



# Technology diffusion: Shift happens – The case of iOS and Android handsets



Amitava Dutta<sup>a,\*</sup>, Abhinay Puvvala<sup>b</sup>, Rahul Roy<sup>b</sup>, Priya Seetharaman<sup>b</sup>

<sup>a</sup> George Mason University, 4400 University Drive, Fairfax, VA 22030, USA

<sup>b</sup> MIS Group, Indian Institute of Management Calcutta, Joka, Kolkata, West Bengal 700104, India

## ARTICLE INFO

### Article history:

Received 30 May 2016

Received in revised form 16 January 2017

Accepted 24 January 2017

Available online 15 February 2017

### Keywords:

Technology diffusion

Structural breaks

Mobile handsets

System dynamics

Causal models

## ABSTRACT

The diffusion of technology artifacts is often marked by abrupt events and incremental evolutionary moves, resulting in shifts in diffusion parameters as well as the underlying mechanics. In this paper, we model the diffusion of Android and iOS based handsets, where new models and operating system versions are released periodically. We relax a common assumption in IT diffusion studies, of holding diffusion parameters constant, and find that there are clear breaks in their values at specific points in time. Using the system dynamics methodology, we then develop and calibrate a causal model of the underlying mechanics. Significant events during evolution of the two platforms are matched temporally with the observed breaks, and the changing mechanics of diffusion across the breakpoints are identified using this causal structure. We find that iOS and Android handset diffusion patterns, although superficially similar, were driven by different mechanics. Our study contributes to the IT diffusion literature by (i) establishing the need to test for, and model, shifts in diffusion parameters over the horizon of interest (ii) offering a method to identify changes in diffusion mechanisms accompanying these shifts and (iii) demonstrating that similar temporal diffusion patterns need not imply similar underlying mechanics.

© 2017 Elsevier Inc. All rights reserved.

## 1. Introduction

The diffusion of technology artifacts has been studied in the literature for a long time and from a variety of perspectives. One major perspective models the temporal pattern of diffusion, with the aim of explaining the observed shape and/or forecasting how the pattern will evolve in the future. The Bass diffusion model (Bass, 1969) is perhaps the best known early work of this genre. Such models typically have one or more parameters which are calibrated using temporal data about the diffusion pattern. A common practice during calibration has been to assume that the model parameters remain unchanged over the time horizon of analysis (Gujarati, 2004). However, for IT artifacts specifically, diffusion is marked by incremental as well as significant abrupt events.

For instance, wireless routers have experienced incremental improvements in transmitted power and antenna design, as well as abrupt events such as introduction of the 802.11n standard. Flat panel displays have incrementally increased in size and have also experienced abrupt changes such as the introduction of liquid crystal technology over plasma. Similar mixes of incremental and abrupt changes have been witnessed in hard disk (magnetic to solid state) and optical drives (recording formats). We also see that, while these events frequently affect diffusion in a positive manner, it is not always the case. For example,

shortly after Apple introduced iPhone 6 with iOS 8, it introduced an update in the form of iOS 8.0.1. Many users complained about loss of network connectivity and malfunctioning of the Touch ID fingerprint sensor. This event did not help iOS 8 diffusion.

The specific IT diffusion context in our study is the diffusion of Android and iOS based mobile handsets. We briefly review relevant characteristics of their diffusion patterns which, together with the general observations above, motivate our research question that is presented immediately following this discussion. Mobile operating systems (mOS) offer a platform on which handsets can provide rich functionality to end users, beyond telephony, through a variety of applications (Apps). Worldwide, the installed base of smartphone handsets grew from 237 million in 2008 to 2562 million in 2016 (Statista, 2016). Table 1 shows the share of mOS as a percentage of handheld units shipped<sup>1</sup>. Android and iOS account for a lion's share of this market. Clearly, the growth of these two platforms represents a major IT diffusion phenomenon.

Figs. 1 and 2 show annual sales and change in sales of Android and iOS smartphones from inception until 2014 (Gartner, 2017). Note that the sales curves for both platforms exhibit changing patterns like decline, stagnation and growth. Moreover, during the timespan covered by Figs. 1 and 2, both platforms experienced incremental and abrupt events. Examples of the former include improvements in battery life and ergonomics. The latter include new handset model and mOS version introduction and developer policy changes instituted by Apple.

\* Corresponding author.

E-mail addresses: adutta@gmu.edu (A. Dutta), p.abhinay@gmail.com (A. Puvvala), rahul@iimcal.ac.in (R. Roy), priyas@iimcal.ac.in (P. Seetharaman).

