses analysis in two ways. First, it helps identify which motivations, responses and combinations of motivations and responses were used more frequently than others, as well as to enable comparison of their respective outcomes. Second, it directs an investigation into the relationships between certain motivations and responses. The aim is not to provide a statistical basis for predicting or inferring. The objective is simply to guide attention to what appear to be those relationships that occur most frequently.

Motivation	Narrative	Reason for Control	Developer Response	Final Outcome
Competing Platforms	4 Adobe developer tools	Cross-Compilers	Influence	In App Store
	10 C64 Emulator	Platform Emulators	Influence	In App Store
	2 Admob	Alternative Advertising Platforms	Influence	In App Store
	17 Google Books	Alternative Payment Platforms	Regroup	In App Store
	3 Adobe Flash	Executable Code from other Platforms	Bypass	In App Store
	22 Jailbreakme	Jailbreaking	Regroup	Not in App Store
	25 Nescaline	Platform Emulators	Bypass	Not in App Store
	16 Financial Times	Alternative Payment Platforms	Bypass	Not in App Store
	32 Readability	Alternative Payment Platforms	Bypass	Not in App Store
Legal Action	33 Routesy	Threat to Apple's Legal Position	Influence	In App Store
	31 Pulse News Reader	Threat to Apple's Legal Position	Regroup	In App Store
	7 Big Brother Security	Data Collection and Privacy Abuse	Regroup	Not in App Store
Brand Image	5 Ari David	Objectionable Content	Influence	In App Store
	8 Bobble Rep	Objectionable Content	Influence	In App Store
	13 EFF Updates App	Objectionable Content	Influence	In App Store
	14 Eucalyptus	Objectionable Content	Influence	In App Store
	23 Mark Fiore	Objectionable Content	Influence	In App Store
	27 NinjaWords	Objectionable Content	Influence	In App Store
	30 Pull My Finger	Objectionable Content	Influence	In App Store
	35 Simply Beach	Objectionable Content	Influence	In App Store
	41 Tweetie	Objectionable Content	Influence	In App Store
	43 Wallpaper Gallery	Objectionable Content	Influence	In App Store
	28 Opera	Duplication of Functionality	Regroup	In App Store
	29 Podcaster	Duplication of Functionality	Regroup	In App Store
	12 Convertbot	Inappropriate Look and Feel	Regroup	In App Store
	36 Someecards	Objectionable Content	Regroup	In App Store
	34 Sekai Camera	Use of Private APIs	Regroup	In App Store
	37 Stanza	Use of Private APIs	Regroup	In App Store
	18 Google Voice	Duplication of Functionality	Bypass	In App Store
	19 GV Mobile	Duplication of Functionality	Bypass	In App Store
	6 Baby Shaker	Objectionable Content	Influence	Not In App Store
	20 Hottest Girls	Objectionable Content	Influence	Not In App Store
	44 Wallpaper Universe	Objectionable Content	Influence	Not In App Store
	21 I Am Rich	Dubious Value	Influence	Not In App Store
	42 VoiceCentral	Duplication of Functionality	Bypass	Not In App Store
	24 Me So Holy	Objectionable Content	Bypass	Not In App Store
	38 Tawkon	Use of Private APIs	Bypass	Not In App Store
	45 Wi-Fi Sync	Use of Private APIs	Bypass	Not In App Store
Network Partners	15 EyeTV	Excessive Cellular Data Usage	Influence	In App Store
	1 3G Skype	Enabling VoIP over 3G	Influence	In App Store
	11 CastCatcher	Excessive Cellular Data Usage	Regroup	In App Store
	26 Netshare	Excessive Cellular Data Usage	Influence	Not In App Store
Unknown	9 Box Office	Unknown	Influence	In App Store
	39 TrapCall	Unknown	Influence	In App Store
	40 Trillian	Unknown	Influence	In App Store

Table 5.6 Listing outcomes by motivations for control and responses to control

5.6.2 Analysing the outcomes of contested platform innovation

It paid to resist control, but the boundaries of innovation rarely changed

Of the 45 instances of contested innovation, 31 were resolved such that the developers’ app eventually entered Apple's platform ecosystem as an innovation, compared to 14 that did not. One could conclude that it pays for developers to resist Apple's controlling actions in cases of contested innovation.

However, an analysis of outcomes by the type of strategy developers employed to respond to Apple produces a more nuanced interpretation. In the majority of stories, 24 of the 45 instances, developers chose simply to attempt to influence Apple. Of these, 19 of the 24 instances were successful, Apple ceded control of innovation in these instances, and the boundaries of permissible innovation were shifted. 10 of the 45 instances resulted developers following a bypassing strategy. Of these, 3 of the 10 instances, were eventually permitted into Apple platform ecosystem unchanged, control of innovation was lost to the developers in these cases, and the boundary of permissible innovation moved. The rest, 7 of the 10 instances, remained being distributed through alternative platform ecosystems. Finally, in 11 of the 45 instances of contested innovation developers chose to regroup, and adjust their apps according to Apple's specifications. Of these, 9 of the 11 instances were accepted into the App Store, and 2 were rejected. Apple retained control of the control of platform innovation in all of these 11 regrouping instances, and the boundary of permissible innovation was unmoved. When looking across the outcomes of these stories according to the developers' responding strategies, it can be seen that Apple lost control of platform innovation on 22 occasions. This meant that the boundary of permissible innovation was moved on just less than half of the 45 instances of contested innovation.

A further analysis, considering outcomes by category of Apple's motivation for control, reveals that Apple ended up admitting only 7 apps after changing its specifications. This is relevant because a change in specifications is a better measure of a shift in the boundaries of permissible innovation. This relates back to the debate concerning whether control is mistakenly applied because of an error in the functions of measurement and evaluation of apps against specifications, or because the specification itself is fundamentally flawed. The latter is a mistake in the formal boundary of permissible control; the former is a mistake in how the boundary of permissible control is applied. The majority of instances of contested innovation occurred after Apple applied control because it perceived that a new app was threatening its brand image. This occurred on 26 of 45 occasions. Of these, 18 of the 26 instances ended up within the App Store, and of the 18, 12 of them remained in their original form. Of these 12 instances, only 3 of the contested apps were readmitted following a change in Apple's specifications (allowing the work of professional satirists). The remaining 9 caused by mistakes in the way Apple measured and evaluated apps which were initially thought to be objectionable content. The next most significant motivation for control, in terms of instances of contested innovation, is that concerning competing platforms. This occurred on 9 of 45 occasions. Of these, 5 of the 9 instances ended up in the App Store, and 3 of them remained in their original form. All 3 of these contested apps were readmitted following changes in Apple's specifications concerning, cross compilers, platform emulators, and alternative advertising platforms. There were 4 instances of control being applied motivated by Apple's need to protect its network partners. Of these, 3 out of the 4 ended up in the App Store, 2 of them remained in their original form. Only 1 of these 2 was readmitted on account of Apple changing its specifications, in this case allowing Voice over IP (VoIP) on 3G networks. The remaining two motivations for control, concerning protection against legal action and control for unknown reasons yielded a total of 6 of instances of control between them. 5 of these contested innovations were eventually allowed into the App Store, 4 of them unchanged, and none led to a change in Apple's specifications. When the instances are analysed of apps being admitted into the App Store as a result of Apple changing its specifications for control, a more robust measure of the boundary of permissible innovation, a more conservative figure is arrived at. In total only 7 of 45 apps are admitted into Apple's platform ecosystem as a result of the boundaries of permissible innovation shifting, when measured as a change in specifications for formal control.

In conclusion, 31 of the 45 instances of contested innovation ended up with applications being admitted, in some form, into Apple's platform ecosystem. 9 of these apps were admitted after they were redesigned to comply with Apple's specifications. 15 of these apps were admitted after Apple reversed a mistake that it had made in terms of how it decided to apply control. Finally 7 of these 45 instances were allowed into Apple's platform ecosystem after it had changed the way a specification of control, which truly marks a shift in the boundaries of permissible innovation. The distinction between the final two groups is the way Apple decides to apply specifications for control verses what specifications for control Apple decides to apply.

A possible logic behind developers' strategies responding to apple's control

From the flow of the stories, it appeared that the first step in responding to Apple's control was to attempt to influence Apple to change its mind. This strategy appeared more successful when attempting to influence Apple that it had made a mistake in the way that it applied a specification for control. 15 of the 22 apps that were admitted unchanged were as a result of Apple reversing an error in the way that it had applied control. This was most frequently observed when Apple was attempting to protect its brand image. For example 7 instances involving the subjective criteria of objectionable content were overturned successfully in this way. In contrast, it appeared harder to influence Apple when its specification for control was considered to be inappropriate. Only 7 of the 22 apps that were eventually admitted unchanged were as a result of a change in specification. Of the 5 apps that remained rejected following an influencing strategy (and where no alternative strategy followed), it was quite clear that they fell afoul of Apple's specifications. Furthermore it was clear that Apple was unlikely to change its rules in these cases. For example, Netshare (#26) was a "tethering app" which allowed users to enable internet access from their PCs via an iPhones cellular connection. Apple has a strict policy of banning uncontrolled "tethering apps", which is unlikely to change until it is sanctioned to do so by network operators. Last, it must also be noted that an influencing strategy was predominant in stories (Box Office #9, Trapcall #39, Trillian #40) where Apple's reasons for control were unknown. In each case, Apple ignored the developer’s request for an app to be accepted as an innovation. It is possible that Apple's decision making process had broken down, because, after influencing, all 3 apps were accepted.

Following the failure of an influencing strategy, the stories indicate that a developer has to decide whether to adopt a bypassing strategy or use a regrouping strategy. The key criteria for a developer to consider is whether they are determined to have their application in the hands of iOS users unchanged but not benefit from the reach of the iOS platform ecosystem, or if they are able and prepared to compromise their app so that it meets Apple's specifications. If the developer prefers the former then a bypassing strategy is a likely course of action. If a developer prefers the latter then a regrouping strategy would seem reasonable.

A regrouping strategy seemed to be prominent in absolute terms (6 instances) when Apple applied control motivated by protecting its brand image, and in relative terms (67% of instances) when motivated by protecting itself from legal action. In the case of the former it was relatively straightforward for developers to adjust code: 1) to remove duplication of functionality (Podcaster #29); 2) adjust the apps look and feel (Convertbot #12); 3) remove objectionable content (Someecards #36); and 4) replace the use of private APIs (Sekai Camera #34). In the case of the latter, when faced with legal action, developers adjust their apps accordingly. This was successful in the case of Pulse News Reader (#31), which was readmitted into the App Store, but not in the case of Big Brother Security (#7).

A number of implications were borne out from the stories when developers chose bypassing strategies. For example, it appeared that apps distributed via alternative app stores, like Cydia, and which can only be accessed by jailbroken iOS devices, did not reappear on Apple's App Store at a later date. Within the corpus of data, none of the 6 apps which ended up on Cydia were later readmitted into the App Store. In contrast, stories that included bypassing strategies involving HTML5 Web Apps, such as Google Voice (#18) and GV Mobile (#19), were later readmitted into the App Store. A possible explanation for this is that Apple strictly forbids jailbreaking, and is unlikely to readmit to its platform ecosystem those developers who flout this rule.

The most prominent use, in relative terms, of bypassing occurred when Apple applied control in order to protect itself against competing platforms. Nearly half of developers whose apps were controlled in this way chose to bypass Apple's platform ecosystem. The developers of the Financial Times (#16) and Readability (#32) bypassed Apple because they did not want to commit to the terms that Apple introduced regarding the use of subscription payment platforms. The developers of Nescaline (#25) circumvented Apple because of its ban on platform emulators. Whilst Adobe did not directly bypass Apple to enable the use of Flash content on iOS, Apple was indirectly bypassed by the developers of Skyfire who enable Flash content via a proxy server.

It would therefore appear that the decisions that developers make when choosing a response to Apple's control can be explained logically.

The boundaries of innovation can change, but slowly and after much effort

Apple's motivation to protect itself against competing platforms appears to be the most fiercely contested boundary of permissible innovation with Apple's iOS platform. Across the 45 stories, it represents the second most frequently used set of specifications concerning the control of innovation. 9 instances of control occur here. Of all the motivations for control that are applied, it results in the lowest proportion of innovations entering Apple's ecosystem. All three strategies are employed by developers in response to control, and the distribution of the use of these strategies is more even than within other motivations. Closer analysis of stories that fall within these controls yields insights into the dynamics of the control of platform innovation.

The reason for Apple to resist competing platforms functioning on iOS can be explained by a commercial desire to retain a control over access to and ownership of its users. The risk is that by allowing other platforms access to iOS users, they might lead to the poaching of users from Apple, or exposing them to an inferior user experience which might damage Apple's brand image by implication. Consequently this motivation for control has embodied a range of specifications guarding against: 1) the functioning of executable code from other platforms; 2) the allowing of cross (platform) compilers; 3) the sanctioning of platform emulators; 4) the permitting of jailbreaking; and 5) the tolerance of alternative advertising and payment platforms.

This dissertation examines platform innovation in terms of apps and platform enablers. All of the stories concerning platform enablers, for example Adobe Flash (#3), fall within this this particular motivation. Platform enablers allow the operation of the core business of these developers, which explains why they are determined to gains access to iOS, as Apple is determined to prevent them.

Furthermore the tension that developers and Apple are engaged in centres on what specifications for control Apple decides to apply, rather than how Apple decides to apply specifications. In each and every story that falls under this motivation for control, it is the true boundary of permissible innovation that is being contested as it is the nature of the specifications that is being debated. Of the 6 occasions, across the 45 stories, where tension led to Apple changing its specifications, 4 occurred within this broad class of control. The specifications regarding: 1) cross-compilers changed thus allowing Adobe Developer Tools (#4); 2) platform emulators changed thus allowing C64 Emulator (#10) but on the condition that executable code for other platform was not to be downloaded; 3) alternative advertising platforms were modified thus allowing Google Admob (#2); and 4) alternative payment platforms were changed, after Google Books (#17) had complied with the previous specification, and after the Financial Times (#16) and Readability (#32) leaving the platform ecosystem.

Following on from this, two key observations can be made from the empirical data concerning these particular stories. First, it would appear that Apple is prepared to compromise or relent with regards to specifications that protect its business. However, it takes time and considerable developer effort for these changes to happen, in contrast to the examples where Apple made errors in how it applied specifications for control. It took between three and twelve months and considerable support from the wider developer community for each of the specifications regarding competing platforms to change. The time and effort required to change control, may explain why numerous developers chose to bypass Apple's platform ecosystem. Secondly, there are specifications for control which Apple does not appear to be willing to change. These criteria concern jailbreaking and enabling code from other platforms to be accessible and executable on iOS devices. Around these two criteria, the boundaries of permissible innovation appear to be rigid.

5.7 Summary of analytical findings.

The chapter sought to describe and explain the dynamics of the control of innovation in Apple's iOS platform ecosystem. It was therefore divided into two parts. The first part concerns the description of the dynamics of the control of innovation in terms of sequences of action that occur when innovation is contested. The second part concerns an explanation of the dynamics of the control of innovation in terms of two opposing sets of mechanisms.

In addressing the first part, 45 stories concerning contested innovation were found in the empirical data extracted from blog posts. Following this, a compact lexicon of eight actions describing developers’ attempts at platform innovation and Apple's attempts to control the innovation of platform complements were identified using semiotic squares. These actions were used to code the stories into a set of 45 sequences of action concerning contested innovation. On the basis of this, it was possible to construct individual narrative networks (Pentland and Feldman 2007) describing each narrative, and then to combine them into one single narrative network. By abstracting the combined narrative network and focussing on the most frequently occurring patterned sequences of action it was possible to identify a generalised sequence of actions that was broadly common across the stories. This generalised sequence of actions could be split into three distinct stages, which could then be used to describe the dynamics of the control of innovation. The first stage represents a build-up to tension between Apple and a developer and concludes in Apple blocking a developer's request for an innovation. The second stage represents a tension between the two parties and describes the developer attempting to influence Apple to reverse its blocking decision. The third and final stage represents a resolution to the tension, and identifies three possible outcomes: 1) Apple may respond to attempts to influence it positively, and allow the innovation onto its platform, possibly refining its rules at the same time; 2) The developer may decide to bypass the Apple's App Store, Platform Ecosystem and Apple's control by seeking to distribute its unchanged innovation to iOS devices through some other channel; 3) The developer may comply with Apple's reasons for blocking the innovation, and regroup and rework the app until it is compliant with Apple's rules.

The second part is concerned with explaining the observed sequences of contested platform innovation in terms of two opposing sets of mechanisms.

First, elements of Apple's mechanism for controlling innovation, the App Approval Process, were analysed using formal organisational control (Ouchi 1977; Ouchi 1978; Ouchi 1979) as a theoretical lens. It was found that Apple favoured specifying sanctions rather than rewards, and that these were focussed on outcomes of the process of innovation rather than on how innovation was attempted. It was also reasoned that Apple specified the control of innovation in these ways as the outcomes and ways of carrying out new innovation are often unknown. It is therefore more conducive for the generation of a broad range of innovation, as well as being easier to control, to state what is not wanted rather than what is. Finally, an investigation into Apple's specifications for control revealed that they could be influenced and that they can change over time.

Second, the means by which developers responded to Apple's control were analysed. It was found that the developers used three mechanisms: influencing; bypassing; and regrouping. It was found that there were occasions when Apple could be influenced to reverse a block. The basis of this reversal was founded on persuading Apple that it had made an error in one of two ways. Either it had made a mistake in how it applied a specification for control, or, alternatively, the error lay in the meaning of the specification itself. Either way, the boundary of permissible innovation had in some way been affected. Furthermore, it was discovered that the successful application of the bypassing mechanism marked an impasse between Apple and a developer. On the one hand a developer succeeded in bypassing Apple's iOS platform ecosystem by distributing an innovation to iOS users through an alternative channel. On the other, Apple retained control over what can be innovated in the platform ecosystem as a result of the implementation of a bypassing strategy. In this instance, the boundary of permissible innovation remains the same. Finally it was considered that the successful implementation of a regrouping mechanism represented Apple's retention of control over platform innovation. By regrouping a developer accepts to redesign a blocked app in order to meet Apple's specifications. Consequently Apple retains control over an unchanged boundary of innovation.

Last of all, a number of additional findings were arrived at in attempting to identify the relationship between: 1) the mechanisms of control; 2) the mechanisms of response to control; and 3) the outcome of contested innovation. First of all, it paid for the developers of the platform complements featured in the stories to resist Apple's control. 31 of the 45 instances of contested innovation concluded with applications being admitted, in some form, into Apple's platform ecosystem. However, only 7 of these were admitted after Apple had changed its formal specifications for control, which is representative of a change in the boundaries of innovation. Changes in the boundaries of innovation were infrequent. Second, a possible logic explaining developers' use of mechanisms to respond to Apple's control was found. The developers first sought to influence Apple to reverse its blocking decision. Should this mechanism fail to get Apple to reverse its decision, developers were faced with a choice of bypassing or regrouping. It was suggested from the data that developers bypass Apple if they were determined that an innovation should reach end users unchanged, but were prepared to forego the use of the iOS platform ecosystem as a distribution mechanism to end users. The data also indicated that developers regrouped if it was essential for their innovations to be distributed from Apple's iOS platform ecosystem and they were prepared to compromise the design of an innovation to meet Apple's specifications. Finally, it was concluded that when Apple was prepared to change the boundaries of permissible innovation, it was done slowly and after considerable effort on the part of large communities of developers.

6 Discussion

The purpose of this chapter is to present and then discuss a theory of contested innovation on curated digital platforms. It starts off by taking the key findings of the results and analysis chapter, and then framing them as a means of describing and explaining contested platform innovation based on examples of new platform complements allied to Apple's iOS platform. This theory is then discussed in depth. First the nature of control in digital platforms is examined theoretically, in terms of how control is enforced by platform owners and resisted by developers of platform complements. Next, the critical role of digitalisation in the control of digital platform innovation is analysed both in terms of how it dissolves the control that platform owners attempt to enforce in some ways, and yet enhances control in others. This is then followed by a discussion, which attempts to unravel a paradox of control and generativity in digital platform innovation by examining the relationships between developer and platform owner, as well as between control and generativity. This leads to discussions as to what forms of innovation come about, and who is really in control of platform innovation. Finally, this discussion ends by exploring the possibility of extending the theory to innovation within other platform contexts.