<sup>1</sup> Data available from International Data Corporation at <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>, accessed 18th January, 2016.

**Table 1**  
Market share of mOS worldwide.

Period	Android	iOS	Windows Phone	BlackBerry	Others
Q1 2016	84.1%	14.8%	0.7%	0.2%	0.2%
Q1 2015	78.0%	17.5%	2.5%	0.4%	0.5%
Q1 2014	81.2%	15.2%	2.5%	0.5%	0.7%
Q1 2013	75.5%	16.9%	3.2%	2.9%	1.5%
Q1 2012	59.2%	22.9%	2.0%	6.3%	9.5%

The presence of multiple archetypes – decline, stagnation, growth – in the diffusion pattern of IT artifacts as seen above, and the accompanying incremental and abrupt events, lead us to surmise that the underlying mechanisms and parameters driving the diffusion change over time. This leads to the following research question: how can we model IT diffusion patterns in a way that reveals changes in underlying causal mechanisms and which accommodates changes in model parameters over time caused by incremental and abrupt events that frequently accompany these phenomena. As we will find in the literature review, the common practice in modeling technology diffusion is to assume that the driving mechanics and process parameters remain constant. This ignores the reality that IT artifacts undergo both gradual and abrupt changes during their diffusion. Hence our research question frames the IT diffusion phenomenon in a more realistic manner and our findings should therefore be of theoretical interest. Moreover, our interest in revealing the changing mechanisms driving the phenomenon is of practical interest in managing the diffusion process because we get a better understanding of the relationships between specific temporal events and their effects on the diffusion patterns, and helps produce actionable information for handset manufacturers and mOS firms.

The paper is organized as follows. In the next section we survey relevant literature on IT diffusion models focusing on modeling techniques and assumptions. A summary of the evolution of iOS and Android follows, highlighting several abrupt events which illustrate the contextual characteristics that motivated this modelling effort. Then, using time series data, we check for the presence of ‘breaks’ in the diffusion pattern. The existence of breaks suggests changes in the underlying diffusion mechanism. In the subsequent section, a causal model of iOS and Android growth is developed using the system dynamics methodology, and then calibrated to accommodate the pattern shifts identified earlier. This model is then analysed to identify the dominant mechanisms that drive the diffusion patterns and how they change over time as the patterns shift. We conclude with a discussion of the contributions of this

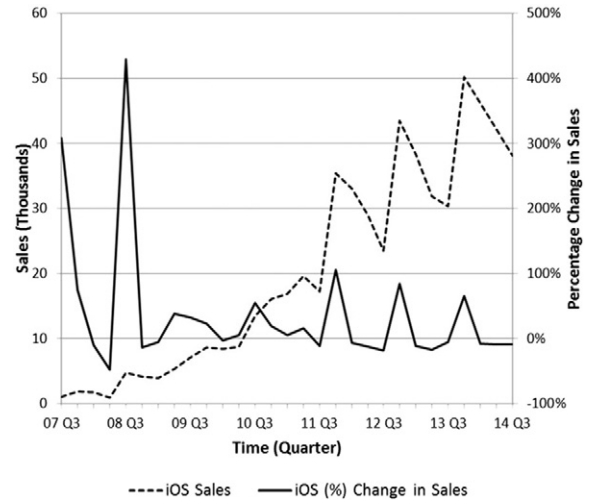


Fig. 2. Sales of iOS handsets.

study to the IT diffusion literature, applications to other technology diffusion settings, and limitations.

**2. Literature review**

Given the broad literature surrounding technology diffusion, it is necessary to focus on the segment that is directly relevant to our investigation. As noted earlier, our specific interest is in modelling shifts in the temporal pattern of diffusion and uncovering the changing mechanisms that result in the observed pattern shifts. Thus we will review the diffusion literature through this filter and exclude other established themes, such as organizational and individual enablers of and barriers to diffusion (Gupta and Jain, 2014) and impact of public policy and regulation (Cho and Choi, 2015).