6.1 Summary of a theory of contested innovation

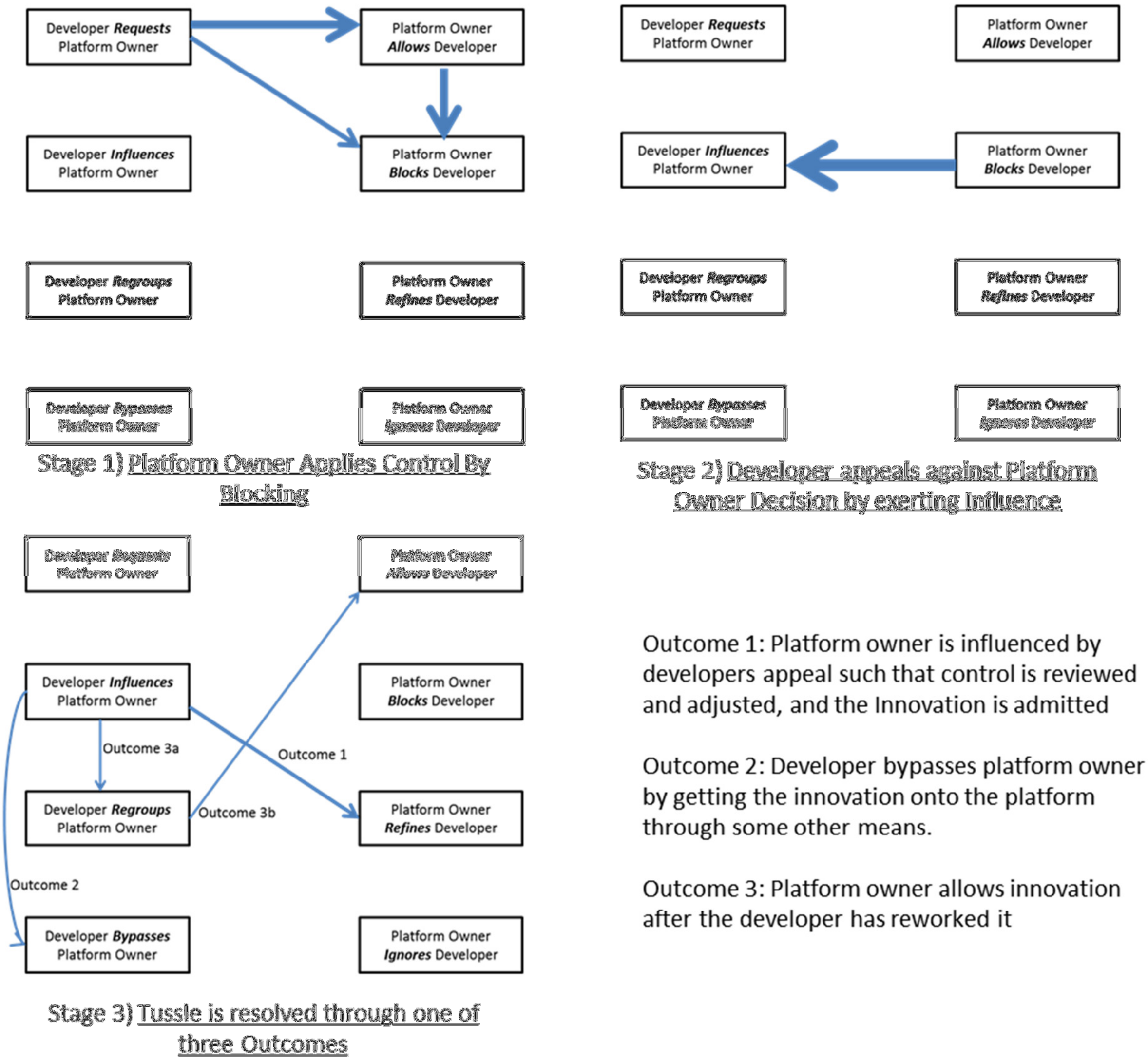
The objective of this thesis is to develop a theory to describe and explain the nature of the innovation of platform complements on digital platforms. A tentative theory is now suggested that achieves this objective, but it is initially bounded by certain constraints. First, the theory focusses on instances of contested innovation. In these cases developers' attempts at innovation of platform complements, such as apps and platform enablers, are blocked by the platform owner. Developers contest this control and attempt to overcome the block. Second, the theory is constructed based on empirical data describing instances of contested innovation on Apple's iOS platform, which is a curated platform. Last, the empirical data describes events concerning contested innovation occurring between 2007 and 2011 inclusive. The theory is a low level process theory designed first to describe and then to explain events within these bounds. It initially makes no claim to describe or explain events outside of these bounds, nor does it attempt to predict what might occur in future instances of contested innovation. An overview of the theory's two constituent parts is now presented.

6.1.1 Description

This description of how innovation of complements occurs within a digital platform is divided into two further parts, a semantic part and a syntactic part. The semantic part suggests two compact sets of actions, one employed by a developer, and the other by a platform owner, in instances of contested innovation. Each set consists of four actions, and both sets of actions are illustrated and described in Table 6.1. The developer and the platform owner employ these actions one at a time in sequence through their interactions concerning whether a development is to be fulfilled as an innovation on Apple's iOS platform. The second part of the descriptive element of the theory concerns the syntax of these actions. The sequencing of the developer and platform owner actions follows one of three different types of pattern as they interact. These three basic patterns of sequences of actions are summarised as narrative networks in figure 6.1. Each of these patterned sequences of actions has three stages: a build-up to tension as the platform owner rejects an innovation; the expression of tension, as the developer subsequently attempts to influence the platform owner; and resolution, as the app or platform enabler is either readmitted or remains outside of the ecosystem. It must be noted that combinations of these basic sequences are also permitted.

Developer (Actions Progressing Innovation)	***Platform Owner*** (Actions Controlling Innovation)
Request	Allow
Bypass	Block
Influence	Refine
Regroup	Ignore

Table 6.1 Table of actions describing the progression and control of innovation


Developer Requests Platform Owner
Platform Owner Allows Developer
Developer Influences Platform Owner
Platform Owner Blocks Developer
Developer Regroups Platform Owner
Platform Owner Refines Developer
Developer Bypasses Platform Owner
Platform Owner Ignores Developer
Stage 1) Platform Owner Applies Control By Blocking
Stage 2) Developer appeals against Platform Owner Decision by exerting Influence
Outcome 3a
Outcome 1
Outcome 3b
Outcome 2
Stage 3) Tussle is resolved through one of three Outcomes

Outcome 1: Platform owner is influenced by developers appeal such that control is reviewed and adjusted, and the Innovation is admitted

Outcome 2: Developer bypasses platform owner by getting the innovation onto the platform through some other means.

Outcome 3: Platform owner allows innovation after the developer has reworked it

Figure 6.1 Patterned sequences of actions in contested platform innovation

6.1.2 Explanation

The second part of the theory attempts to provide an explanation for patterned sequences of actions that are observed. The theory proposes that instances of contested innovation are a dynamic process, which is driven by opposing motivations on the sides of the developer and on the side of the platform owner. On the one hand, the developer is driven by the commercial need to get an app or platform enabler accepted and commercialised as an innovation. On the other hand, the platform owner is driven by the commercial need to control for appropriate third party innovation in order to protect the integrity of the platform. In instances of contest innovation these two motivations clash, and act as forces through two respective mechanisms.

Apple’s Apps Approval Process acts as a mechanism by which it attempts to control for appropriate platform complements to admit onto its platform. The mechanism is constructed from four main components which are illustrated in figure 6.2. The first component concerns Apple's specifications for control, which consist of the iOS Developer Program License and the App Store Review Guidelines. Apple typically specifies outcomes of innovation that will not be tolerated. These specifications are available to developers and inform them of what is, and is not, appropriate by means of what and how to innovate. The next two components of the control mechanism concern Apple's measurement and evaluation of the developers' code in terms of deciding whether it should be admitted into the platform ecosystem as an innovation. The evaluation is carried out against Apple's specifications of control. The final component of the mechanism of control consists of the reward or sanction that Apple metes out to the developer according to whether the code falls inside or outside of the set of specifications. If the code falls within the specifications then the developer is rewarded by having the code admitted into the platform ecosystem, generally as an App on the App Store. If the code falls outside of the specifications it is rejected by Apple.

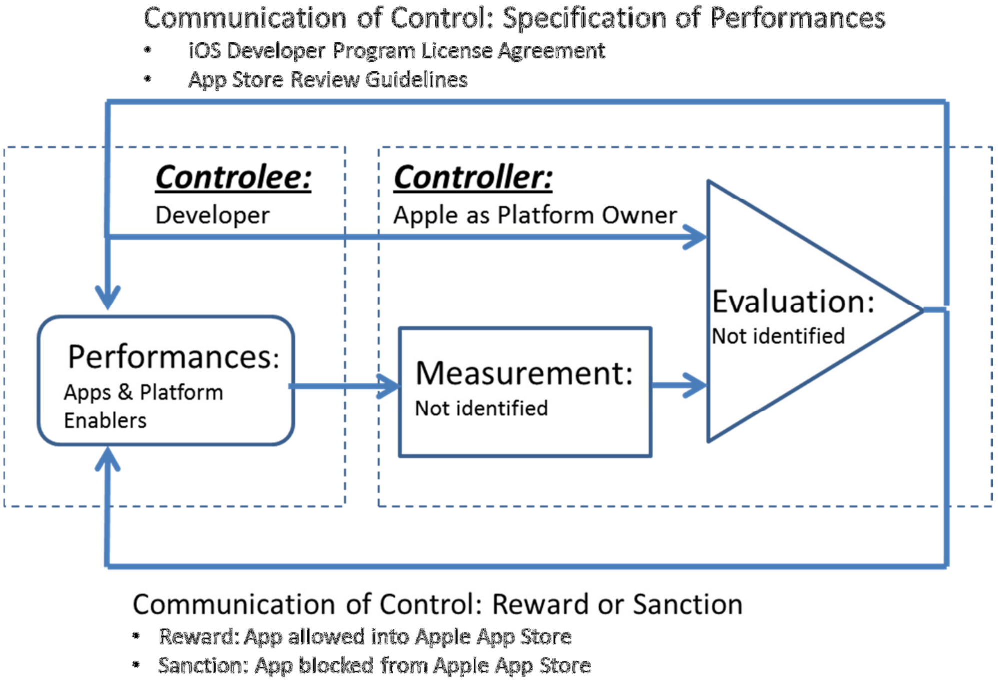

Communication of Control: Specification of Performances
• iOS Developer Program License Agreement
• App Store Review Guidelines
Controlee:
Developer
Controller:
Apple as Platform Owner
Evaluation:
Not identified
Performances:
Apps & Platform Enablers
Measurement:
Not identified
Communication of Control: Reward or Sanction
• Reward: App allowed into Apple App Store
• Sanction: App blocked from Apple App Store

Figure 6.2 Apple's apps approval process as a control mechanism

(After Ouchi 1978).

Should Apple, in its role as platform owner, decide to reject a platform complement as an innovation, the developer, motivated by their need to commercialise their code, may well respond to Apple's attempt at control. The proposed theory provides three mechanisms by which the developer can respond. These are identified in figure 6.1. The developer will first attempt to influence Apple in order to persuade them to reverse their decision. If this is unsuccessful the developer may then either try one of two actions. The developer may try to bypass Apple, by distributing there code as an innovation to iOS users through some channel other than Apple's official iOS ecosystem. Alternatively the developer may adjust the code so that it conforms to Apple's specification in the hope that Apple will allow it the second time around.

6.2 The nature of control in digital platform innovation

6.2.1 Enforcement

In the context of this study, control of a platform is the ability to regulate a platform and platform innovation at will. The objects of the platform owner's attempts at controlling innovation are platform complements, in the form of apps and platform enablers, and created by third party developers. Platform owners control these digital artefacts as their profits are tied to the overall economic value of their platform (Boudreau and Hagiu 2009). Consequently it can be in the interests of a platform owner like Apple to curate the content that is created for their platform in order to ensure that its entirety is as valuable as possible. This may entail subjective judgements with respect to what is in the best interests of the platform ecosystem, which may in turn be the basis of tensions between platform owner and complementors.

The digital platform, to which participants are contributing complements, are digital infrastructures (Tilson et al. 2010) and are subject to social technical forces, or modalities, which regulate their behaviours (Lessig 2006). Three out of four of Lessig's modalities of regulation are employed by Apple to enhance its power over other members of the iOS platform ecosystem. These three modalities concern: markets; architecture, which Lessig refers to as code; and law.

First, Apple currently owns a sizeable share of the smartphone market, partly as a result of its ability to harness network externalities (Katz 1994). Its command of this market leads it to significant power over developers who wish to address this market with their complements, as it acts as the gatekeeper between developers and users. Developers are therefore dependent on the platform owner for access to a key resource (Pfeffer and Salancik 2003) that is required for their businesses to function

Second, as owner of the platform, Apple is assumed to possess decision rights over the platform architecture (Baldwin and Woodard 2009). This enables it to design, build and manage architectural control points (Woodard 2008). These can act as bottlenecks (Jacobides et al. 2006) which provide the platform owner with technical control over other ecosystem members' access to unique and valuable platform resources and functionality. In the context of this research, the concept of an architectural control point is embodied in Apple's App Store, which is where developers' apps, if approved by Apple, are placed so that they can be purchased by end users. Apple has designed its platform in such a way that no other channel to market for native apps, for example an alternative app store, is available to complementors and end users without inconvenient modification (Jailbreaking) to iOS devices.

The last of Lessig's modalities of regulation that is of interest concerns that of law. Apple owns property rights over its platform which allows it to use contractual and legal instruments such as licenses to regulate access to its platform resources, hence the iOS Developer Program License Agreement. This provides it with the "power of exclusion", or "Bouncers' Rights" (Boudreau and Hagiu 2009; Strahilevitz 2006), associated with legal asset ownership. Apple is in its rights to prevent ecosystem members accessing architectural control points, such as the App Store, which it owns.

These three modalities of regulation conspire to provide Apple with the power required to exercise the mechanism of control that is the Apps Approval Process. This is a control mechanism largely based on outcome and behavioural control (Tiwana et al. 2010), whose sanction is exclusion from the App Store, and whose reward is inclusion. Whilst the empirical data does not provide much insight into the internal process and means by which Apple carries out measurement and evaluation of developers' code, it provides rich data with regards to the specifications of control.

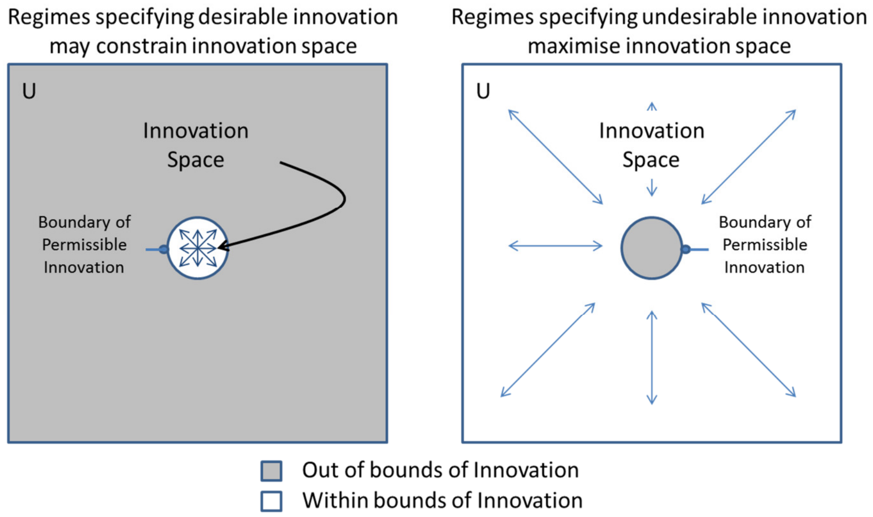
Apple's specifications for control are formally laid out in two documents, the iOS Developer Program License Agreement and the App Store Review Guidelines. As such, they lay out the boundary of permissible innovation within the iOS platform ecosystem. However, it is apparent from the following statement, that these boundaries are not fixed or fully determined:

"This is a living document, and new apps presenting new questions may result in new rules at any time. Perhaps your app will trigger this." (Apple 2011, p1)

When examining these specifications it becomes apparent that this process is concerned with the application of control once code is written and before it can be distributed to end users, rather than as the code is being written. Consequently Apple's focus appears to be more on the formal control of outcomes rather than of behaviours. Furthermore these rules seem more concerned with specifying undesirable outcomes and accompanying sanctions, rather than with desirable outcomes and appropriate rewards.

The reason that outcome control is specified rather than behavioural control might come down the fact that it is easier to manage. Behavioural control requires knowledge of a task programme, or how a task is to be completed (Eisenhardt 1985). In the case of innovating code this may be difficult to achieve. The very nature of innovation requires new outcomes, and possibly new methods to achieve them, which cannot necessarily be known in advance. This is confirmed by Svahn and Henfridsson (2012) in their examination of tensions between product innovation and IT innovation in digitalised artefacts. They implicitly identify processes of digital innovation, and the behaviours that drive them, as hard to define and know in advance, unlike more traditional forms of product innovation. IT innovation would apparently not be suited to behavioural control centred on centralised authoritarian surveillance.

This logic may also apply to the reason why the specifications focus on undesirable outcomes of innovation and the fact that they will be met with sanctions in the form of rejection from the App Store. The advantage of open innovation is that by outsourcing innovation, fresh ideas and concepts for complements, which have never been thought of by the platform owner, can be generated by third parties. In many instances the platform owner does not know what kind of third party innovation it wants. It is therefore suggested that in the control of innovation and new ideas, it is therefore easier to control for what is not wanted, which is possibly easier to know, than for what is wanted. Furthermore, as the diagrams in figure 6.3 illustrate, generative capacity, is much greater if developers are kept outside of areas of undesirable innovative potential delineated by a boundary of permissible innovation, rather than inside limited areas of innovative potential.



Regimes specifying desirable innovation may constrain innovation space
Regimes specifying undesirable innovation maximise innovation space
U
Innovation Space
Boundary of Permissible Innovation
U
Innovation Space
Boundary of Permissible Innovation
Out of bounds of Innovation
Within bounds of Innovation

Figure 6.3 Relationship between generative capacity and regimes of control

(Own Figure)

6.2.2 Resistance

The empirical data shows Apple's control being resisted by developers, who may attempt to influence Apple's decision. If unsuccessful in this endeavour they may resist further by bypassing, or they may capitulate by redeveloping their code in an attempt to make it acceptable to Apple.

The results show that on occasion Apple can be influenced to reverse its blocking decision. This seems to occur in two ways. First, Apple may be convinced that it made an error with regards to how it used a specification for control. The empirical data indicated that these were often errors of interpretation in terms of how a specification for control was interpreted when measuring and evaluating developer code. On these occasions, the code in question is subsequently allowed onto the platform as an innovation, and Apple may adjust the way in interprets its specifications. Second, Apple may be persuaded that it made an error with respect to what specification is defined for control. If Apple can be persuaded that a specification is inappropriate, then the decisions made based on this specification are also inappropriate. On these occasions, all instances of the code in question may subsequently be allowed into the ecosystem as innovations, and Apple may actually rewrite its specification.

The results also indicate that Apple's control over its developers can be subverted. It was seen that developers, on failing to influence Apple, were able to bypass and distribute their code to iOS users through alternative channels to Apple's App Store.

These actions show that Apple's attempts at controlling the nature and boundary of innovation can be contested, subverted, overturned and even changed by developers.

Consequently, the boundary of permissible innovation seems not to be fixed, it can be flexible. What is and is not permitted as a platform complement within Apple iOS ecosystem can change. A developer's code may at first be allowed as an innovation, then it may be disallowed, Apple’s decision may then be contested before its control is subverted, overturned and its specification ultimately changed. Some innovation of platform complements within the ecosystem would therefore seem to come about as a result of a dynamic process which results from the tension between the two parties. Furthermore, it would also appear that Apple might not be fully in control of the process concerning the innovation of platform complements on its platform.

6.3 Digitalisation and the control of digital platform innovation

Just as the digital nature of platforms was seen to facilitate their generative capacity for innovation, it is worth considering how the characteristics of digitalisation may affect the control of innovation of platform complements. The following paragraphs examine how the digital nature of platforms has the potential for both dissolving the control of innovation in some ways as well as for precipitating it in others.

6.3.1 Dissolving control through digitalisation

The results show that the developer can effectively bypass Apple's control by distributing a blocked app to consumers through other channels such as the internet for HTML5 based web apps or through alternative app stores, like Cydia, for native apps. As will be illustrated, Apple enforces this control on both apps producers and consumers, through its assumed ownership of the platform design rules (Baldwin and Clark 2000). However the following illustration also shows how the layered modular architecture (Yoo et al. 2010a) of the platform provides developers with the means to circumvent this control.

In order that a new app written in native code can be run on an iOS device, the associated code must first be placed on the device and then the code must be installed. In practice an application comes as several files, which must be installed and set up on the iOS device correctly so that functionality of the app will execute properly. This process is known as unpackaging. The iPhone operating system (iOS) is constructed such that it automatically handles the process of downloading an application, unpackaging the application, installing and setting up the files. However it is setup by default such that this process can only function with files downloaded over an internet connection directly from Apple's App Store. This is an additional mechanism of control, which Apple uses to ensure that only Apps approved by Apple through the Apps Approval Process can run on the iPhone. Apple does not grant the user permission to change these settings, so that under normal circumstances the user has no choice other than to use the App Store as the sole source of complementary functionality. Put simply, alternative sources of iOS Apps are not available to an unmodified iOS device such as an iPhone.

With respect to enabling the consumption of apps that are not officially approved by Apple, both the supplier and consumer of the app must effectively bypass Apple's App Store. With respect to apps written in iOS native code, the developer bypassed the App Store by placing the code on an alternative app store such as Cydia.

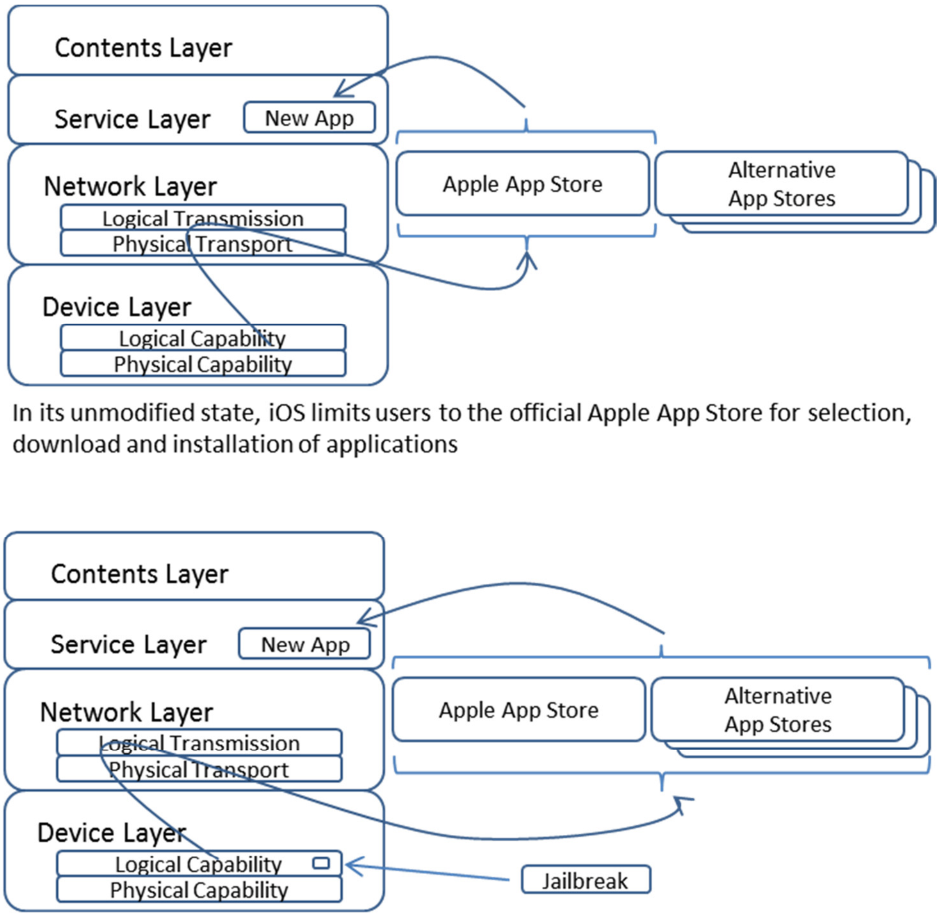
However, as was described above, users of unmodified iOS devices are constrained by Apple to accessing new apps from the official Apple App Store alone. In order to access unofficial iOS apps, users must bypass Apple's control as well. They do this through the practice of "jailbreaking" whereby they hack or modify the kernel, or the main component of their device’s operating system. Through "jailbreaking" the iOS kernel is modified such that the device is free to access alternative sources for applications and, most importantly, is able to unpack and install these unofficial applications. "Jailbreaking" is enabled by small portions of written code are able to breach an iOS device's closed and secured run time environment by exploiting security holes in the operating system. For example a jailbeaking method, based around a package of code known as "JailbreakMe 2.0" (Story #22 in this thesis), was able to exploit a security gap in Safari, the stock iOS web browser. The code was disguised within Adobe PDF file, which, when accessed by the iOS browser, was executes when accessed the PDF viewer within Safari and then opens iOS for further third party modification. Amongst these additional modifications is the ability to access, unpackage and install unofficial apps from unofficial app stores such as Cydia. These unofficial applications, unpoliced by Apple, enable modifications to the devices look, feel and operation that would most likely be prohibited on the official store. As story #22 concerning Jailbreakme makes explicit, Apple generally patches these security holes, preventing further jailbreak attempts using the same mechanism. However this does not prevent new jailbreaking methods, which find new security holes, to be developed.

Apple's apparent loss of control over the distribution of apps can be explained in terms of the characteristics of digital artefacts and the process of digitalisation.

First, Apple is able to impose control as a result of its assumed governance of the platform design rules (Baldwin and Clark 2000). The functionality in iOS which limits users to Apple's official App Store is effectively an additional architectural control point (Woodard 2008) deliberately constructed by Apple. Users must accept the inclusion of this architectural control point on purchase of the device, although it is not illegal for them to modify the device operating system to bypass it.

Second, Apple lose outright control over its ability to dictate which distribution channels are open to developers and consumers, due to the fact that the core component of its platform can be modified and reprogrammed by third parties. Apple has designed its system architecture such that it is a partially closed platform. It is only open to the degree that complements, in the form of apps developed by third parties, are allowed if Apple judges them appropriate. Other aspects of the platform are closed. For example, the architectural component that forms the App Store is tightly coupled to the operating system, which both form key components of the platform. No alternative app stores are permitted, or possible without unauthorised modifications to the platform. The iOS platform, and the components around it, can also be more generally viewed as a specific form of layered architecture (Benkler 2006). In totality it consists of a device running an operating system, which is connected to the internet, and runs services or applications that can manipulate information as content. In its unadulterated form, the architecture is not a complete layered modular architecture (Yoo et al. 2010a) as Apple closes certain possible combinations between components. As figure 6.4 indicates, it is a layered and partially modular architecture. However, with digital nature of iOS is such that it can be reprogrammed (Yoo et al. 2010a) and modified (Kallinikos et al.

forthcoming). In fact it can modified so that its tight couplings with proprietorial components like the official App Store can be loosened, enabling other components or modules in the form of alternative app stores to be substituted. As control between these layers and components are dissolved, the architecture becomes more open and closer to a true layered modular architecture (Yoo et al. 2010a).



In its unmodified state, iOS limits users to the official Apple App Store for selection, download and installation of applications

When the logical capability of the device is modified with additional code, such that iOS is "jailbroken", it becomes possible for the user to access a variety of App Store for the downloading and installation of apps. The user is no longer constrained to Apples curated choice of apps.

Figure 6.4 The dissolving effect of a layered modular architecture on control.

(After Benkler 2006)

Physical artefacts can of course be modified too. It is perfectly possible to "pimp" an automobile to the degree that its engine may be modified or even replaced. However the nature of digital artefacts is such that profound modifications such as jailbreaking are simple, free, require no additional equipment and are rapidly communicated and replicated.

Furthermore, the loose binding between the architectural layers is such that many alternative forms of bypassing are available. The example illustrated above details the case of blocked apps being distributed as native applications. However, as in the story of the Financial Times (Story #16) shows, developers can bypass Apple's control by converting their ideas into web apps. These applications are accessed and executed independently of the constraints that Apple's iOS places on native applications, although they are subject to other constraints. The alternative means that developers have of affecting Apple's control of innovation is an influencing strategy. This was seen to be effective when Apple could be persuaded that it had either misinterpreted its own specifications for control, or that its specifications were inappropriately defined. Whilst this strategy or response to control may not be directly enabled by the digital characteristics of the iOS platform, it is possible that there are indirect effects. Apple has power over developers, as these third parties are dependent on the platform owner for access to key resources (Pfeffer and Salancik 2003) such as consumers. However Apple is also dependent on developers for high quality platform complements in the form of apps to attract consumers. The power dynamics between the two parties are asymmetric as a result of the one to many relationships between Apple and its developers. Whilst this may dilute the power of individual third parties with respect to influencing Apple, some third parties are more powerful than others. The power of individual third parties may come down to a number of factors including their relative size and importance to consumers. The bargaining power of these developers, as they try to influence Apple, may be enhanced by the fact that they can bypass Apple and reach its consumers though distribution channels other than Apple's official App Store. This potential increase in their ability to influence Apple indirectly comes about as a result of the digital nature of the platform. This is illustrated in the story of Google Voice (story #18) where Google bypassed Apple by developing an HTML5 based variant of the its app, before its native code variant was eventually allowed on to the iOS platform.

6.3.2 Precipitating control through digitalisation

Whilst the properties of digital objects, as described in the literature review, may help explain the facility digitalised objects have for innovation, the argument can also be turned on its head. The properties of digital objects that have been documented thus far (Faulkner and Runde 2010; Kallinikos et al. forthcoming; Yoo et al. 2010a; Zittrain 2009), can be extended, and in some cases inverted, and combined with theories of formal organisation control (Eisenhardt 1985; Ouchi 1977; Ouchi 1978; Ouchi and Maguire 1975). This can provide insights into how digitalised objects are uniquely positioned to enhance a platform owner's ability to specify, measure, evaluate behaviours and outcomes. This argument is first applied to the potential of enabling the behavioural control of platform innovation, before being applied to the potential of enhancing outcome control.

The potential to enable behavioural control

The characteristic of digital artefacts for recombinability (Faulkner and Runde 2010) is potentially the target for the behavioural control of the innovation of digital objects, such as apps. The ability to control developers' behaviour as they recombine modules (Kallinikos et al. forthcoming), such as APIs to create apps is one half of the control of innovation on platforms. Whilst the characteristic of recombinability may suggest the potential for unbounded innovation (Yoo et al. 2010a) through the generation of many permutations of modules, limits can be placed on the degree of allowable permutations through the application of specified rules.

These rules governing allowable permutations of modules can be applied, or measured, in real time, which suggests the surveillance of behaviour. This potential for surveillance is further facilitated by the granular nature of digital objects (Kallinikos et al. forthcoming) which implies that behavioural control can involve measurement and possible intervention at the minutest level.

Kallinikos et al. (forthcoming) suggest that digital objects are borderless and can be distributed over a variety of physical locations. In this way behavioural control, or the measurement and evaluation of adherence of action to specified rules can be decentralised and applied by software running on developers' workstations, thus making use of the property of digital objects to be infinitely expansible (Faulkner and Runde 2010). Furthermore the ability to evaluate, or compare actual behaviours to specified behaviours, and to make decisions based on the outcomes of these comparisons can be automated by making use of the programmable properties of digitalised artefacts.

From time to time, the controlling party may wish to change and re-specify the rules as to how modules can be recombined, which can be implemented by making use of the characteristic of reprogrammability (Yoo et al. 2010b). Depending on the outcome of the evaluation of the how modules are being recombined, the behaviour may or may not be allowed, which provides evidence of the materiality of digital objects, as opposed the immaterial nature of digital objects (Yoo et al. 2010b).

Through the process of behavioural control, organisations can inscribe behavioural outcomes into digital technology (Hanseth et al. 1996) by specifying and designing affordances into digital artefacts which constrain users to certain prescribed courses of action. In this way some of the elements of generativity (Zittrain 2009), such as leverage, adaptability and accessibility are inverted, such that controlling parties attempt to design tools that enable desirable digital innovation, whilst controlling against undesirable innovation, and the ability to modify these tools to facilitate undesirable forms of innovation, or the use of unapproved tools. Finally recombinability is controlled to an extent through the ability of the platform owner to decide which modules are made visible to developers for use, and which are kept invisible.

The natural location for this type of control is within an integrated developer environment (IDE) specified by a platform owner for use by third party developers of platform complements. In this way their application of control converts a boundary resource (Ghazawneh and Henfridsson 2010) into resourcing and securing entity. Following on from the earlier discussion concerning Apple's apparent preference for outcome control, it was suggested that effective behavioural control requires knowledge of a task programme. With the knowledge of a task programme comes the ability to specify what actions are required. The issue that Apple faces is that as it purposely avoids specifying what innovative outcomes it requires, in order to maximise potential the range of innovation, it is unable to specify how these unknown developments should be created. It is, however, in a position to specify undesirable innovative actions, such as the combining of functionality that may lead to undesirable outcomes. It is the avoidance of these actions that can be digitally encoded into a developer environment as behavioural control in the form of affordances in design.

The potential to enable outcome control

The property of recombinability (Faulkner and Runde 2010) can be inverted in order to develop an additional notion of decomposability. Decomposability, and the ability to inspect data that results, can be thought of as the target for outcome control, which is the other half of the formal control of platform innovation. Decomposability is enabled by the granular and modular nature (Kallinikos et al. forthcoming) of digital objects, which means that they can be deconstructed into components. The deconstruction of digital objects, and their subsequent measurement and evaluation can go down to the lowest level of resolution, namely to individual bits, with the implication that everything innovated by the developers, as controlees, is potentially visible to the platform owner as controller. The granular nature of these innovations as digital objects, therefore allows for the inspection, comparison against specified rules and judgement of standalone components, assemblages of components as code, and functionality that they yield. Judgements and evaluations involve decisions, which can by automated through algorithms, which in turn implies programmability. Furthermore, whilst the digitalisation of artefacts may by partially enabled by their property of being potentially distributed (Kallinikos et al. forthcoming) or decentralised, this does not prevent the centralised nature of some digital operations. For example, whilst behavioural control may be applied efficiently as a decentralised function, the evaluation of the final products of apps development through outcome control may be expressed as a centralised function under the direct auspices of the platform owner as controller.

The natural location for digitally enhanced outcome control is in the measurement and evaluation functions of an app review process such as Apple's. The application of digital techniques would allow for automated and deep inspection of code in order to identify and filter out breaches of outcome specifications. However whilst this automation may well facilitate the identification of code that breaches specifications that can be objective defined, such as the calling of private APIs, it is not well suited to subjective decisions. So for example, whilst the use of private APIs could be effectively detected, the decision as to whether an app was in poor taste would be more challenging.

Digitalised control of innovation in practice

This research has found Apple to be a secretive organisation. Little publically available information was found with regards to the internal processes that Apple uses for the measurement and evaluation of iOS developer innovation. Copies of the specifications for control, which Apple prohibits developers making publically available, were found on the public internet. Consequently, little was discovered by way of how Apple makes internal decisions within the Apps Approval Process. It is safe to assume that, given the volume of applications on which Apple must pass judgement, significant resources must be dedicated to this function in order that it works efficiently and effectively. On the one hand, it is assumed that significant human input is required in order that subjective judgements concerning the content and function of the submitted apps can be made. On the other hand, significant automated and digitalised processes would facilitate the verification of the form of the submitted code against objective specifications of outcome control.

In contrast, the locus of where Apple would most logically implement behavioural control is more visible. The integrated development environment that Apple makes available to developers for the construction of iOS code is known as X Code IDE. Whilst this research did not study this development environment, there is little to suggest, by way of developer comments posted on the internet, that this environment is used to regulate the form and functionality of iOS code. At a very cursory level, it is assumed that Apple does not use this IDE as a means to implement behavioural control. Furthermore, developers are free to use other IDEs, such as AppCode and JetBrains, in order to develop iOS code.

Evidence from one of the stories used in this research would suggest that Apple might be hesitant to specify behavioural control in the building of code. The story of Adobe Developer Tools (Story #4) concerns Apple's attempt at prohibiting IDEs other than its own. Apple attempted to ban alternative IDEs such as Adobe developer tools in March 2010 to the general ire of developers. After considerable mounting pressure from the developer community, many of whom threatened to stop developing iOS apps, Apple reversed its decision in September 2010. This reversal allowed developers the freedom to choose which IDE they use. Even though Apple might have the architectural and legal power to impose behavioural control within a specified IDE, it is quite possible that they lack the social power (Lessig 2006) to maintain this control. The reason for this is simple, developers, a vital resource (Pfeffer and Salancik 2003) upon which Apple depend to maintain the value of their ecosystem might dessert them.

6.4 Unfolding the paradox of control and generativity

The theory of digital innovation that is put forward in this thesis describes and explains processes of innovation that occur in circumstances of tension and conflict. Tension occurs as two parties, platform owners and developers want opposing outcomes concerning the potential innovation of platform complements. They enter conflict after one party confronts the other by attempting to realise an innovation, and the other reacts to it by blocking the innovation. The conflict plays out as a series of interactions, as each of the two parties tries to get the outcome they want.

This conflict appears to be precipitated by a shift in the balance of power resulting from the effects of digitalisation, which dissolves the platform owner’s ability to control the distribution of platform complements to consumers. It is a by-product of digitalisation that enables developers to circumvent Apple's architectural control points in order to access vital resources such as paying consumers. Digitalisation strengthens the position of third party developers.

This section investigates the consequences digitalisation has on digital platform innovation in terms of form. It examines: 1) the form or nature of innovation as a process; 2) the different types of innovation that arise from this process; 3) the changes to the boundaries of innovation that result from the process; 4) the form of the relationship between the capacities of control and generativity.

6.4.1 Digital platform innovation as a dialectical process

The results of this research would suggest that innovation emerges from interactions driven by tension and conflict. As a consequence innovation would appear to be a dynamic process of change rather than occurring as a single discrete act. Van de Ven and Poole (1995) suggest four basic theory types that can form an explanatory basis for processes of change and development in organisations. These types concern life-cycle, teleological, dialectical and evolutionary theories. Each theory type concerns different forms of change event sequences, is driven by different mechanisms and operates at different levels within organisations. Of these four, it would seem that dialectical theory types suit the tensions that occur within processes of digital platform innovation (Ghazawneh & Henfridsson 2012).

The essence of dialectical theories of change concern the progression from tension, to confrontation and then conflict between at least two different parties who have opposing interests. There are a number of possible outcomes from conflict including: the defeat of one party by the other; a stalemate amongst the entities; or a synthesis of the interests of both parties, which is mutually satisfactory, but is different from the positions that both parties initially held.

In this way, dialectical theories explain institutional stability and change in terms of a balance of power between opposing parties (Van de Ven and Poole 1995). Stability between two opposing parties is maintained through struggle and accommodation in order to maintain a status quo. However, should the balance of power shift then one party may decide to confront and battle with the other (Hargrave and Van de Ven 2006). The logical order of events in Hegelian dialectical change starts with a pre-existing dominant state of affairs known as a thesis. When an opposing group has sufficient power, the dominant state of affairs is challenged by an alternative, an antithesis, and the two parties engage, interact and conflict. The most advantageous outcome is a synthesis of the two oppositions that benefits both parties. Unfortunately, this is not guaranteed and the incumbent may have sufficient power to suppress its challenger. Similarly the challenger may have gained sufficient power to simply overthrow the incumbent and replace the status quo with another. Finally, it is possible that a "worst of all worlds" synthesis is agreed by both parties that benefits neither.

At the heart of dialectical theories of change are the concepts of conflict, power, and politics (Hargrave and Van de Ven 2006). Conflict is the generating mechanism of change with these types of theory. Power is a precondition for conflict between two parties to be expressed. Politics, by way of strategies and tactics employed by the opposing parties, are the means by which they interact and engage in conflict. In this way, it would seem that the process theory developed in this thesis, in order to describe and explain contested innovation on digital platforms, could be classified as dialectical in form. There are two parties, the platform owner and the developer, and each has conflicting interests as one would like to see developed code launched as an innovation, whilst the other would like to block it. The tension is magnified by the fact that the digital nature of the platform is such that the developer can bypass the platform owner's control should he wish, and which increases his bargaining power. The two parties may tip into conflict and enter a series of interactions based on the strategies discussed earlier. In this model, Apple's specifications are the thesis, the developer’s code embodies the antithesis, and the outcome of their conflict represents a synthesis in all its forms. The three broad patterns of interactions that are seen merely reflect the possible types of strategies that are employed by the conflicting parties.

Whilst the dialectic entered into by the platform owner and the developer is broadly Hegelian in form, there is also possibly another type of dialectical form within this theory. A dialectical model was put forward by Bakhtin (1981), known as the tension dialectic, which is dialogical in form. This form represents two opposing entities, whose existence mutually implies the existence of the other, such that they exist by nature of their opposition. The two entities exist in tension, and there is constant interplay between the two sides. These dialogical relationships are often manifested in institutional phenomena that exist simultaneously in tension. Examples of these are given by Hargrave and Van de Ven (2006): integration-differentiation; exploration-exploitation; and interdependence-dependence. In this way, this research would indicate that the phenomena of control and generativity exist in a dialogical relationship within digital platform innovation.

6.4.2 Classes of platform innovation as an outcome of dialectical process

Attention is now directed to the types of innovation that result from the dialectical form of the process of contested innovation. Up until now this thesis has explicitly studied the platform complement, in the form of a new app or platform enabler, as the object of innovation. However, in some cases, the outcome of platform innovation is not just a new platform complement. The empirical evidence shows that in some cases Apple's specification for control changes following instances of contested innovation. A change in Apple’s specification for control represents a change in Apple’s design rules, which in turn is a change in architectural arrangements. This implies that there is also concurrent platform innovation in and of itself.