One pervasive presence in diffusion modeling has been the Rogers diffusion of innovations model (Rogers, 1962) whose underlying mechanism is that of contagion, where actual adopters influence potential adopters. Different parameters, such as propensity to innovate or imitate, modulate the diffusion. Numerous studies have used this classical model to understand large scale technology diffusion (Baskerville and Pries-Heje, 1998; Fichman, 1992, 2004), although a weakness in capturing shifts in diffusion patterns has also been noted (Kauffman and Techatassanasoontorn, 2009). Information technologies which have been examined using this model include multimedia message service (Hsu et al., 2007); mobile phone adoption (Kauffman and Techatassanasoontorn, 2012; Watanabe et al., 2009); BITNET adoption in academe (Levin et al., 2012); open source software such as Apache web servers (Lakka and Michalakelis, 2012); and mobile social networking (Scaglione et al., 2015). Variants of the contagion mechanism have also appeared, such as punctuated equilibrium (Loch and Huberman, 1999), social networks (Susarla et al., 2012) and proportional hazards (Greenan, 2015).

Bitnet growth was modeled in Gurbaxani (1990) using the Logistic and Gompertz functions, both of which are consistent with a contagion mechanism. The Gompertz model was found to be a good fit for diffusion of mobile telephony in a developing country context (Gupta and Jain, 2014). Internet growth was earlier modeled in Rai et al. (1998) using the same two functions and the exponential function. Interestingly, although the exponential function is not based on a contagion mechanism, it produced the best fit among the three models. One reason offered by the authors for the poorer fit of the Logistic and Gompertz models was the assumption that contagion parameters remain unchanged over the period of analysis. Some recent studies have examined this scenario of changing diffusion parameters. Meade and Islam (2006) review diffusion studies which attempt to capture the time varying

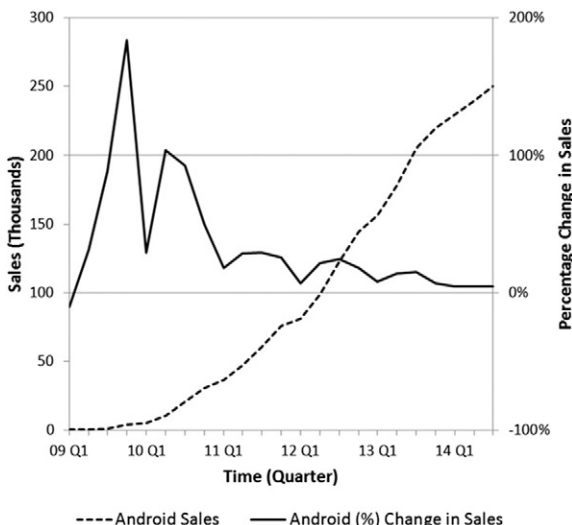


Fig. 1. Sales of Android handsets.

nature of the model parameters, also termed adaptive estimation. Change in marketing mix variables, introduction of successive generations of a technology and diffusion of the same technology in multiple countries were some of the contexts that were examined (Lakka et al., 2013). For example, Watanabe et al. (2009) modelled growth of mobile phones in Japan as waves of diffusion using a Bi-Logistic model where model parameters change from one wave to the next.

The preceding review confirms the extensive attention devoted to IT diffusion, and also reveals some of the gaps in the literature that we aim to address in our work. The first shortcoming evident in the literature is the widespread assumption of constant model parameters. This is suspect particularly in technology diffusion scenarios for a variety of reasons. Technology capabilities change over time, sometimes rapidly. The switch from analog to digital signal transmission, the introduction of CDMA or GSM standards for mobile telephones, are examples of such abrupt technology advances that affected diffusion. Second, Consumer or user perceptions of the value proposition evolve over time, especially when additional offerings become available through the technology (Ceccagnoli et al., 2012), such as in the case of e-commerce on the Internet or mobile value added services in the case of smartphones (Koch and Kerschbaum, 2014). Third, regulations evolve and affect the interaction between technology and individuals, such as limits on the number of competing mobile carriers (Casey and Töyli, 2012; Kauffman and Techatassanasoontorn, 2009). In short, the drivers of the contagion process in technology settings may cause diffusion parameters to change over even relatively short periods of time. So the first contribution by our study to the literature is that it quantitatively identifies breakpoints in the diffusion pattern and allows model parameters to change across them.

The second shortcoming that emerges from the preceding review is that these models strive to fit the observed diffusion pattern well, but essentially view the diffusion process itself as a black box. There is minimal focus on the mechanics of diffusion. While this may meet the needs of some applications such as forecasting, it is inadequate for other applications where one is seeking to deliberately intervene and manage the diffusion process by design. For example, policy makers may be interested in understanding the potential impact of changes in password policies on diffusion behavior (Baskerville and Pries-Heje, 1998). Unless one understands the cause effect relationships between drivers and the observed behavior, meaningful intervention in the diffusion process is not possible. This, our second contribution to the literature, is by developing a causal model of the diffusion pattern, using the system dynamics methodology, which opens the black box and reveals underlying mechanisms and how they change over time. This allows our model to address another shortcoming in the current models which is their inability to represent periods of slowdown, or outright decrease, in the diffusion pattern (Gary et al., 2008; Paich and Sterman, 1993).