Given the foundation that this thesis has in concepts of platform innovation (Gawer 2009a), which are in part derived from notions of modular innovation (Baldwin and Clark 2000) and architectural innovation (Ulrich 1995), it follows that analysis now turns to these sources. In their critique of traditional categorisations of product innovation, Henderson and Clark (1990) extend classes of innovation beyond the incremental and radical. Their justification for adding two additional classes, modular innovation and architectural innovation, is based on an understanding of new forms of product design. This understanding was built on notions of the decomposability of design (Simon 1962), but also of how technical knowledge is distributed in terms of dominant design (Anderson and Tushman 1990) and design hierarchies (Clark 1985).

Their conceptualisation of classes of innovation revolves around two factors. First, whether the components or modules of a design are merely rehashed or whether they are redesigned internally. Second, whether the architecture, which defines how the modules of a design inter-relate, remains the same or is changed. On this basis:

- Incremental innovation reflects refinement in existing designs without making changes to the conceptual basis of how modules work internally or how they inter-relate architecturally.
- Radical innovation, in contrast, represents a rethinking of the design and capabilities of modules, as well as they architectural way that they inter-relate and the overall capabilities of the system as a whole.
- Modular innovation concerns a novel reconfiguration of the of the functionality that is permitted within a component, according to the relevant design rule (Baldwin and Clark 2000)
- Architectural innovation marks a change in design rules, which allows for new or novel configuration of design components.

Applying this classification to Apple's iOS platform ecosystem the following examples of non-contested innovation can be given. The emergence of the iOS platform with the launch of not just the iPhone and iOS operating system, but also the App Store and iOS Software Development Kit, represented radical innovation. Its launch involved establishing new components and an architecture by which they can be structured and function. The launch of the first iOS app which measured an individual pulse rate using the iPhone's camera represents modular innovation. This is because it is a novel reconfiguration of functionality that already exists and is permitted within the iOS architecture. The launch of subsequent iOS apps which replicate the pulse measurement function, but with minor improvements, such as the ability to log measurements is simply incremental innovation. Finally, an updated release of iOS, which enables new functionality on the iPhone, is an architectural innovation, as it represents a change in the system's design rules. These examples indicate that platform innovation concerning platform complements, such as apps, can be considered to be incremental or modular innovation. In contrast architectural and radical innovation requires the innovation of the architectural arrangements of modular components within the platform itself.

With this in mind Henderson and Clark's classification can be applied to the types of platform innovation that are found in the empirical data. The three strategies employed by developers to resist platform owners control are taken in turn. Henderson and Clark's classification is matched against successful outcomes of each of these three strategies, which is summarised in table 6.2.

Strategy Applied	Successful Outcome Enables	Type of Innovation
Regroup	New App / Platform Enabler	Incremental / Modular
Influence	New App / Platform Enabler	Modular
	New / Modified Specification	Architectural
Bypass	New App / Platform Enabler	Incremental / Modular

Table 6.2 Types of innovative outcomes by successful strategy of resistance

Starting with the least contentious outcome, apps that result from a developer following a regroup strategy are either instances of incremental or modular innovation. These apps are constructed within Apple's specifications. At best they will reflect new combinations of permitted code as modular innovation. However they will most likely be incremental functional advances on apps that already exist.

Apps that enter the App Store as a result of a successful influencing strategy are of interest. In these cases Apple will have either reappraised the way it interprets and applies its rules, or changed the nature of its rules, in order that the app is allowed as an innovation. Those apps that enter the App Store as a consequence of Apple reappraising the way it interprets and applies its rules represent incremental or modular innovation. In accepting that it has made a mistake in the way it interprets its rules, the specifications of control are not changed, nor are the design rules, nor the architecture to boot. However in cases of successful influencing strategies, Apple is forced to change its specification for control. This results in a change to its design rules with respect to how components can be combined, which is architectural innovation. This architectural innovation occurs at the same time as modular innovation. As a result of the architectural innovation, Apple may relent and allow the blocked app, which precipitated a change in design rules, into the App Store. This admitted app is a modular innovation as it represents a novel combination of functionality.

Finally apps which are accessible to consumers through alternative channels as a result of their developers following a bypass strategy do not generally represent architectural innovations. The fact that these apps are distributed through alternative channels means that architectural innovation has at some point occurred. For example the architecture of the iOS device is altered through the act of jailbreaking, which in turn enables jailbroken (bypassing) apps to be distributed. A new means of jailbreaking is an architectural innovation, and the non-approved apps which it allows access to are incremental or modular innovations.

It can therefore be seen that the dialectic process that makes up contested platform innovation can result in different classes of innovative outcome. The object of attention regarding platform innovation has thus far been the platform complement. However, there are also occasions when instances of contested innovation concerning platform complements may also result in additional innovation of platform architecture.

The class of innovation that comes about is largely determined by which strategy a developer is able to successfully pursue. Furthermore, certain instances of a successful bypassing strategy can result in more than one innovative outcome. For example, when, after continuous influencing by developers of platform emulators like C64 Emulator (Story #10), Apple changed its rules concerning platform emulators, two types of innovation occurred. On the one hand there was architectural innovation, as Apple's design rules has changed. On the other hand there was also modular innovation, as apps like C64 Emulator were allowed into the App Store. The poses the question as to whether examples like C64 Emulator are instances of a synthesis of modular and architectural innovation, as mooted in Eaton et al (2012). The research in this thesis identifies and explains the circumstance when modular and architectural innovation occur together. It demonstrates that on most occasions modular innovation follows architectural innovation.

6.4.3 Dialectics, outcomes and the boundary of permissible innovation

The boundary of permissible innovation describes the threshold of what complementary platform innovation is and is not allowed. This boundary is of interest not just as a locus for the innovation of platform complements, but also as a site of innovation of platform architecture. The reason for this is that it forms the basis of the specification of control with regards to what complementary innovation is, or is not, allowed into an app store. In this way it makes up part of a platform's design rules, as it specifies the relationship between platform modules (Baldwin and Clark 2000). Consequently an adjustment in the boundary of permissible innovation will be reflected as a rearrangement of what relationships are allowed amongst platform modular components. This is an architectural innovation (Henderson and Clark 1990) within the platform. With this in mind, the discussion now reflects on the impact of the dialectical nature of contested innovation and the various outcomes of innovation on this boundary.

The earlier literature on modularity and platform assumes that through the establishment of design rules (Baldwin and Clark 2000), the platform owner was able to exercise stable control over platform innovation. These design rules enable the platform owner to pre-specify the interconnections between platform modular components and complement modular components (Baldwin and Woodard 2009). By establishing this control over a platform, the platform owner is able to ensure technological compatibility (Schilling 2000), manage the evolutionary trajectory of the platform (Schilling 2009), and regulate the innovation of complements (Boudreau and Hagiu 2009). This lead to early platform literature (Gawer and Cusumano 2002) that not only assumed that the platform owner had control over innovation of the platform architecture but also assumed control over what platform complements were innovated. This perspective would consider the boundary of permissible innovation as static and firmly under the control of the platform owner.

This research presents evidence to the contrary. It shows that third parties, such as developers, are in certain circumstances, able to influence a platform owner to change its boundary of permissible innovation. Not only does this lead to platform innovation in terms of new platform complements, but it can also lead to innovation of the platform architecture itself as the specification is changed. There is evidence to show that the boundary of permissible innovation is not static but instead dynamic, and that platform owners may not be able to control their platforms as tightly as was first thought.

Subsequent platform literature (Gawer 2009a) suggests that the platform owner gives up responsibility for combining modules and control of the platform innovation of complements. This research indicates that platform owners do not give up control of this type of platform innovation entirely, as evidenced by the blocking actions of platform owners in the empirical data. In this way instances of contested innovation of platform complements follow a dialectical process. In these instances developers are empowered by the consequences of digitalisation in their conflict with platform owners. Following on from this, there are occasions when platform owners will decide to cease blocking an innovation, and move the boundary of permissible innovation to allow the application and those similar to it. However the implicit assumption in this more recent platform research (Baldwin and Woodard 2009) is that the platform owner retains exclusive control over the architecture and over innovation of platform architecture.

This research challenges this assumption, and provides evidence for parties other than the platform owner, influencing the innovation of platform architecture. There are numerous occasions of contested innovation in this thesis (e.g. stories #2, #4, #5, #10 and #18) which resulted in an adjustment of the boundary of permissible innovation.

All this provides evidence that factors such as the digitalisation of platforms have adjusted the balance of power between the platform owners and ecosystem members. Whilst the balance of power still favours the platform owner, their authority and ability to control platform innovation in terms of complements and the platform core is weakened somewhat. The empirical results would indicate that there is not a dramatic weakening of control on the side of the platform owner. Only 7 of 45 instances of contested innovation resulted in Apple adjusting the boundary of permissible innovation and core platform innovation being influenced by the developer. The decision to allow changes to the boundary of innovation still remains with Apple, who only appears to be persuaded after significant pressure from developers over a significant period of time.

6.4.4 Unfolding the paradox of control and generativity

There is a curious relationship between the characteristics of control and generativity. Too much control relative to the platform's generative capacity reduces incentives for developers to innovate platform complements. The value of the platform begins to suffer accordingly. Too little control relative to a platforms generative capacity can lead to unmanageable levels of platform innovation. The platform becomes a "Wild West" where consumers lack trust in the many platform complements that become available. Again the value of the platform suffers. But when the balance between the two capacities is achieved a veritable thousand flowers of innovation bloom, and the platform is vibrant, healthy and valuable. In this way it would seem paradoxical that control could actually fuel generativity leading to increased innovative output in spite of the apparent tension between the two (Tilson et al. 2010).

It would seem that their relationship is broadly dialogical in form (Bakhtin 1981). This form represents two opposing entities, whose existence mutually implies the existence of the other, such that they exist by nature of their opposition. The two entities exist in tension, and there is constant interplay between the two sides. These dialogical relationships are often manifested in institutional phenomena that exist simultaneously in tension. Examples of these are given by

Hargrave and Van de Ven (2006): integration-differentiation; exploration-exploitation; and interdependence-dependence.

Traditionally these types of dialogical relationships are explained as dualisms (Farjoun 2010), consisting of two separate and independent parts. For example, three of Poole and Van de Ven's (1989) four modes of analysis of paradoxical relationships between organisational variables require those same variables to be examined apart. These three approaches are as follows: 1) accept the paradox, without analysing it further; 2) separate the elements of the paradox into different spatial levels and explain the phenomenon from each perspective; and 3) separate the elements of the paradox into different temporal planes and explain the phenomenon from each perspective. This approach to separation is reflected in other theories that attempt to explain organisational variables in dualistic relationships, such as organisational ambidexterity (Gibson and Birkinshaw 2004) and punctuated equilibrium (Gersick 1991).

The disadvantage of the dualistic approach is that it does not seem to account for the fact that variables, such as control and generativity, when in balance, appear to reinforce each other. An alternative approach is therefore put forward by Farjoun (2010), which is to see paradoxical relationships in terms of dualities. As a duality, control and generativity are interdependent and mutually enabling. Control in Apple's platform calls for generativity and inventiveness to find new acceptable ways of innovating. Generativity in Apple's platform calls for control, so that order, stability and an acceptable standard of innovation is produced to maintain the health of the ecosystem. This approach to understanding paradox echoes Poole and Van de Ven's (1989) fourth and final approach. This calls for the paradox to be transcended and an alternative conception of the relationship between the variables to be found.

This sweet and sour dialogical relationship between control and generativity feeds the dialectic tension between platform owner and developer. As the two parties struggle with each other with their respective thesis and antithesis, there is the possibility that a higher level of innovation is reached through synthesis. On the one hand this might simply come in the form of the innovation of a new platform complement that is lifted through conflict to a standard worthy of Apple's curation. On the other, it might result in the considered evolution of the platform itself, through architectural innovation, which in turn enables the innovation of new and exciting platform complements.

6.5 Extending the theory to other platform contexts

Attention now turns to investigating the suitability of this tentative theory of platform innovation for applications in contexts beyond the one used in this study. This final investigation is relatively cursory, and is intended to indicate a space of contexts in which use of the theory could be possibly attempted if it was deemed helpful. There is of course the likelihood of extraneous factors, which have not been considered in this analysis, and which may contradict the conclusions presented here. In this spirit, this investigation starts by identifying the basic characteristics of the phenomenon that is studied. It then attempts at identifying whether the model could be applied in other contexts by matching their characteristics against those of this study.

The phenomenon that is studied contains several basic characteristics. This research is concerned with describing and explaining a process by which the owner of a curated digital platform decides whether new apps and enablers, as platform complements, should be distributed to its consumers. The platform owner largely bases this decision on whether new complements meet certain criteria and avoid others. The platform owner is able to enforce the control of what platform complements are distributed as it is the sole distributor of platform complements over the most convenient distribution channel available to consumers. The developers of platform complements have some power to influence the platform owner in its decision by means of airing their grievances in public and to powerful intermediaries such as regulators. In addition developers can bypass the platform owner to furnish their innovations to consumers through distribution channels that are generally deemed much less attractive than the official route.

The first area to consider is whether the model could be extended within the context of Apple's iOS platform. Assuming that the Apple's mechanism for control remained the same and that it did not allow alternative App Stores there is no reason to believe that developers would continue to contest instances of blocked innovation. It is likely that the model could be applied to not just more instances of contested innovation, if they could be found, in the time frame of this study, but that it might also apply to future instances as well. However, other than describing the possibilities for future action, this model is not equipped to make predictions as to what strategies developers are likely to pursue in future instances of contested innovation on Apples platform. The model could also be extended to include other dynamics of innovation other than the interactions that arise through contested innovation. The grammar of actions that were identified can accommodate the cases of non-contested innovation, where a request that an app be available for distribution is simply accepted by Apple. The grammar can also express the case where an app is rejected and not contested. Furthermore, as can be seen from table 5.2, the grammar is able to express interactions which are more complex than the series of actions that are labelled as influence, bypass and regroup. In fact these sequences of actions can added to each other such that complex cases on contested innovation can be described as combinations of influence, bypass and regroup sequences.

The second area to consider is whether the model is applicable in instances of contested innovation on other digital platforms. The conditions of this theory are that the platform is curated such that the owner applies control based on the outcome of development efforts by third parties. Decisions regarding control are made against specifications. The reward or sanction concerns the right or otherwise to be distributed over a channel to market, through which the platform owner is the unique route. In order that the model functions, the developer must be able to have recourse to public debate or external bodies, as well as a means to digitally circumvent the platform owner’s means of distribution. There is reason to believe that this model could be applied to other platform ecosystems where these conditions hold true. Several examples of where these conditions do not hold true are examined, followed by several examples where they do.

The model would not hold true for the Android mobile platform, which is not curated, simply because the owners of the platform, exercise no control over the point of distribution of platform complements to consumers. An Android user is free to access Android Apps from Google's Play Store, the Amazon App Store, App Brain and numerous other "App Stores" as well as directly from developers. The model may well breakdown in this environment, as the incentives for a developer to protest against Google blocking an app are significantly reduced. Developers can simply distribute their app from one of the alternatives.

In contrast the model applies to Apple's other platform ecosystem based around its desktop and laptop operating system OS X. Apple have built a platform ecosystem around OS X that resembles iOS in terms of methods of control and methods of distribution. In this way specifications for control are documented in a Mac Developer Program License Agreement and

App Store Review Guidelines for Mac OS X Apps, and Apple has a stranglehold on the distribution of OS X apps through the Mac App Store. Finally the model also applies to the regime of control that Microsoft has established within the Windows Phone 7 platform ecosystem. Like Apple, Microsoft curate apps within their ecosystem, consequently they review apps against documented specifications before allowing them on to their app store, the Windows Phone Marketplace. Microsoft has constructed an environment like Apple, where the Windows Phone Marketplace is the only point of distribution for Windows Phone 7 devices that have not been hacked. Furthermore it is likely that the theory would describe and explain the dynamics of the innovation of complements on Facebook's platform, which is also curated, and follows a similar regime to Apple's and Microsoft's platforms.

It is proposed that the theory developed in this thesis can be used to explain and describe the dynamics of contested innovation of complements on curated digital platforms in general, as long as they match Apple's conditions of monopolising the channel of distribution of complements to consumers.

6.6 Chapter summary

This discussion sought to set forth and then take stock of a proposed theory of contested innovation on digital platforms. It starts by presenting the two components of the tentative theory, which first describe and then explain examples of contested innovation of platform complements on Apple's iOS platform. The nature of control is then examined in terms of how Apple is able to ensure that only the apps and platform enablers that meet its satisfaction are allowed onto the platform. This is then matched by an investigation of how developers are able to resist this control. This discussion of the nature of control is then enhanced by examining how the unique digital characteristic of the iOS platform both constrain and enable the control of innovation. Attention is then turned to the characteristics of the process of contested innovation in an attempt to unfold an apparent paradox in digital platform innovation. The paradox concerns the fact that whilst control and generativity would appear to oppose each other, and hinder platform innovation, they would seem to enhance generativity when they are well balanced. Finally the discussion ends with some thoughts as to how the proposed theory might suit contexts other than contested innovation on Apple's iOS platform.

The discussion reveals a theory, which attempts to describe and explain a heady mix of opposing forces and actors that both conflict and yet conspire to produce valuable innovation. The discussion suggests that in certain contexts, contested digital platform innovation is driven by a paradox of control and generativity. This paradox can be explained by suggesting that the control of innovation and the generative capacity of a platform, which enables innovation, exist in a dialogical relationship. These capacities exist as a duality, opposing and yet interdependent and mutually reinforcing each other. The tension that results fuels a dialectical relationship between platform owner and developers of platform complements. The tension between these two actors can erupt into conflict over contested innovation. Both parties are empowered by a number of factors which fan the flames of conflict. Amongst these factors is the digital nature of the platform. On the one hand this empowers platform owners to attempt architectural control, for example by attempting to monopolise the distribution of platform complements. On the other hand, digitalisation empowers developers to bypass the architectural control that platform owners place on them, as exemplified by utilising alternative digital channels for the distribution of banned complements to consumers. This affects, to some degree, the balance of power between platform owner and developer. The net effect of this is that on some occasions, developers are able to influence platform owners to make architectural changes to platforms to enable new types of complements in the form of apps and platform enablers. Following a dynamic process of sequential actions alternately progressing and blocking innovation, contested platform innovation may yield both the innovation of platform complements, and, on occasion, the innovation of the platform itself. Moreover, the effects of digitalisation would appear to allow developers some influence on architectural innovation.

7 Conclusion

This final chapter concludes my thesis. It starts by providing an overview of the research and a summary of the findings. It then proceeds by distilling these findings into a series of contributions to theory, method and practice. The validity and limitations of the research are examined before the thesis closes by considering future research that can build on what is presented here.

7.1 Overview of thesis and summary of findings

The research presented in this thesis is situated in on-going work within information systems, and management studies in general, concerning platform innovation. There is a growing body of platform literature in the field of management studies, ranging from product development to industrial economics. The particular interest that information studies has in platforms is as digital artefacts and as digital infrastructures.

7.1.1 Background and research questions

The particular focus of this study is how the innovation of platform complements is controlled within the context of a particular platform regime. There are certain platform regimes where the owners curate the type of platform complements available to end users. This is typically done in order to maintain the overall quality and value of the platform. The capacity for the platform owner to limit what complements are available for consumption comes down to two factors. The first concerns the ability to monopolise the distribution of complements to consumers. The second is reflected in the platform owner's right to refuse entry of complements, as new innovations, onto a platform and for subsequent distribution.

The process regarding the innovation of complements on digital platforms concerns the initiation, development and commercial implementation of apps and platform enablers. As part of the process of innovation on a curated platform, the platform owner must decide at some point whether a complement that has been developed can be admitted onto the platform as an innovation. This occurs at the start of the implementation stage of an innovation process.

This thesis is not directly concerned with the overall process of platform innovation. It is concerned instead with the process of control that is used to regulate whether the development of a new complement should be commercially implemented on a platform as an innovation. Industry platforms are considered to be modular architectures (Baldwin and Clark 2000). Consequently, the understanding in the platform literature (Baldwin and Woodard 2009; Gawer 2009a) is that since the responsibilities between the actors concerned are fixed and stabilised in advance through design rules (Baldwin and Clark 2000), the platform owner has the power to assert what is, and what is not, permissible innovation. On this basis, it is implicitly assumed that the process employed for controlling whether innovation should be permitted is simple and straightforward. A potential platform complement is either admitted into the ecosystem, or it is not.

However, there are numerous examples of contested platform innovation that indicate that this controlling process is not so straightforward. In such cases the process descends into a series of complex interactions, as the platform owner and developer appear to conflict over whether a complement should be allowed or not, and quite where the boundary of permissible innovation lies. On occasion it is also seen that a platform complement, which is at first refused entry, is, following conflict, admitted into the platform ecosystem. This would challenge the notion that the platform owner is fully in control.

The information systems literature has acknowledged that there is an inherent tension between control and generativity in digital platforms (Ghazawneh and Henfridsson 2012). The information systems literature has even acknowledged that there is a paradoxical relationship between these two forces (Tilson et al. 2010) where the outcome of this tension, rather than being a reduced level of innovation, can actually lead to an increased capacity for platform innovation. Furthermore the information systems literature has even begun to question to the authority of the platform owner in digital platforms (Tilson et al. 2012). However, there is a lack of empirical evidence for this phenomenon by way of description and explanation of what occurs and why.

Consequently the research presented in this thesis aims to answer an overall research question:

What is the nature of control in digital platform innovation?

This research question is then broken down into two sub questions whose aims are twofold. The first seeks to simplify and describe the complex interactions that occur in instances of contested innovation, so that the dynamics of control can be better analysed. This is addressed in the first sub question:

How do the dynamics of the control of innovation change over time?

The second attempts to analyse the dynamics of control in order to provide an explanation as to why they unfold in the way that they do. This is addressed in the second sub question:

How is the control of innovation asserted and lost on digital platforms?

The objective of the research is to develop a theory that describes, explains and enhances an understanding of the dynamics of control in the innovation of platform complements. The outcome is intended to be a low level process theory to seek explanations of examples of contested platform innovation, observed through the interactions between platform owners and developers.

7.1.2 Approach to research

Empirical data was collected concerning multiple examples of digital platform innovation on Apple's iOS platform. Apple's platform was chosen because it matched the required context of a curated platform. These examples of contested innovation were expressed as sequences of developer and platform owner actions as they conflict over the boundaries of permissible innovation. The empirical data concerning the actions taken by both parties was taken from Tech Blogs which recorded details of these instances of contested innovation.

The sequences of actions were then grouped into stories themed around instances of contested innovation. The sequences were then coded using a compact set of actions derived from semiotic squares (Greimas and Rastier 1968) before being expressed as narrative networks (Pentland and Feldman 2007). Following this, the superimposing of different narrative networks, describing all of the observed instances of contested innovation, onto each other allowed generalised patterns of interaction to be found. This enabled analysis to address the first sub question.

The application of a formal theory of control (Ouchi 1977; Ouchi 1978; Ouchi 1979) enabled mechanisms to be identified from within the data concerning how control is asserted and lost. The analysis that this enables addresses the second sub question.

7.1.3 Empirical findings

The findings of this research are broadly divided into two. First, there are findings which are based on empirical data, as presented in the results and analysis chapter. Second, there are findings, which are more speculative in nature and are derived from the discussion chapter.

The findings derived from the analysis of empirical results fell into two further parts matching each of the research questions. The first set of findings largely serves to describe the control of innovation, and the second set of findings seeks to explain the phenomenon. 45 stories concerning contested innovation were found within the empirical data extracted from blog posts. A compact lexicon of eight actions describing developer’s attempts at platform innovation and Apple's attempts to control innovation was then developed using semiotic squares. These eight actions were used to code the stories into a set of 45 sequences of action concerning contested innovation. It was then possible to construct individual narrative networks (Pentland and Feldman 2007) describing each narrative, and then to combine them into one single narrative network. By abstracting the combined narrative network and focussing on the most frequently occurring patterned sequences of action it was possible to identify a generalised sequence of actions that was broadly common across the stories. This generalised sequence of actions could be split into three distinct stages, which could be used to describe the dynamics of the control of innovation. The first stage represents a build-up to tension between Apple and a developer and concludes in Apple blocking a developer’s request for an innovation. The second stage represents a tension between the two parties and describes the developer attempting to influence Apple to reverse its blocking decision. The third and final stage represents a resolution to the tension, and identifies three possible outcomes: 1) Apple may respond to attempts to influence it positively, and allow the innovation onto its platform, possibly refining its rules at the same time; 2) The developer may decide to bypassing the Apple's App Store and Apple's control by seeking to distribute its unchanged innovation to iOS devices through some other channel; 3) The developer may comply with Apple's reasons for blocking the innovation, and regroup and rework the app until it is compliant with Apple's rules.

Attention was then turned to explaining the sequences observed in instances of contested platform innovation in terms of two opposing sets of mechanisms. First, the elements of Apple’s mechanism for controlling innovation, the Apps Approval Process, were analysed using formal organisational control (Ouchi 1977; Ouchi 1978; Ouchi 1979) as a theoretical lens. It was found that Apple favoured specifying sanctions rather than rewards, and that these were focussed on outcomes of the process of innovation, rather than on how innovation was attempted. It was reasoned that Apple specified the control of innovation in these ways as by its very nature, the outcomes and ways of carrying out new innovation are often unknown. It is therefore more conducive for the generation of a broad range of innovation as well as being easier to control, by stating what is not wanted rather than what is wanted and how it should be achieved. An investigation into Apple's specifications for control revealed that they could be influenced and that they can change over time. Second, the means by which developers responded to Apple's control were analysed. It was found that the developers used three mechanisms, as identified, above. It was found that there were occasions when Apple could be influenced to reverse a block. The basis of this reversal was on persuading Apple that it had made an error in one of two ways. Either it had made a mistake in how it applied a specification for control, or, alternatively, what it had applied was incorrect, that an actual specification for control was inappropriate. Either way the boundary of permissible innovation had in some way been affected. Furthermore, it was discovered that the successful application of the bypassing mechanism marked an impasse between Apple and a developer. On the one hand a developer might succeed in bypassing Apple's iOS platform by distributing an innovation to iOS users through an alternative channel. On the other, Apple retains control over what can be innovated on the platform as a result of the implementation of a bypassing strategy, and the boundary of permissible innovation remains the same. Finally it was considered that the successful implementation of a regrouping mechanism represented Apple's retention of control over platform innovation. By regrouping a developer accepts to redesign a blocked app in order to meet Apple's specifications. Consequently Apple retains control over an unchanged boundary of innovation.

A number of additional findings were arrived at when the relationship between mechanisms of control, mechanisms of response to control, and the outcome of contested innovation were analysed. Overall, it seemed worthwhile for developers to resist Apple's control. 31 of the 45 instances of contested innovation ended up with applications being admitted, in some form, onto Apple's platform, following an initial blocking move on the part of Apple. However, only 7 of these were realised as innovations in the iOS platform ecosystem, after Apple had changed its formal specifications for control, which was found to be representative of a change in the boundaries of innovation. Changes in the boundaries of innovation were infrequent. A possible logic explaining developers' use of mechanisms to respond to Apple's control was found. The developer's first sought to influence Apple to reverse it blocking decision, and no evidence was uncovered to suggest that the corresponding action was detrimental to developers. Should this mechanism fail to get Apple to reverse its decision, developers were faced with a choice of bypassing or regrouping. It was suggested from the data that developers bypass Apple if they are determined that an innovation should reach end users unchanged, but are prepared to forego the use of the iOS platform as a distribution mechanism to end users. The data also indicated that developers regrouped if it was essential for their innovations to be distributed from Apple's iOS platform and they were prepared to compromise the design of an innovation to meet Apple's specifications. Finally, following an examination of the most contested set of criteria for control, falling under Apple's motivation to protect iOS from competing platforms, it was concluded that when Apple is prepared to change the boundaries of permissible innovation changed, it is done slowly after considerable effort on the part of large communities of developers.

7.1.4 Theoretical findings

The discussion recasts the findings of the results and analysis section as a tentative theory of the control of platform innovation before going yet deeper in terms of analysis. Attention is turned at first to the nature of control in digital platform innovation in terms of how it is enforced by platform owners, and how it is resisted by developers. It was suggested that three of Lessig's four modalities of regulation (2006) provide Apple with the power required to enforce its App Approval Process as a mechanism of control. Apple's significant share of the smartphone market, it's assumed control over the iOS platform architecture including the distribution of apps, and its legal ownership of these critical resources meant that developers were dependent (Pfeffer and Salancik 2003) upon it for access to consumers. Following this, the role of digitalisation in the control of digital platform innovation was examined. It was suggested that digitalisation on hand dissolved the platform owner's capacity for the control of platform innovation, and, on the other hand, provided news means of control. The loose coupling between the layers of a layered modular architecture (Yoo et al. 2010a) mean that it can be possible for developers to bypass a platform owner's architectural control points (Woodard 2008) and thereby use a platform in unintended ways. To counteract this, digitalisation potentially enables greater platform control. The characteristics of digital artefacts are such that it is possible to monitor code remotely as it is created, and thus generate affordances for permissible combinations of code in software development environments. This is currently not done in major smartphone platforms, but may lead in future to a greater emphasis on behavioural control of platform innovation rather than on output control. Finally, the discussion moves on to attempting to unfold the paradox of control and generativity in digital platform innovation. This paradox can be explained by suggesting that the control of innovation and the generative capacity of a platform, which enables innovation, exist in a dialogical relationship. These capacities exist as a duality, mutually interdependent, opposing and yet mutually reinforcing each other. The tension that results fuels a dialectical relationship between platform owner and developers of platform complements. The tension between these two actors can erupt into conflict over contested innovation. Both parties are empowered by a number of factors which fan the flames of conflict. Amongst these factors is the digital nature of the platform. On the one hand this empowers platform owners to attempt architectural control, for example by attempting to monopolise the distribution of platform complements. On the other hand, digitalisation empowers developers to bypass the architectural control that platform owners place on them, as exemplified by utilising alternative digital channels for the distribution of banned complements to consumers. This affects, to some degree, the balance of power between platform owner and developer. The net effect of this is that on some occasions, whilst platform owners retain overall control, developers are able to influence them to make architectural changes to platforms to enable new types of complements in the form of apps and platform enablers. Following a dynamic process of sequential actions alternately progressing and blocking innovation, contested platform innovation may yield both the innovation of platform complements, and, on occasion, the innovation of the platform architecture itself.

7.2 Research contribution

The following paragraphs aim to distil the preceding detailed overview of the research and its findings, in order to summarise the contributions of this research to theory, method and practice.

7.2.1 Contribution to theory

The primary theoretical contribution of this thesis is to the information systems literature concerning innovation and digital infrastructures, of which digital platforms like Apple's iOS form part (Tilson et al. 2012). The understanding of the process that governs the control of innovation in digital platforms is not well understood in the information systems literature (Tiwana et al. 2010).

The notion that software platforms contain a tension between control and generativity is encountered repeatedly across the growing body of information systems research into software platforms. There is consequently an emerging consensus (Eaton et al. 2012; Ghazawneh and Henfridsson 2012; Tilson et al. 2010; Tilson et al. 2012; Tiwana et al. 2010) that digital platform innovation is often a dynamic process full of tension. In their research commentary concerning the interplay of architectural design, platform governance, and the environmental dynamics of ecosystems on the evolution of platforms, Tiwana et al. (2010) refer to a delicate balance of control by a platform owner and autonomy amongst developers. Tilson et al (2010; 2012) discuss a tension which arises when platform owners seek to gain the advantages of generativity through third party innovation whilst attempting to control this innovation in their own interests. The same is the case in (Ghazawneh and Henfridsson 2012) when they present research which reports the need for platform boundary resources for resourcing and securing platform ecosystems through third party development. They refer to "a tension in the design of these boundary resources between maintaining platform control and stimulating 3rd party developers to build" (p. 2). Furthermore, the idea that actors, such as third party developers, can challenge platform owners' assumed control over innovation has also surfaced in IS platform research. Tilson et al. (2012) acknowledge this possibility, and, using the terminology of Eaton et al. (2011) suggest that platform owners control of innovation might be "influenced" by other ecosystem members.

There is a need for empirical research both to describe and explain the tensions that exist on some digital platforms. This dissertation uses empirical evidence to develop a process understanding of the innovation of complements on digital platforms. It shows: first, that the control of innovation is a complex and dynamic process; second, that it is indeed driven by a tension between control and generativity.

The primary theoretical contribution of this thesis to the IS knowledge is to develop and present a process theory of contested innovation of complements on curated digital platforms, which describes and explains the dynamics of digital platform innovation. The first part of this contribution is provided by empirically demonstrating how digital platform innovation can be described as a set of patterned sequences of generic actions between developers of platform complements and platform owners. These serve to demonstrate that the innovation of complements on digital platforms can be both a dynamic and contested process. The second part of this contribution is through explaining how this process is both dynamic and contested. It demonstrates how instances of contested innovation can be explained as a dialectical process between developer and platform owner. How on the one side, the platform owner attempts to control for the innovation of a developers platform components using specifications and rules to enforce formal control. And how on the other side, the developer attempts to resist the platform owners control using a number of strategies. Furthermore the theory explains how this dialectic is propelled by a dialogic tension between the forces of control and generativity. Whilst the application of dialectical thinking is not novel within the field of IS, for example in terms of the dialectical relationship between standards and flexibility (Hanseth et al. 1996), the granularity of analysis afforded by analysis of sequences of generic actions over 45 cases of contested innovation is. An additional element of this contribution to the IS literature of digital infrastructure, is that the digitalisation of platforms dissolves, to some extent, the capacity for a platform owner to enforce the control of innovation. The rebalancing of power between the developer and the platform owner that results from this increases the intensity of the dialectic between the two parties.

This last point leads to an additional set of contributions with respect to the literature concerning technology and innovation management. Until now, information systems understanding of the control of platform innovation is inherited from the reference literature (Gawer 2009b). In this view, the platform owner has the capacity to control what innovation in the form of complements is admitted into a platform ecosystem, simply by opening or closing the platform accordingly. This assumption is based on two underlying principles: property rights (Hart and Moore 1990) and design rules (Baldwin and Clark 2000). The platform owner has ownership of the platform architecture, and consequently the decision rights (Baldwin and Woodard 2009) over the design rules, which define how modular components of complements interface with and draw upon the functionality of modular components of the platform. Property rights bestow "bouncer's rights" (Strahilevitz 2006) upon the platform owner, which enables the ejection or blocking of complements that are deemed undesirable. Consequently it is assumed the process of controlling platform innovation is straightforward, that it is a simple decision concerning rejection or admittance, and that the developer has no recourse. However, the reference literature has paid little attention as to how the digitalised essence of the IT artefact may affect understanding of platform architecture and innovation. The implications of the process of digitalisation rebalancing power between platform owner and developers provide further contributions, and an opportunity for the field of IS to make a unique contribution to the reference literature.

With this in mind, the second theoretical contribution of this thesis concerns the need to readdress the balance of power concerning the control of platform architecture. This thesis demonstrates that the process of digitalisation has dissolved platform owners control over innovation by enabling developers to use bypassing strategies, which increases their bargaining power with respect to the platform owner. This results in occasions where developers are able to influence platform owners to change specifications of control concerning which types of platform complements are allowed or not. The implication of this is that third parties are able to influence architectural innovation, which is no longer entirely in the control of the platform owner, and challenges the prevailing view in the technology and management of innovation literature (Baldwin and Clark 2000; Baldwin and Woodard 2009).

A third contribution which follows on from this, describes the concept of a "boundary of permissible innovation". In the context of a curated platform, this boundary delineates types of platform complements that can be implemented, from those which are not allowed. Not only is it a useful concept to understand the nature of a curated platform, but it also strengthens the argument which questions the absolute control of the platform owner over the architecture of a platform. The boundary of permissible innovation distinguishes those architectural arrangements of platform and complement component modules that are allowed from those which are not. This term therefore describes the part of the design rules of the platform concerning allowable interfaces between modular components of platform and complement. When a third party, such as a developer, is able to influence a platform owner to change a specification in order to allow a new type of complement, the boundary of permissible innovation shifts, which again indicates that the architecture has changed and is innovated anew. Not only does this challenge the idea that the platform owner controls architecture, it also challenges, given the frequency of changes in the boundary of permissible innovation seen across the instances of contested innovation, the view that architectural innovation is an infrequent event vis a vis modular innovation (Henderson and Clark 1990).

The contributions that this research provides to the technology and innovation management literature have potential implications on concepts related to the debate around modular architectures, such as notions of dominant design and incumbent firms. These potential implications need to be given more consideration.

7.2.2 Contribution to method

The second potential area for contribution concerns method. A different approach is needed for researching into issues of digital platform evolution including the dynamics of the innovation of platform complements. A key concern is that the phenomenon is distributed across organisational boundaries and stakeholders. This issue makes the study of sequences of actions more challenging than within the boundaries of an organisation.

The choice of a method of sequence analysis in order to study the processes of innovation is not unusual (Van de Ven and Poole 1990). Furthermore there is nothing new in any of the components used for the method of sequence analysis in this research. The use of narrative networks as a method of sequence analysis is established in the organisation studies and information systems literature (Pentland et al. 2010; Pentland et al. 2011). The use of blogs as source of data is beginning to emerge in IS research (Davidson and Vaast 2009; Vaast and Davidson 2008). Nor is the use of the Greimas square in the production of coding scheme novel (Fiol 1989).

However the use of these elements in combination in order to carry out sequence analysis is new. It was found to be an appropriate means to uncover and describe interactions across organisational boundaries, as it enabled interesting analysis and theorisation. This combination of elements has the potential to be useful in future research that requires sequence analysis, and that is reported online in the public domain. But whilst it is an unusual, and possibly interesting, combination of approaches, it does not form a contribution to method in its own right.

7.2.3 Contribution to practice

Finally, the third area of potential contribution concerns practice. This thesis provides insights into the management of a highly successful curated platform, which can be adopted by others. Digital platforms and their associated ecosystems are becoming an important source of direct and indirect value generation. Whilst there may be numerous factors that contribute to a successful digital platform ecosystem, the management of innovation of content and applications, which is the balance of controlling and encouraging third party innovation, is essential. This dissertation describes in detail how, at the time of writing, one of the most successful digital platform owners, Apple, established a regime of governance that manages that function. Whilst there are numerous other factors that may contribute to a successful platform, such as the ability to find relevant apps, quality of hardware, and effective tools and capabilities for developers, it is hoped that this research may inform owners and potential owners of other digital platforms with regards to effective governance. The ability to monopolise distribution and judiciously select what complements are distributed ensure an effective curated regime to be implemented and managed. In this environment it is essential to have clear rules with regards to what will not be tolerated, rather than specifying what is desirable, as this maximises the space for third party innovation. Tension between developers and platform owners appears important to the process of innovation and to that end policies must be strictly enforced an innovation refused. However, it is important that there are ways out to diffuse tension and encourage innovation. First developers must be able to be admitted in an ecosystem at a later date if they rebuild their code to match desired specifications. Second alternative distribution channels can be tacitly tolerated for apps that are rejected, but it is essential that the user experience in these alternative channels is inferior to the one that is monopolised by the platform owner. Finally, although there may be bottom lines, such as the integrity of the platform, which cannot be compromised, room for compromise over platform architecture can lead to novel innovation through synthesis.

7.3 Validity and research limitations

Quality is ensured in the process of research so that the claims made are truthful and to enable research to be reviewed and made accountable to agreed standards (Bauer and Gaskell 2000). Quality criteria and frameworks are well established for research using quantitative case study methods (Yin 1994). The work described in this thesis is, however, presented as qualitative research and these quantitative quality criteria and frameworks do not translate well (Bauer and Gaskell 2000). The concepts of reliability and validity of measures, sample size and representative sampling are appropriate in quantitative research where they are easily applied and make logical sense. However, these concepts are not easily applied in qualitative research, as in this thesis. In qualitative research it is often the range and variety of meaning in a corpus of data that is of interest, where interpretation and subjectivity of data may not be compatible with reliability of measures. The following quality criteria, based on Bauer and

Gaskell (2000), are to be applied to the qualitative research carried out in this dissertation: transparency and procedural clarity; corpus construction; thick description; contribution; communicative validation. The aim is that by fulfilling of these criteria, both confidence in and relevance of qualitative research can be assured. Each of these criteria is now taken in turn and evaluated against the research presented in this thesis. By uncovering weaknesses, it is hoped that opportunities for future research can be found.

Great efforts were made in order to document and make transparent the procedure of data elicitation and analysis undertaken in this research. Whilst not explicitly presented in this thesis, each of the 4664 blog entries captured as empirical data is documented. The transformation of this data into 45 stories is logged, as are all the transformations made on the data in order to arrive at the analysis. Furthermore it is hoped that the procedural steps to carry out this analysis are well documented in the methods section. As a consequence I believe this research could be reconstructed and imitated with reasonable ease.

The concept of corpus construction in qualitative research is functionally equivalent to representative sampling and sample size in quantitative research (Bauer and Gaskell 2000). However, rather than having an objective of ensuring that the distribution of a sample is representative of a wider population, the aim is to maximise the variety of unknown representations. In this respect the research presented in this thesis built an extensive corpus of data regarding the phenomenon. First, it collected data over the entire window of time up until the point of data analysis that the phenomenon (third party innovation on the iOS platform) existed. Second, the source from which data was obtained is fairly exhaustive in its coverage of the phenomenon. Whilst I cannot be certain that every instance of contested innovation on Apple's platform that was reported in the blogosphere was captured, I have confidence that much of it was. My confidence that the range of observations captured was sufficient is supported by the fact that no novel observations were seen in the data after thirty stories were collected.

As a marker of relevance it is thought (Bauer and Gaskell 2000) that research must reveal something of interest in terms of a surprising insight or contrary evidence. In this way, I would cautiously offer the insight that this thesis provides into the contested nature of the control of platform innovation, which questions the prevailing consensus (Baldwin and Clark 2000; Baldwin and Woodard 2009) regarding the absolute control a platform owner has over architecture and the corresponding design rules of a platform. The research presented in this dissertation indicates that developers can generate sufficient power that they can influence the architectural design of a platform themselves.

Communicative validation tests the relevance of qualitative research by presenting the analysis to the sources of the data and seeking their corroboration. Again, this research was lacking in this respect, partly on account of its remoteness to the actors being studies. Given the secretive nature of Apple as an organisation, it is thought unlikely that a source within the company would be willing to officially, or unofficially comment on the findings. It is, however, more likely that third party developers would be willing to comment on the research, despite Apple's occasional attempts at controlling third parties commenting on the Apps Approval Process. In this respect this research may well have fallen victim of its methodology, where data was garnered indirectly from online sources, rather than directly from informants.

The objective of thick description is to provide confidence in qualitative research by providing "extensive verbatim reporting of sources" (Bauer and Gaskell 2000). In this respect the research presented in this thesis also falls short. The objective of this research was to describe and explain broad patterns and sequences of actions across numerous examples of contested innovation. The focus of the research was on a small set of actions at an abstracted level, and little attention was paid to the detailed contextual information within the textual detail of the blog entries. In this way the study reflects what Abbott (1992) terms the narrative positivism of focussing on sequences of events in themselves.

The approach taken analysing the unfolding of events in this research is based on narrative networks (Pentland and Feldman 2007), which is a form of sequence analysis in the broader tradition of narrative analysis. Sequence analysis was chosen over other more contextually focussed means of narrative analysis, as the intention of the study was to recognise broad patterns of event sequences across many different instances of contested innovation. An approach to narrative analysis which focussed on the deeper context of stories (Barthes 1977) would simply be too time consuming given the range of examples to be studied. The research could have alternatively used one of the more formal methods of sequence analysis (Abell 2004; Corsaro and Heise 1990). However it was felt that given the novelty of sequence analysis to the researcher and the fact that many other aspects of the study, such as the use of blogs as a data source, were novel, then the simplicity of narrative networks were most appropriate. The final advantage of narrative networks was the fact that they facilitated the construction of

Markov matrices which were essential when combining the stories to form an overall impression of the key sequences of actions amongst the stories.

Finally, it must be noted that the theoretical analysis was limited to applications of Ouchi's formal modes of control (1977,78). There are, of course, alternative modes of control. For example informal modes of control, such as clan control (Kirsch 1997) are achieved by people strategies to ensure that individuals have understood and internalised organisational goals, thereby minimising the divergence of their preferences. The decision was taken to focus on formal modes on control alone in order to provide focus to the research.

7.4 Future research

It is proposed that the theory developed and presented in this research applies to the generalised context of contested innovation of complements on curated digital platforms.

However, the empirical data that was used to develop the proposed theory concerned instances of contested innovation on Apple's iOS platform in isolation. In addition, the source of data, in the form of blogs, is somewhat remote from the parties involved in the tensions and conflicts that arise. Finally in seeking to focus on broad patterns of sequences of actions alone, the context of these tussles is set to one side. Focussing on the qualitative nature of this research, it is apparent that it lacks both thick description and communicative validation, which if addressed may increase confidence in the research as well as well as its relevance.

Future research could focus on breadth as well as depth. It would be of interest to test the theory against other digital platforms, such a Microsoft's Windows Phone 7 platform, where content is curated in a similar manner to Apple's iOS. The purpose of this would be to validate the theory, or to find reasons why it does not hold.

In terms of focussing on depth, there might be much to be gained by attempting to interview both developers and individuals involved with a platform owner's App Approval Process. It is recognised of course, that given its secretive nature, access to Apple might be challenging. However, a more in depth study could be fruitful for two reasons. On the one hand it could be used as a means to validate the model, and on the other, it may add contextual detail to improve an overall understanding. The most extreme examples of contested innovation could be taken from the thesis, and the platform owner and respective developer approached. Beyond validating the basic actions and sequences observed it would be of interest to get both parties understandings of strategies of control and resistance. This would include eliciting reasons why particular courses of action were taken, find out their understanding of who is in control and their interpretation of the paradox of control and generativity.

7.5 Final remarks

And so my story appears to come to an end. And it ends as it starts: "Axio Era y No Era".

My story is based on studying stories. I use aspects of narrative analysis in order to analyse stories concerning conflict to find a common fabula. In this way, Apple is a powerful actor who attempts to control which apps and platform enablers it admits into its iOS platform ecosystem. At times it rejects platform complements, created by third party developers and which it deems unsatisfactory, from its platform ecosystem. It is understandable that some of these third parties contest Apple's decision, as for many it represents the only viable channel to Apple's lucrative consumers. Some developers capitulate and change their inventions to satisfy Apple. Others try their luck elsewhere. However some, after much effort, are able to influence Apple to change its ways and allow their innovation and others like it.

It is the digital characteristics of software platforms that enable developers to realise their inventiveness. It is also the digital characteristics iOS that enable developers to apply their inventiveness to bypass Apple's control if needs be, and this poses a threat to Apple. As a consequence the balance of power between developer and platform owner may, to some extent, be equalised with regards to what is innovated. And yet, the processes of digitalisation could afford Apple greater control over how developers build platform complements. In some ways digitalisation may have eroded Apple's control over platform innovation, and yet in other ways it may enhance it. I sense there may be yet more stories to tell.

"It was and it was not - Axio era y no era ...”

Appendix A Web References

Chapter 1

[1.1] 'An iPopping Phenomenon', The Economist 24 March 2012, accessed 01 April 2012, <http://www.economist.com/node/21551058>

[1.2] Wilson, H (2012). 'Apple is world's most valuable company after iPhone frenzy drives record profits', The Daily Telegraph, 25 Jan 2012, accessed 01 April 2012, <http://www.telegraph.co.uk/technology/apple/9037186/Apple-is-worlds-most-valuable-company-after-iPhone-frenzy-drives-record-profits.html>

[1.3] Kumparak,G (2012). 'Apple’s Q1 2012: $46.3B In Revenue, 37M iPhones And 15.4M iPads Sold', Techcrunch, 25 Jan 2012, accessed 01 April 2012, <http://techcrunch.com/2012/01/24/apples-q1-2012-46-3b-in-revenue-37m-iphones-and-15-4m-ipads-sold/>

[1.4] 'Apple has paid out $4 billion to iOS developers', Edible Apple, 22 Feb 2012, accessed 01 April 2012, <http://www.edibleapple.com/2012/02/22/apple-has-paid-out-4-billion-to-ios-developers/>

[1.5] Ashdown, D (2012). 'Nokia Puts All Its Chips on Windows at Vegas', Juniper Research Blog, 18 January 2012, accessed 01 April 2012, , <http://www.juniperresearch.com/analyst-xpress-blog/2012/01/18/nokia-puts-all-its-chips-on-windows-at-vegas/>

[1.6] Ziegler, C (2011). 'Nokia CEO Stephen Elop rallies troops in brutally honest 'burning platform' memo? (update: it's real!)', Engadget, 08 February 2011, accessed 01 April 2012, <http://www.engadget.com/2011/02/08/nokia-ceo-stephen-elop-rallies-troops-in-brutally-honest-burnin/>

[1.7] Wordsworth, R (2011). 'Android Market reaches 500,000 app mark', T3, 23 October 2011, accessed 01 April 2012, <http://www.t3.com/news/android-market-reaches-500000-app-mark>

[1.8] Bonnington, C (2011). 'Google’s 10 Billion Android App Downloads: By the Numbers', Wired.com, 08 December 2011, accessed 01 April 2012, <http://www.wired.com/gadgetlab/2011/12/10-billion-apps-detailed/>

[1.9] Gruber, J (2010). 'Where Are the Android Killer Apps?', Daring Fireball, 19 November 2010, accessed 01 April 2012, <http://daringfireball.net/2010/11/where_are_the_android_killer_apps>

[1.10] 'Apps Approval Process' 2012, Apple, accessed 01 April 2012, <https://developer.apple.com/appstore/guidelines.html >

[1.11] '500,000 apps on the App Store' 2012, Apple, accessed 01 April 2012, <http://www.apple.com/iphone/built-in-apps/app-store.html>

Chapter 5

[5.1] Kendrick, J. (2009). 'T-Mobile Germany Blocks iPhone Skype Over 3G and WiFi ', jkOnTheRun, 07 April 2009, accessed 17 February 2012, <http://jkontherun.com/2009/04/06/t-mobile-germany-blocks-iphone-skype-over-3g-too/>

[5.2] Schatz, A. (2009). 'Group Presses FCC to Back Skype', Wall Street Journal, 03 April 2009, accessed 17 February 2012, <http://online.wsj.com/article/SB123876873806886721.html>

[5.3] internetnews.com, (2009). 'AT&T's Skype Move Won't Stop Net Neutrality Push ', internetnews.com, 07 October 2009, accessed 17 February 2012, <http://www.internetnews.com/government/article.php/3842736>

[5.4] Lardinois, F. (2010). 'Banner Exchanges Come to the iPhone: AdMob Launches Download Exchange ', ReadWriteWeb, 31 March 2010, accessed 17 February 2012, <http://www.readwriteweb.com/archives/admob_launches_download_exchange_for_iphone.php>

[5.5] Omar, (2010). 'Mobile advertising and the iPhone ', The Life and Times of AdMob, 09 June 2010, accessed 17 February 2012, <http://blog.admob.com/2010/06/09/mobile-advertising-and-the-iphone/>

[5.6] Justin, (2010). 'Mobile Advertising Is At A Turning Point, With Google, Apple And The FTC Determining The Future ', Mobile Marketing Watch, 07 April 2010, accessed 17 February 2012, <http://www.mobilemarketingwatch.com/mobile-advertising-is-at-a-turning-point-with-google-apple-and-the-ftc-determining-the-future-5991/>

[5.7] Hamoui, O. (2010). 'An Update on Apple's Terms of Service their apps.', Google Mobile Ads Blog, 09 September 2010, accessed 17 February 2012, <http://googlemobileads.blogspot.com/2010/09/update-on-apples-terms-of-service.html>

[5.8] Frommer, D. (2010). 'Adobe Flash Coming To Apple's iPhone', Silicon Alley Insider, 17 June 2010, accessed 17 February 2012, <http://www.alleyinsider.com/2008/6/adobe_flash_apple_iphone_maybe_someday>

[5.9] Eran Dilger, D. (2010). 'Dropping Adobe Flash boosts Apple's MacBook Air battery life by 2 hours ', AppleInsider, 04 November 2010, accessed 17 February 2012, <http://www.appleinsider.com/articles/10/11/04/dropping_adobe_flash_boosts_apples_macbook_air_battery_life_by_2_hours.html>

[5.10] Carr, A. (2010). 'Adobe CTO on MacBook Air, HTML5: Flash Battery Problems a “False Argument” ', Fast Company, 08 November 2010, accessed 17 February 2012, <http://www.fastcompany.com/1700949/adobe-on-macbook-air-html5-flash-battery-problems-a-false-argument>

[5.11] Marsal, K. (2010). 'Skyfire iOS browser approved by Apple, converts Flash video to HTML5 ', AppleInsider, 02 November 2010, accessed 17 February 2012, <http://www.appleinsider.com/articles/10/11/02/skyfire_ios_browser_approved_by_apple_converts_flash_video.html>

[5.12] Siegler, M. (2011). 'Steve's Last Laugh: Adobe Killing Off Flash For Mobile Devices ', TechCrunch, 09 November 2011, accessed 17 February 2012, <http://techcrunch.com/2011/11/08/mobile-flash-is-coming-soon-i-swear/>

[5.13] Ha, A. (2009). 'The iPhone finally gets Flash apps', VentureBeat, 05 October 2009, accessed 17 February 2012, <http://digital.venturebeat.com/2009/10/05/flash-apps-finally-come-to-the-iphone-but-not-flash-websites/>

[5.14] Gruber, J. (2010). 'New iPhone Developer Agreement Bans the Use of Adobe's Flash-to-iPhone Compiler ', Daring Fireball, 08 April 2010, accessed 17 February 2012, <http://daringfireball.net/2010/04/iphone_agreement_bans_flash_compiler>

[5.15] Goldman, J. (2010). 'Adobe Pricey Love Note to Apple ', Tech Check with Jim Goldman, 14 May 2010, accessed 17 February 2012, <http://www.cnbc.com/id/37129479>

[5.16] Shankland, S. (2010). 'Apple relents on Flash-derived iPhone, iPad apps ', CNET News, 09 September 2010, accessed 17 February 2012, <http://news.cnet.com/8301-30685_3-20015954-264.html>

[5.17] David, A. (2010). 'Apple Denies Free Political Speech ', Yes, But, However!, 22 May 2010, accessed 17 February 2012, <http://yesbuthowever.com/apple-denies-free-speech-9000007/comment-page-1/>

[5.18] Brustein, J. (2010). 'Apple's Struggle With Political Apps', Bits, 26 May 2010, accessed 17 February 2012, <http://bits.blogs.nytimes.com/2010/05/26/apples-struggle-with-political-app/>

[5.19] Siegler, M. (2009). 'Feel Like Shaking A Baby To Death? There's An App For That.', TechCrunch, 22 April 2009, accessed 17 February 2012, <http://www.techcrunch.com/2009/04/22/feel-like-shaking-a-baby-to-death-theres-an-app-for-that/>

[5.20] Arthur, C. (2009). 'Apple pulls iPhone ‘Baby Shaker’ game ', Guardian, 23 April 2009, accessed 17 February 2012, <http://www.guardian.co.uk/technology/2009/apr/23/apple-iphone-baby-shaker>

[5.21] Kovach, S. (2011). 'Apple Bans App That Was Storing iPhone Passcodes ', SAI: Silicon Alley Insider, 15 June 2011, accessed 17 February 2012, <http://www.businessinsider.com/apple-bans-passcode-app-2011-6?op=1>

[5.22] Tom, (2009). 'Apple Rejects My Caricature App ', Tom's MAD Blog, 09 November 2009, accessed 17 February 2012, <http://www.tomrichmond.com/blog/2009/11/09/apple-rejects-my-caricature-app/>

[5.23] Fox News, . (2009). 'Apple Reverses Decision, Approves iPhone App Showing Political Caricatures', Fox News, 14 November 2009, accessed 17 February 2012, <http://www.foxnews.com/story/0,2933,575149,00.html>

[5.24] Wilson, M. (2008). 'Apple Yanks Another Popular App from iTunes, This Time Box Office ', Gizmodo, 04 August 2008, accessed 17 February 2012, <http://gizmodo.com/5032642/apple-yanks-another-popular-app-from-itunes-this-time-box-office>

[5.25] Reisinger, D. (2008). 'Why Apple and Facebook Need to Leave Apps Alone ', Mashable!, 05 August 2008, accessed 17 February 2012, <http://mashable.com/2008/08/04/apple-and-facebook-need-to-leave-apps-alone/>

[5.26] Foresman, C. (2008). 'Now Playing, formerly Box Office, is back on the App Store', Infinite Loop, 14 August 2008, accessed 17 February 2012, <http://arstechnica.com/journals/apple.ars/2008/08/14/now-playing-formerly-box-office-is-back-on-the-app-store>

[5.27] Sorrel, C. (2009). 'iPhone C64 Emulator Approved by Apple, Removed Again for Hiding Banned Feature', Gadget Lab, 08 September 2009, accessed 17 February 2012, <http://www.wired.com/gadgetlab/2009/09/iphone-c64-emulator-approved-by-apple-removed-again-for-hiding-banned-feature/>

[5.28] Topolsky, J. (2010). 'As Apple relaxes App Store rules, C64 emulator for iOS gets BASIC again', Engadget, 18 September 2010, accessed 17 February 2012, <http://www.engadget.com/2010/09/18/as-apple-relaxes-app-store-rules-c64-emulator-for-ios-gets-basi/>

[5.29] Reisinger, D. (2009). 'Behind 10 eyebrow-raising App Store rejections', The Digital Home, 26 August 2009, accessed 17 February 2012, <http://news.cnet.com/8301-13506_3-10317057-17.html>

[5.30] Mark, (2009). 'A Well-Timed Letter of Rejection ', Tapbots, 28 August 2009, accessed 17 February 2012, <http://tapbots.com/blog/app-store/a-well-timed-letter-of-rejection>

[5.31] Corynne, (2009). 'Apple Rejects EFF Updates App, Claims Parody Content Is Objectionable ', Electronic Frontier Foundation, 02 June 2009, accessed 17 February 2012, <http://www.eff.org/deeplinks/2009/06/oh-come-apple-reject>

[5.32] blog.montgomerie.net, (2009). 'Whither Eucalyptus?', blog.montgomerie.net, 21 May 2009, accessed 17 February 2012, <http://www.blog.montgomerie.net/whither-eucalyptus>

[5.33] Seff, J. (2009). 'Apple changes its mind, accepts Eucalyptus app', Macworld, 24 May 2009, accessed 17 February 2012, <http://www.macworld.com/article/140764/2009/05/eucalyptus.html>

[5.34] blog.montgomerie.net, (2009). 'Hither Eucalyptus!', blog.montgomerie.net, 24 May 2009, accessed 17 February 2012, <http://www.blog.montgomerie.net/hither-eucalyptus>

[5.35] Holland, J. (2009). 'EyeTV backdoor opens 3G TV streaming for iPhone', Electricpig, 16 October 2009, accessed 17 February 2012, <http://www.electricpig.co.uk/2009/10/16/eyetv-backdoor-opens-3g-tv-streaming-for-iphone/>

[5.36] Bean, J. (2009). 'EyeTV app removed from the App Store over streaming workaround ', TUAW, 17 October 2009, accessed 17 February 2012, <http://www.tuaw.com/2009/10/17/eyetv-app-removed-from-the-app-store-over-streaming-workaround/>

[5.37] Emigh, J. (2009). 'Apple Restores Banned 3G TV iPhone App', Computerworld, 19 October 2009, accessed 17 February 2012, <http://news.idg.no/cw/art.cfm?id=6E28EABD-1A64-6A71-CEED3F8AE380AB0A>

[5.38] Wauters, R. (2009). 'Financial Times Tweets That It Now Has An iPhone App. Yup, It's Pink. ', TechCrunch, 14 July 2009, accessed 17 February 2012, <http://www.techcrunch.com/2009/07/14/financial-times-tweets-that-it-now-has-an-iphone-app-yup-its-pink/>

[5.39] Wauters, R. (2011). 'FT Bypasses Apple's iTunes, Launches HTML5 Web App (Free Access First Week)', TechCrunch, 07 June 2011, accessed 17 February 2012, <http://techcrunch.com/2011/06/07/ft-bypasses-apples-itunes-launches-html5-web-app-free-access-first-week/>

[5.40] Andrews, R. (2011). 'Financial Times Apps Finally Pulled From iOS ', paidContent, 31 August 2011, accessed 17 February 2012, <http://paidcontent.org/article/419-apple-has-finally-pulled-financial-times-from-ios/>

[5.41] Ionescu, D. (2009). 'Google Makes iPhone the New eBook Reader: Watch Out, Kindle ', PC World, 06 February 2009, accessed 17 February 2012, <http://www.pcworld.com/article/159067/>

[5.42] Weintraub, S. (2011). 'WSJ, Kobo falling in line in Apple's App Store, Google Books is out ', 9to5Mac, 24 June 2011, accessed 17 February 2012, <http://9to5mac.com/2011/07/24/wsj-kobo-falling-in-line-in-apples-app-store-google-books-is-out/>

[5.43] Ionescu, D. (2011). 'Google Books Bows to New Apple App Store Rules', Daniel Ionescu / PC World, 26 June 2011, accessed 17 February 2012, <http://www.pcworld.com/article/236555/google_books_bows_to_new_apple_app_store_rules.html>

[5.44] Kincaid, J. (2009). 'Apple Is Growing Rotten To The Core: Official Google Voice App Blocked From App Store ', TechCrunch, 27 July 2009, accessed 17 February 2012, <http://www.techcrunch.com/2009/07/27/apple-is-growing-rotten-to-the-core-and-its-likely-atts-fault/>

[5.44] Kincaid, J. (2009). 'Apple Is Growing Rotten To The Core: Official Google Voice App Blocked From App Store ', TechCrunch, 27 July 2009, accessed 17 February 2012, <http://www.techcrunch.com/2009/07/27/apple-is-growing-rotten-to-the-core-and-its-likely-atts-fault/>

[5.45] Kincaid, J. (2009). 'FCC Takes On Apple And AT&T Over Google Voice Rejection ', TechCrunch, 31 July 2009, accessed 17 February 2012, <http://www.techcrunch.com/2009/07/31/fcc-takes-on-apple-and-att-over-google-voice-rejection/>

[5.46] Needleman, R. (2010). 'Hands-on with Google Voice for the iPhone ', CNET News, 26 January 2010, accessed 17 February 2012, <http://news.cnet.com/8301-19882_3-10440921-250.html>

[5.47] Dolcourt, J. (2010). 'Hands-On: Google Voice for iPhone still slightly underbaked ', Crave: The gadget blog, 16 November 2010, accessed 17 February 2012, <http://reviews.cnet.com/8301-19512_7-20022984-233.html>

[5.48] Kincaid, J. (2009). 'Apple Yanks The Cord On GV Mobile. ', TechCrunch, 27 July 2009, accessed 17 February 2012, <http://www.techcrunch.com/2009/07/27/apple-yanks-the-cord-on-gv-mobile-is-it-trying-to-kill-google-voice-on-the-iphone/>

[5.49] Boy Genius Report, B. (2009). 'GV Mobile 2.0 first look ', Boy Genius Report, 27 December 2009, accessed 17 February 2012, <http://www.boygeniusreport.com/2009/12/26/gv-mobile-2-0-first-look/>

[5.50] Kincaid, J. (2010). 'Google Voice On The iPhone Comes Full Circle As Apple Accepts GV Mobile+', TechCrunch, 19 September 2010, accessed 17 February 2012, <http://techcrunch.com/2010/09/18/google-voice-iphone-2/>

[5.51] Siegler, M. (2009). 'Yep, iPorn Is [Was] Here For The iPhone ', TechCrunch, 25 June 2009, accessed 17 February 2012, <http://www.techcrunch.com/2009/06/24/yep-iporn-is-here-for-the-iphone/>

[5.52] Wauters, R. (2009). 'NoPorn: Apple Removes “Hottest Girls” From The App Store ', TechCrunch, 25 June 2009, accessed 17 February 2012, <http://www.techcrunch.com/2009/06/25/noporn-apple-removes-hottest-girls-the-app-store/>

[5.53] Frommer, D. (2008). 'Apple's iPhone-App-Approval Mouse Falls Off Treadmill: Buy The $1000 App That Does Nothing (AAPL) ', Silicon Alley Insider, 06 August 2008, accessed 17 February 2012, <http://www.alleyinsider.com/2008/8/apple-s-iphone-app-approval-mouse-falls-off-treadmill-buy-the-1000-app-that-does-nothing-aapl->

[5.54] Frommer, D. (2008). 'Worthless, $1000 “I Am Rich” iPhone App Disappears (AAPL)', Silicon Alley Insider, 06 August 2008, accessed 17 February 2012, <http://www.alleyinsider.com/2008/8/worthless-1000-i-am-rich-iphone-app-disappears>

[5.55] Gruber, J. (2010). 'PDF Security Exploit Allows ‘JailbreakMe’ Website to Jailbreak iPhones and iPad Over the Web ', Daring Fireball, 02 August 2010, accessed 17 February 2012, <http://daringfireball.net/linked/2010/08/02/jailbreakme>

[5.56] Asad, T. (2010). 'Apple Releases iOS 4.0.2 for iPhone, iOS 3.2.2 for iPad to Patch JailbreakMe PDF Exploit ', Redmond Pie, 11 August 2010, accessed 17 February 2012, <http://www.redmondpie.com/download-ios-4.0.2-for-iphone-4-3gs-ipod-touch-ipad-3.2.2/>

[5.57] Goodin, D. (2010). 'Jailbreak hole in iOS 4.1 will be hard to close ', The Register, 09 September 2010, accessed 17 February 2012, <http://www.theregister.co.uk/2010/09/09/ios_4_dot_1_jailbreak/>

[5.58] McGann, L. (2010). 'Mark Fiore can win a Pulitzer Prize, but he can't get his iPhone cartoon app past Apple's satire police ', Nieman Journalism Lab, 15 April 2010, accessed 17 February 2012, <http://www.niemanlab.org/2010/04/mark-fiore-can-win-a-pulitzer-prize-but-he-cant-get-his-iphone-cartoon-app-past-apples-satire-police/>

[5.59] Valentino-DeVries, J. (2010). 'Cartoonist: Apple Backs Down After Denying iPhone App', Digits, 16 April 2010, accessed 17 February 2012, <http://blogs.wsj.com/digits/2010/04/16/cartoonist-apple-backs-down-after-denying-iphone-app/>

[5.60] Stelter, B. (2010). 'Steve Jobs Says Apple Made a Mistake in Rejecting Pulitzer Winner's App ', Media Decoder, 16 April 2010, accessed 17 February 2012, <http://mediadecoder.blogs.nytimes.com/2010/04/16/steve-jobs-says-apple-made-a-mistake-in-rejecting-pulitzer-winners-app/>

[5.61] Frommer, D. (2009). 'Apple Rejects Another iPhone App: Religious Photo Parodies ‘Objectionable’ ', Silicon Alley Insider, 11 May 2009, accessed 17 February 2012, <http://www.eff.org/deeplinks/2009/06/oh-come-apple-reject>

[5.62] Frommer, D. (2009). 'Apple Rejects Another iPhone App: Religious Photo Parodies ‘Objectionable’ ', Silicon Alley Insider, 11 May 2009, accessed 17 February 2012, <http://www.businessinsider.com/apple-rejects-another-iphone-app-religious-photo-parodies-objectionable-2009-5>

[5.63] Computerworld, (2009). 'Apple approves, then pulls Nintendo emulator for iPhonewrite this up!” ', Computerworld, 22 December 2009, accessed 17 February 2012, <http://news.idg.no/cw/art.cfm?id=B80DBDB0-1A64-67EA-E42378A6BB8F212F>

[5.64] Arn, (2008). 'Nullriver Introduces 3G/EDGE Tethering App for iPhone [Updatedx4] ', MacRumors, 31 July 2008, accessed 17 February 2012, <http://www.macrumors.com/2008/07/31/nullriver-introduces-3g-edge-tethering-app-for-iphone/>

[5.65] Arn, (2008). 'NetShare iPhone Tethering App Reappears in App Store ', MacRumors, 01 August 2008, accessed 17 February 2012, <http://www.macrumors.com/2008/08/01/netshare-tethering-app-reappears-on-app-store/>

[5.66] Gruber, J. (2009). 'Ninjawords: iPhone Dictionary, Censored by Apple ', Daring Fireball, 04 August 2009, accessed 17 February 2012, <http://daringfireball.net/2009/08/ninjawords>

[5.67] Gruber, J. (2009). 'Phil Schiller Responds Regarding Ninjawords and the App Store ', Daring Fireball, 06 August 2009, accessed 17 February 2012, <http://daringfireball.net/2009/08/phil_schiller_app_store>

[5.68] Kingsley-Hughes, A. (2008). 'No Opera Mini for the iPhone', Hardware 2.0, 31 October 2008, accessed 17 February 2012, <http://blogs.zdnet.com/hardware/?p=2901>

[5.69] Wauters, R. (2010). 'Surprise, Surprise: Opera Mini iPhone App Gets Apple's Stamp Of Approval', TechCrunch, 12 April 2010, accessed 17 February 2012, <http://techcrunch.com/2010/04/12/surprise-surprise-opera-mini-iphone-app-gets-apples-stamp-of-approval/>

[5.70] Kafasis, P. (2008). 'A Bridge Too Far ', O'Reilly Media, 14 September 2008, accessed 17 February 2012, <http://blogs.oreilly.com/iphone/2008/09/a-bridge-too-far.html>

[5.71] Perez, S. (2008). 'Podcaster Developer Uses Little-Known “Ad Hoc” Mode To Distribute Banned iPhone App ', ReadWriteWeb, 15 September 2008, accessed 17 February 2012, <http://www.readwriteweb.com/archives/podcaster_developer_uses_little_known_ad_hoc_to_distribute_banned_app.php>

[5.72] Dan, (2009). 'RSS Player: Podcaster Reborn and ACCEPTED Into The App Store - UPDATED ', UNEASYsilence, 27 January 2009, accessed 17 February 2012, <http://uneasysilence.com/archive/2009/01/13783/>

[5.73] Arn, (2008). 'Apple Rejecting Apps Based on ‘Limited Utility’ ', MacRumors iPhone Blog, 04 September 2008, accessed 17 February 2012, <http://www.macrumors.com/iphone/2008/09/04/apple-rejecting-applications-based-on-limited-utility/>

[5.74] Arn, (2008). 'Apple Broadens App Store Acceptance, Approves ‘Pull My Finger’', MacRumors, 12 December 2008, accessed 17 February 2012, <http://www.macrumors.com/2008/12/12/apple-broadens-app-store-acceptance-approves-pull-my-finger/>

[5.75] Swisher, K. (2010). 'Pulse iPad App Gets Steve Jobs's Praise in Morning...Then Booted From App Store Hours Later After NYT Complains', BoomTown, 08 June 2010, accessed 17 February 2012, <http://kara.allthingsd.com/20100608/popular-pulse-news-reader-ipad-app-gets-steve-jobs-praise-in-morning-then-booted-from-app-store-hours-later-after-new-york-times-complaint/>

[5.76] Swisher, K. (2010). 'Meet the Two Grad Students Who Freaked Out the NYT-The Pulse iPad App Creators Speak!', BoomTown, 09 June 2010, accessed 17 February 2012, <http://kara.allthingsd.com/20100608/meet-the-two-grad-students-who-freaked-out-the-nyt-the-pulse-ipad-app-creators-speak/>

[5.77] Siegler, M. (2011). 'Apple Smacks Readability In The Face With Subscription Rules; All SaaS In Trouble', TechCrunch, 21 February 2011, accessed 17 February 2012, <http://techcrunch.com/2011/02/21/readability-app-rejection/>

[5.78] Siegler, M. (2011). 'As Apple Ponders Their Subscription Ruling, Readability Goes Full HTML5 ', TechCrunch, 10 March 2011, accessed 17 February 2012, <http://techcrunch.com/2011/03/09/readability-html5/>

[5.79] Ha, A. (2009). 'Apple kills Routesy app, my iPhone gets less useful ', VentureBeat, 27 June 2009, accessed 17 February 2012, <http://digital.venturebeat.com/2009/06/27/apple-kills-routesy-app-my-iphone-gets-less-useful/>

[5.80] Toto, S. (2010). '“Problematic Wi-Fi Access”: Apple Bans Augmented Reality App Sekai Camera [Update]', TechCrunch, 04 March 2010, accessed 17 February 2012, <http://techcrunch.com/2010/03/04/tonchidot-sekai/>

[5.81] Kahney, L. (2010). 'Apple Quietly Reinstates Banned Bikini Shopping App', Cult of Mac, 23 February 2010, accessed 17 February 2012, <http://www.cultofmac.com/apple-quietly-reinstates-banned-bikini-shopping-app/31388>

[5.82] some blog, (2009). 'Someecards iPhone app rejected for making fun of celebrities? ', some blog, 07 October 2009, accessed 17 February 2012, <http://someecards.wordpress.com/2009/10/07/someecards-iphone-app-rejected-for-making-fun-of-celebrities/>

[5.83] Siegler, M. (2009). 'Apple Has No Sense Of Humor. Luckily, Google Does. ', TechCrunch, 08 November 2009, accessed 17 February 2012, <http://www.techcrunch.com/2009/11/07/apple-has-no-sense-of-humor-luckily-google-does/>

[5.84] Wauters, R. (2010). 'Apple Demands Removal Of USB Sharing Feature In Stanza iPhone App ', TechCrunch, 02 February 2010, accessed 17 February 2012, <http://www.techcrunch.com/2010/02/02/apple-stanza-usb/>

[5.85] Carthy, R. (2010). 'Tawkon Releases BlackBerry Radiation App, Visits Apple To Discuss Rejection ', TechCrunch, 17 May 2010, accessed 17 February 2012, <http://techcrunch.com/2010/05/17/tawkon/>

[5.86] Conneally, T. (2011). 'Steve Jobs says ‘No Interest’ in iPhone 4 radiation app, devs release it anyway ', BetaNews, 23 March 2011, accessed 17 February 2012, <http://www.betanews.com/article/Steve-Jobs-says-No-Interest-in-iPhone-4-radiation-app-devs-release-it-anyway/1300885402>

[5.87] Zibreg, C. (2011). 'After 201 days in review, Apple greenlights iPhone app that unmasks blocked calls (updated with video)', 9 to 5 Mac, 04 April 2011, accessed 17 February 2012, <http://www.9to5mac.com/59223/after-201-days-in-review-apple-greenlights-iphone-app-that-unmasks-blocked-calls/>

[5.88] Smw, (2009). 'Trillian for iPhone: Two months later, still no response from Apple...', Cerulean Studios' Blog, 13 October 2009, accessed 17 February 2012, <http://blog.ceruleanstudios.com/?p=757>

[5.89] Smw, (2009). 'Trillian for iPhone: Now available on the App Store!', Cerulean Studios' Blog, 19 November 2009, accessed 17 February 2012, <http://blog.ceruleanstudios.com/?p=815>

[5.90] Siegler, M. (2009). 'Apple rejects an app for offensive words on Twitter ', VentureBeat, 10 March 2009, accessed 17 February 2012, <http://venturebeat.com/2009/03/10/apple-rejects-an-app-for-offensive-words-in-twitter-trends/>

[5.91] Siegler, M. (2009). 'Apple backs down on Tweetie app. Swear words for all! ', VentureBeat, 10 March 2009, accessed 17 February 2012, <http://venturebeat.com/2009/03/10/apple-backs-down-tweetie-with-swear-words-for-all/>

[5.92] Kevin, (2009). 'There's No App for That', Riverturn Blog and Talk Back, 29 July 2009, accessed 17 February 2012, <http://www.riverturn.com/blog/?p=455>

[5.93] Schonfeld, E. (2009). 'Google Voice Is Coming Back To The iPhone Via The Browser, Thanks To VoiceCentral', TechCrunch, 24 December 2009, accessed 17 February 2012, <http://www.techcrunch.com/2009/12/24/voicecentral-iphone-browser/>

[5.94] Siegler, M. (2009). 'Apple Continues To Right App Store Wrongs. Obama “Hope” App Is Go. ', TechCrunch, 21 August 2009, accessed 17 February 2012, <http://www.techcrunch.com/2009/08/21/apple-continues-to-right-app-store-wrongs-obama-hope-app-is-go/>

[5.95] Perez, S. (2008). 'Apple Approves Smutty iPhone App By Mistake ', ReadWriteWeb, 24 November 2008, accessed 17 February 2012, <http://www.readwriteweb.com/archives/apple_approves_smutty_iphone_app_by_mistake.php>

[5.96] Ricker, T. (2010). 'Wi-Fi Sync app rejected by Apple, headed to Cydia for $9.99', Engadget, 14 May 2010, accessed 17 February 2012, <http://www.engadget.com/2010/05/14/wi-fi-sync-rejected-headed-to-cydia-for-9-99/>

Bibliography

Abbott, A. "A primer on sequence methods," *Organization science* (1:4) 1990, pp 375-392.

Abbott, A. "From causes to events," *Sociological Methods & Research* (20:4) 1992, p 428.

Abbott, A., and Hrycak, A. "Measuring resemblance in sequence data: An optimal matching analysis of musicians' careers," *American journal of sociology* (96:1) 1990, pp 144-185.

Abell, P. *The syntax of social life: The theory and method of comparative narratives* Oxford, 1987.

Abell, P. "Narrative explanation: an alternative to variable-centered explanation?," *Annual Review of Sociology* (30:1) 2004, pp 287-310.

Abernathy, W. J., and Utterback, J. M. "Patterns of Industrial Innovation," *Technology Review* (80:7) 1978, pp 40-47.

Adomavicius, G., Bockstedt, J., Gupta, A., and Kauffman, R. "Making Sense of Technology Trends in the Information Technology Landscape: A Design Science Approach," *MIS Quarterly* (32:4) 2008, pp 779-809.

Allen, G. N., Burk, D. L., and Davis, G. B. "Academic data collection in electronic environments: Defining acceptable use of Internet resources," *Management Information Systems Quarterly* (30:3) 2006, p 4.

Anderson, P., and Tushman, M. L. "Technological Discontinuities and Dominant Design: A Cyclical Model of Technological Change," *Administrative Science Quarterly* (35) 1990, pp 604-633.

Apple "iTunes Connect Developer Guide 7.3," Apple Inc., 2012.

Arthur, W. B. "Competing technologies, increasing returns, and lock-in by historical events," *The Economic Journal* (89) 1989, pp 116-131.

Bakhtin, M. "The Dialogic Imagination.(Caryl Emerson and Michael Holquist, Trans.)," Austin, TX: University of Texas Press, 1981.

Bal, M. *Narratology: Introduction to the theory of narrative* University of Toronto Press, 1985.

Baldwin, C. Y., and Clark, K. B. "Managing in the Age of Modularity," *Harvard Business Review* (75:5) 1997, pp 84-93.

Baldwin, C. Y., and Clark, K. B. *Design Rules: The Power of Modularity, Volume 1* MIT Press, 2000.

Baldwin, C. Y., and Clark, K. B. "Architectural innovation and dynamic competition: The smaller “footprint” strategy," Harvarrd University, Boston, 2006.

Baldwin, C. Y., and Woodard, C. J. "The Architecture of Platforms: A Unified View," in: *Platforms, Markets and Innovation,* A. Gawer (ed.), Edward Elgar, Cheltenham, 2009, pp. 19-44.

Bar-Ilan, J. "Information hub blogs," *Journal of information science* (31:4) 2005, p 297.

Barnett, B. "Perfect mother or artist of obscenity? Narrative and myth in a qualitative analysis of press coverage of the Andrea Yates murders," *Journal of Communication Inquiry* (29:1) 2005, p 9.

Barthes, R. "Introduction to the structural analysis of narratives.," in: *Image-music-text,* S. Heath (ed.), Fontana, 1977, pp. 79-124.

Bartis, E., and Mitev, N. "A multiple narrative approach to information systems failure: a successful system that failed," *European Journal of Information Systems* (17:2) 2008, pp 112-124.

Bauer, M., and Gaskell, G. *Qualitative Researching with Text, Image and Sound: A Practical Handbook for Social Research* Sage Publications Ltd, 2000.

Beniger, J. R. *The Control Revolution: Technological and Economic Origins of the Information Society* Harvard University Press, 1986.

Benkler, Y. *The wealth of networks : how social production transforms markets and freedom* Yale University Press, New Haven, 2006, pp. xii, 515 p.

Blood, R. "How blogging software reshapes the online community," *Communications of the ACM* (47:12) 2004, pp 53-55.

Boje, D. M. "The storytelling organization: A study of story performance in an office-supply firm," *Administrative Science Quarterly* (36:1) 1991, pp 106-126.

Boudreau, K. "The boundaries of the platform: Vertical integration and economic incentives in mobile computing," MIT, Boston, 2005.

Boudreau, K. "Open Platform Strategies and Innovation: Granting Access vs. Devolving Control," *Management Science* (56:10) 2010, pp 1849-1872.

Boudreau, K. J., and Hagiu, A. "Platform rules: multi-sided platforms as regulators," in: *Platforms, Markets and Innovation,* A. Gawer (ed.), Edward Elgar Publishing Limited, Cheltenham, UK, 2009, pp. 163-191.

Bourdieu, P., and Nice, R. *Outline of a Theory of Practice* Cambridge University Press, 1977.

Brown, A. D. "Narrative, politics and legitimacy in an IT implementation.," *Journal of Management Studies*:35) 1998, pp 35-58.

Burke, K. *A grammar of motives* Univ of California Press, 1969.

Campagnolo, D., and Camuffo, A. "The concept of modularity in management studies: a literature review," *International Journal of Management Reviews* (12:3) 2010, pp 259-283.

Chandler, D. *Semiotics: the basics* Taylor & Francis, 2007.

Chesbrough, H. W. *Open innovation: The new imperative for creating and profiting from technology* Harvard Business Press, 2003.

Ching, W. K., and Ng, M. K. *Markov Chains: models, algorithms and applications* Springer-Verlag New York Inc, 2006.

Chomsky, N., and Miller, G. A. "Finite state languages," *Information and control* (1:2) 1958, pp 91-112.

Choudhury, V., and Sabherwal, R. "Portfolios of control in outsourced software development projects," *Information Systems Research* (14:3) 2003, pp 291-314.

Clark, K. B. "The Interaction of Design Hierarchies and Market Concepts in Technological Evolution," *Research Policy* (14:5) 1985, pp 235-251.

Clegg, S. *Frameworks of power* Sage Publications Ltd, 1989.

Corsaro, W., and Heise, D. "Event structure models from ethnographic data," *Sociological methodology* (20:1) 1990, pp 1-57.

Courtés, J., Quémada, B., and Rastier, F. "Analyse sémiotique du discours," *Paris, Hachette*) 1991.

Czarniawska-Joerges, B. *Narrating the organization: Dramas of institutional identity* University of Chicago Press, 1997.

Czarniawska-Joerges, B. *A narrative approach to organization studies* Sage Publications, Inc, 1998.

Dalcher, D. "Stories and histories: Case study research (and beyond) in information systems failures," in: *The handbook of information systems research,* M.E. Whitman and A.B. Woszczynski (eds.), IGI Publishing, 2004, pp. 305-322.

Davidson, E., and Vaast, E. "Tech talk: An investigation of blogging in technology innovation discourse," *Professional Communication, IEEE Transactions on* (52:1) 2009, pp 40-60.

Dunteman, G. H. *Principal components analysis* SAGE publications, Inc, 1989.

Eaton, B., Elaluf-Calderwood, S., Sorensen, C., and Yoo, Y. "The Dynamics of Digital Service System Innovation: Control and Generativity in Apple iOS," *(Submitted to an international journal: 2nd round)*) 2012.

Eaton, B., Elaluf-Calderwood, S., Sørensen, C., and Yoo, Y. "Dynamic Structures of Control and Generativity in Digital Ecosystem Service Innovation: The Cases of the Apple and Google Mobile App Stores," in: *Information Systems and Innovation Group Working Paper Series*, The London School of Economics and Political Science, 2011.

Eisemann, T. R., Parker, G., and Van Alstyne, M. "Opening Platforms: how, when and why?," in: *Platforms, Markets and Innovation,* A. Gawer (ed.), Edward Elgar Publishing Limited, Cheltenham, UK, 2009, pp. 131-162.

Eisenhardt, K. M. "Control: Organizational and economic approaches," *Management Science* (31:2) 1985, pp 134-149.

Etzioni, A. "Political unification: A comparative study of leaders and forces," RE Krieger Pub. Co.(Huntington, NY), 1965.

Fagerberg, J., Mowery, D. C., and Nelson, R. R. *The Oxford handbook of innovation* Oxford University Press, USA, 2006.

Farjoun, M. "Beyond dualism: Stability and change as a duality," *The Academy of Management Review (AMR)* (35:2) 2010, pp 202-225.

Farrell, J., and Katz, M. L. "Innovation, rent extraction, and integration in systems markets," *The journal of industrial economics* (48:4) 2000, pp 413-432.

Faulkner, P., and Runde, J. "The Social, the material, and the ontology of non-material technological objects," *Working Paper*) 2010.

Fiol, C. M. "A semiotic analysis of corporate language: Organizational boundaries and joint venturing," *Administrative Science Quarterly* (34:2) 1989, pp 277-303.

Foucault, M., and Gordon, C. *Power/knowledge: Selected interviews and other writings, 1972-1977* Pantheon, 1980.

Gao, L. S., and Iyer, B. "Analyzing complementarities using software stacks for software industry acquisitions," *Journal of Management Information Systems* (23:2) 2006, pp 119-147.

Gawer, A. "Platform Dynamics and Strategies: from Products to Service," in: *Platforms, Markets and Innovation,* A. Gawer (ed.), Edward Elgar Publishing Limited, Cheltenham, UK, 2009a, pp. 45-76.

Gawer, A. "Platforms, Markets and Innovation: An Introduction," in: *Platforms, Markets and Innovation,* A. Gawer (ed.), Edward Elgar Publishing, Inc., Cheltenham, UK, 2009b, pp. 1 - 16.

Gawer, A., and Cusumano, M. A. *Platform leadership : how Intel, Microsoft, and Cisco drive industry innovation* Harvard Business School Press, Boston, Mass., 2002, pp. xiv, 305 p.

Gawer, A., and Cusumano, M. A. "How Companies Become Platform Leaders," *MITSloan Management Review* (49:2) 2008, pp 28-35.

Gawer, A., and Henderson, R. "Platform Owner Entry and Innovation in Complementary Markets: Evidence from Intel," *Journal of Economics & Management Strategy* (16:1), Spring 2007 2007, pp 1-34.

Gersick, C. J. G. "REVOLUTIONARY CHANGE THEORIES: A MULTILEVEL EXPLORATION OF THE PUNCTUATED EQUILIBRIUM PARADIGM," *The Academy of Management Review* (16:1) 1991, pp 10-36.

Ghazawneh, A., and Henfridsson, O. "Governing third-party development through platform boundary resources,") 2010.

Ghazawneh, A., and Henfridsson, O. "Balancing Platform Control and External Contribution in Third-Party Development: The Boundary Resources Model," *Information Systems Journal* (22:2) 2012, p Forthcoming.

Gibbs, J. P. *Control: sociology's central notion* Univ of Illinois Press, 1989.

Gibson, C., and Birkinshaw, J. "The Antecedents, Consequences, and Mediating Role of Organizational Ambidexterity," *Academy of Management Journal* (47:2) 2004, pp 209-226.

Gordon, T. J., and Greenspan, D. "Chaos and fractals: New tools for technological and social forecasting," *Technological Forecasting and Social Change* (34:1) 1988, pp 1-25.

Gregor, S. "The nature of theory in information systems," *MIS Quarterly* (30:3) 2006, pp 611-642.

Greimas, A., and Rastier, F. "The Interaction of Semiotic Constraints," in: *Yale French Studies*, Yale University Press, 1968, pp. 86-105.

Greimas, A. J. *On meaning: Selected writings in semiotic theory* F. Pinter, 1987.

Greimas, A. J. *Maupassant: The semiotics of text* John Benjamins Publishing Company, 1988.

Hanseth, O., and Lyytinen, K. "Design Theory for Dynamic Complexity in Information Infrastructures: The Case of Building Internet," *Journal of Information Technology* (25:1) 2010, pp 1-19.

Hanseth, O., Monteiro, E., and Hatling, M. "Developing information infrastructure: The tension between standardization and flexibility," *Science, technology & human values* (21:4) 1996, p 407.

Haraway, D. "The promises of monsters: a regenerative politics for inappropriate/d others," in: *Cultural studies,* L. Grossberg, C. Nelson and P. Treichler (eds.), Routledge, 1992, pp. 295-337.

Hargrave, T. J., and Van de Ven, A. H. "A collective action model of institutional innovation," *The Academy of Management Review ARCHIVE* (31:4) 2006, pp 864-888.

Harris, M. L., Collins, R. W., and Hevner, A. R. "Control of flexible software development under uncertainty," *Infomation Systems Research* (20:3) 2009, pp 400-419.

Hart, O., and Moore, J. "Property Rights and the Nature of the Firm," *Journal of Political Economy*) 1990, pp 1119-1158.

Hébert, L. *Dispositifs pour l'analyse des textes et des images* Pulim, Limoges, 2007.

Henderson, R. M., and Clark, K. B. "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms," *Administrative Science Quaterly* (35:1), March 1990, pp 9-30.

Hendricks, W. O. "The structural study of narration: Sample analyses," *Poetics* (1:3) 1972, pp 100-123.

Hine, C. (ed.) *Virtual Methods: Issues in Social Research on the Internet*. Berg Publishers, Oxford, 2005.

Holmstrom, B. "The firm as a subeconomy," *Journal of Law, Economics, and Organization* (15:1) 1999, pp 74-102.

Hookway, N. "Entering the blogosphere': some strategies for using blogs in social research," *Qualitative Research* (8:1) 2008, p 91.

Huber, G. P., and Van de Ven, A. H. *Longitudinal field research methods: Studying processes of organizational change* Sage Publications, Inc, 1995.

Iansiti, M., and Levien, R. "Strategy as ecology," *Harvard Business Review* (82:3) 2004, pp 68-81.

Jackson, B. S. *Semiotics and legal theory* Routledge, 1987.

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

Jakobson, R. "Closing statement: Linguistics and poetics," in: *Style in langage,* T.A. Sebeok (ed.), Massachusetts Institute of Techonology, 1960.

Jameson, F. *The political unconscious: Narrative as a socially symbolic act* Routledge, 1983.

Kallinikos, J., Aaltonen, A., and Marton, A. "The Distinct Ontology of Digital Objects: A New Frontier for IS Research,") forthcoming.

Kane, G. C., and Fichman, R. G. "The shoemaker's children: using Wikis for information systems teaching, research, and publication (includes comments and editorial note)," *Management Information Systems Quarterly* (33:1) 2009, p 3.

Katz, M. L., and Shapiro, C. "Technology Adoption in the Presence of Network Externalities," *Journal of Political Economy* (94:4) 1986.

Katz, M. L., Shapiro C. "Systems Competition and Network Effects," *Journal of Economic Perspectives* (8) 1994.

Kirsch, L. J. "Portfolios of control modes and IS project management," *Information Systems Research* (8:3) 1997, pp 215-239.

Kirsch, L. J. "Deploying common systems globally: The dynamics of control," *Information Systems Research* (15:4) 2004, pp 374-395.

Kling, R., and Iacono, S. "Computing as an occasion for social control," *Journal of Social Issues* (40:3) 1984, pp 77-96.

Kozinets, R. V. *Netnography: Doing ethnographic research online* Sage Publications Ltd, 2009.

Langley, A. "Strategies for theorizing from process data," *Academy of Management Review*) 1999, pp 691-710.

Langlois, R. N. "External economies and economic progress: The case of the microcomputer industry," *The Business History Review* (66:1) 1992, pp 1-50.

Langlois, R. N. "Modularity in Technology and Organization," *Journal of Economic Behavior & Organization* (49:1) 2002, pp 19-37.

Langlois, R. N. "The vanishing hand: The changing dynamics of industrial capitalism," *Industrial and Corporate Change* (12:2) 2003, pp 351-385.

Langlois, R. N., and Robertson, P. "Networks and Innovation in a Modular System: Lessons from the Microcomputer and Stereo Component Industries," *Research Policy* (21) 1992, pp 297-313.

Lessig, L. *The future of ideas : the fate of the commons in a connected world*, (1st Vintage Books ed.) Vintage Books, New York, 2002, pp. xxiv, 352 p.

Lessig, L. *Code : version 2.0*, ([2nd ed.) Basic Books, New York, 2006, pp. xvii, 410 p.

Lukes, S. *Power: A radical view* Macmillan, London, 1981.

Malone, T. W., Crowston, K., Lee, J., Pentland, B., Dellarocas, C., Wyner, G., Quimby, J., Osborn, C. S., Bernstein, A., and Herman, G. "Tools for inventing organizations: Toward a handbook of organizational processes," *Management Science* (45:3) 1999, pp 425-443.

Manning, C. D., Schütze, H., and MITCogNet *Foundations of statistical natural language processing* MIT Press, 1999.

Mattson, T., and Davidson, E. "The A-List vs. the Long Tail: Technology Bloggers and Reputation," *AMCIS 2009 Proceedings*) 2009, p 574.

Mautner, G. "Time to get wired: Using web-based corpora in critical discourse analysis," *Discourse & Society* (16:6) 2005, p 809.

Myers, D. "Computer game semiotics," *Play and Culture* (4:4) 1991.

Myers, M. "Hermeneutics in Information Systems Research," in: *Social Theory and Philosophy for Information Systems,* J. Mingers and W. L.P. (eds.), John Wiley & Sons, 2004.

Myers, M., and Avison, D. *Qualitative Research in Information Systems: A Reader* Sage Publications Ltd, 2002.

Nelson, R. R., and Winter, S. G. "In search of useful theory of innovation," *Research Policy* (6:1) 1977, pp 36-76.

Nelson, R. R., and Winter, S. G. *An evolutionary theory of economic change* Belknap press, 1982.

Nidumolu, S. R., and Subramani, M. R. "The matrix of control: Combining process and structure approaches to managing software development," *Journal of Management Information Systems* (20:3) 2003, pp 159-196.

O'Connor, P. "User-generated content and travel: A case study on Tripadvisor. com," *Information and Communication Technologies in Tourism 2008*) 2008, pp 47-58.

Olson, D. L. "Ethical aspects of web log data mining," *International Journal of Information Technology and Management* (7:2) 2008, pp 190-200.

Orlikowski, W. J., and Scott, S. "Sociomateriality: Challenging the Separation of Technology, Work and Organization," Department of Management. London School of Economics and Political Science, London, UK, p. 47.

Ouchi, W. G. "The relationship between organizational structure and organizational control," *Administrative Science Quarterly* (22:1) 1977, pp 95-113.

Ouchi, W. G. "The transmission of control through organizational hierarchy," *Academy of Management Journal* (21:2) 1978, pp 173-192.

Ouchi, W. G. "A conceptual framework for the design of organizational control mechanisms," *Management Science* (25:9) 1979, pp 833-848.

Ouchi, W. G., and Maguire, M. A. "Organizational control: Two functions," *Administrative Science Quarterly* (20:4) 1975, pp 559-569.

Parker, G., and Alstyne, M. V. "Innovation, Openness & Platform control," in: *Proceedings of the 11th ACM conference on Electronic commerce*, ACM, Cambridge, Massachusetts, USA, 2010, pp. 95-96.

Parker, G. G., and Van Alstyne, M. W. "Two-Sided Network Effects: A Theory of Information Product Design," *Management Science* (51:10) 2005, pp 1494-1504.

Parnas, D. L. "On The Criteria To Be Used in Decomposing Systems into Modules " *Comm. ACM* (15 12), December 1972, pp 1053-1058

Pentland, B. "Building process theory with narrative: From description to explanation," *The Acadamy of Management Review* (24:4) 1999a, p 711.

Pentland, B., and Reuter, H. "Organisational routines and grammars of action," *Administrative Science Quarterly* (39) 1994, pp 494-510.

Pentland, B. T. "Organizing moves in software support hot lines," *Administrative Science Quarterly* (37:4) 1992, pp 527-548.

Pentland, B. T. "Grammatical models of organizational processes," *Organization science*) 1995, pp 541-556.

Pentland, B. T. "Organizations as networks of action," in: *Variations in organization science: in honor of Donald T. Campbell,* S.K. Hall (ed.), 1999b, p. 237.

Pentland, B. T., and Feldman, M. S. "Narrative Networks: Patters of Technology and Organization," *Organization Science* (18:5), September-October 2007 2007, pp 781-795.

Pentland, B. T., and Feldman, M. S. "Designing routines: On the folly of designing artifacts, while hoping for patterns of action," *Information and Organization* (18:4) 2008, pp 235-250.

Pentland, B. T., Hærem, T., and Hillison, D. "Comparing Organizational Routines as Recurrent Patterns of Action," *Organization Studies* (31:7) 2010, p 917.

Pentland, B. T., Haerem, T., and Hillison, D. "The (N) Ever-Changing World: Stability and Change in Organizational Routines," *Organization science*) 2011, p orsc. 1110.0624 v1111.

Pfeffer, J., and Salancik, G. R. *The external control of organizations: A resource dependence perspective* Stanford Business Books, 2003.

Poole, M. S., and Van de Ven, A. H. "Using Paradox to Build Management and Organization Theories," *Academy of Management Review* (14:4) 1989, pp 562-578.

Propp, V. I. A. *Morphology of the Folktale* Univ of Texas Press, 1968.

Puri, A. "The web of insights: The art and practice of webnography," *International journal of market research* (49:3) 2007, p 387.

Riessman, C. K. *Narrative analysis* Sage Publ., 1994.

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

Sabherwal, R., and Robey, D. "An empirical taxonomy of implementation processes based on sequences of events in information system development," *Organization science* (4:4) 1993, pp 548-576.

Saltzer, J. H., Reed, D. O., and Clark, D. D. "End-to-End Arguments in Systems Design," *ACM Transactions on Computer Systems* (2:4), November 1984 1984, pp 277-288.

Schilling, M. A. "Toward a General Modular System Theory and its Application to Interfirm Product Modularity," *Academy of Management Review* (25:2) 2000, pp 312-334.

Schilling, M. A. "Technological Leapfrogging: LESSONS FROM THE US VIDEO GAME CONSOLE INDUSTRY," *California Management Review* (45:3) 2003, pp 6-32.

Schilling, M. A. "Protecting or Diffusing a Technology Platforms: Tradeoffs in Appropriability, Network Externalities, and Architectural Control," in: *Platforms, Markets and Innovation,* A. Gawer (ed.), Edward Elgar Publishing, Ltd., Cheltenham, UK, 2009, pp. 192-218.

Schoneboom, A. "Diary of a working boy," *Ethnography* (8:4) 2007, p 403.

Silverman, D. *Doing Qualitative Research* Sage Publications Ltd, 2009.

Simon, H. A. "The Architecture of Complexity," *Proceedings of the American philosophical society* (106:6) 1962, pp 467-482.

Strahilevitz, L. J. "Information Asymmetries and the Rights to Exclude," *Michigan Law Review* (104:8) 2006, pp 1835-1898.

Svahn, F., and Henfridsson, O. "The Dual Regimes of Digital Innovation Management," Hawaii International Conference on System Sciences, IEEE, Hawaii, 2012, pp. 3347-3356.

Taylor, F. W. *The principles of scientific management* 1st World Library, 1911.

Theodoridis, S., Koutroumbas, K., and Smith, R. "Pattern recognition.," Academic Press, 1999.

Thompson, J. D. *Organizations in action: Social science bases of administrative theory* Transaction Pub, 1969.

Tilson, D., Lyytinen, K., and Sørensen, C. "Digital Infrastructures: The Missing IS Research Agenda," *Information Systems Research* (21:5 - 20th Anniversary Special Issue of Emerging Challenges) 2010.

Tilson, D., Sørensen, C., and Lyytinen, K. "Change and Control Paradoxes in Mobile Infrastructure Innovation: The Android and iOS Mobile Operating Systems Cases," in: *45th Hawaii International Conference on System Science (HICSS 45)*, Maui, Hawaii, USA, 2012, p. 10.

Tiwana, A., and Keil, M. "Control in Internal and outsourced software projects," *Journal of Management Information Systems* (26:3) 2009, pp 9-44.

Tiwana, A., Konsynsky, B., and Bush, A. A. "Research Commentary - Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics," *Information Systems Research*:Articles in Advance - Published online ahead of print Novemeber 18, 2010) 2010, pp 1-13.

Tushman, M. L., and Anderson, P. "Technological discontinuties and organizational environments," *Administrative Science Quaraterly* (31) 1986, pp 439-465.

Ulrich, K. "The Role of Product Architecture in the Manufacturing Firm," *Research Policy* (24), Jan 1 1995, pp 419-440.

Vaast, E., and Davidson, E. "New Actors and New Media in Technology Discourse: An Investigation of Tech Blogging," *ICIS 2008 Proceedings*) 2008, p 162.

Van de Ven, A. H., Polley, D. E., Garud, R., and Venkatraman, S. (eds.) *The Innovation Journey*. Oxford University Press, New York, 1999.

Van de Ven, A. H., and Poole, M. S. "Methods for studying innovation development in the Minnesota Innovation Research Program," *Organization science* (1:3) 1990, pp 313-335.

Van de Ven, A. H., and Poole, M. S. "Explaining development and change in organizations," *The Academy of Management Review* (20:3) 1995, pp 510-540.

Van Dijk, T. A. *News analysis: Case studies of international and national news in the press* Lawrence Erlbaum, 1988.

von Hippel, E. *Democratizing Innovation* The MIT Press, Cambridge, Massachusetts, 2005.

Wagner, E., Galliers, R., and Scott, S. "Exposing best practices through narrative: The ERP example," in: *Information Systems Research,* B. Kaplan, D.P. Truex, D. Wastell, A.T. Wood-Harper and J.I. DeGross (eds.), Springer, 2004, pp. 433-451.

Wakeford, N., and Cohen, K. "Fieldnotes in public: Using blogs for research," in: *The Sage handbook of online research methods,* N.G. Fielding, R.M. Lee and G. Blank (eds.), Sage Publications Ltd, 2008, pp. 307-326.

Ward, M. H. "Thoughts on blogging as an ethnographic tool," 23rd annual ascilite conference, Citeseer, Sydney, Australia, 2006, pp. 3-6.

Weber, M., Henderson, A., and Parsons, T. *The theory of social and economic organization* New York: The Free Press, 1947.

West, J. "How Open is Open Enough? Melding Proprietary and Open Source Platform Strategies," *Research Policy* (32:7) 2003, pp 1259-1285.

White, H. "The value of narrativity in the representation of reality," *Critical Inquiry* (7:1) 1980, pp 5-27.

Woodard, C. J. "Architectural Control Points," Third International Conference on Design Science Research in Information Systems and Technology (DESRIST 2008), Atlanta, GA, 2008.

Yin, R. K. *Case Study Research: Design and Methods* Sage, Thousand Oaks, CA, 1994.

Yoo, Y. "Digital Materiality and the Emergence," in: *Materiality and Organizing: Social Interaction in a Technological World,* P.M. Leonardi, B.A. Nardi and J. Kallinikos (eds.), University of Michigan Press, Ann Arbor, MI, forthcoming.

Yoo, Y., Henfridsson, O., and Lyytinen, K. "The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research," *Information Systems Research* (21:5 - 20th Anniversary Special Issue of Emerging Challenges) 2010a.

Yoo, Y., Lyytinen, K., and Boland, R. "Distributed innovation in classes of networks," 41st Hawaii International Conference on System Sciences, IEEE, Hawaii, 2008.

Yoo, Y., Lyytinen, K., Boland, R., Berente, N., Gaskin, J., Schutz, D., and Srinivasan, N. "The Next Wave of Digital Innovation: Opportunities and Challenges: A Report of an NSF Research Workshop on "Digital Challenges in Innovation Research"."

Yoo, Y., Lyytinen, K., Thummadi, V., and Weiss, A. "Unbounded innovation with digitalization: A case of digital camera," Annual Meeting of the Academy of Management, 2010c.

Zittrain, J. "The Generative Internet," *Harvard Law Review* (119) 2006, pp 1974-2040.

Zittrain, J. *The Future of the Internet: And How to Stop It* Penguin, 2009, p. 352.

Zittrain, J. "Will the Web Break?," *Philosophical Transactions of the Royal Society,* 2012.