System dynamics (SD) (Richardson, 1996) offers an alternate approach to model technology diffusion and has complementary strengths to models discussed above (Casey and Töyli, 2012; Choi et al., 2010; Daim et al., 2006; Dutta and Roy, 2005). By representing cause-effect relationships among the underlying variables in the phenomenon, one can offer a causal explanation for the observed growth pattern using the reinforcing and balancing feedback loops present in the structure, understand the impact of different policy scenarios and identify handles which can be used to alter the course of the phenomenon. In summary, our study addresses two gaps evident from the preceding review. One is the need to recognize that incremental and abrupt events accompany diffusion of IT artifacts and the diffusion parameters can shift as a result. Second, while it may be acceptable to view the diffusion process as a black box for some applications, there are interventional settings in which it would be useful to know the changing mechanics of diffusion in order to assess the potential consequences of interventions. In the remainder of the paper, we develop a diffusion model to address these shortcomings using the specific context of iOS and Android based handsets.

### 3. Evolution of Android and iOS

As noted earlier, mobile operating systems (mOS) provide a platform for handsets to offer enhanced functionality through various Apps developed by third parties. Together, Android and iOS account for a lion's share of this market. We briefly discuss the evolution of Android and iOS to paint a picture of the context within which their diffusion has occurred.

#### 3.1. The Android story

Android is a Linux-based operating system for mobile telephones and tablets, that has been developed by the Open Handset Alliance in partnership with Google and other companies (Burgelman et al., 2009). The source code is available under free and open source licenses. To use the Android trademark, device manufacturers must ensure that the device complies with the Compatibility Definition Document (CDD) and get permission from Google. They then have complete freedom to use and customize without having to pay a royalty to Google.

Several versions of Android have been released since the original Alpha - Beta (1.1), Cupcake (1.5), Donut (1.6), Eclair (2.0–2.1), Froyo (2.2–2.2.3), Gingerbread (2.3–2.3.7), Honeycomb (3.0–3.2.6), Ice Cream Sandwich (4.0–4.0.4), Jelly Bean (4.1–4.3.1), KitKat (4.4–4.4.4, 4.4W–4.4W.2) and Lollipop (5.0–5.1). The inter-version release time has been approximately 147 days. Some of the important events are shown in Fig. 3.

While Android diffusion has been rapid, user experience has not been uniformly good. Fig. 4 shows a plot of normalized count of searches for the key words 'Android Issues' and 'Android Problems' alongside introductions of different versions of Android. The pattern points to growing usability issues until about December 2011.

The price/performance profile of Android handsets has improved over time. The first device, launched on 22nd October 2008, ran on a 528 MHz processor, offered 300 min of talk time, weighed 5.6 oz., had a 3.2 in. screen, and a 3.2 megapixel camera. The price was \$399. During 2008 to 2011 the number of manufacturers increased from one to eleven (German, 2011), ninety-five models were launched, the minimum price dropped from \$330 to about \$30. However, the user satisfaction level fell from 3.50 (on a scale of 1–5 where 5 indicates highest satisfaction level) to 2. Table 2 summarizes these changes.

Google Playstore was built for distributing third party Apps for Android. For App developers, the Android market offers several attractive features. The source code was open source, the Software Development Kit (SDK) needed to connect their Apps with Android was distributed free, and Apps did not have to undergo a review and acceptance process before being hosted on Playstore. Google exercised little control over the Apps developed and hosted on the Android Market. However, the experience of Android App developers has not been uniformly positive. Given the large number of manufacturers and phone models having different input mechanisms, processor types and screen sizes, developers had to test their software for this multitude of hardware configurations. Coexistence of different versions of the Android operating system complicated the situation further.

#### 3.2. The iOS story

Apple's iOS, the operating system for Apple smartphones and tablets was unveiled on January 9th 2007 as the iPhone operating system and was not referred to as iOS until its fourth major release in April 2010. Like Android, iOS also runs on a Unix core. Its installed base is much smaller than that of Android. The iOS platform, like the OSX operating system of Apple, is very vertically integrated and has been limited to devices made by Apple. When iOS was introduced, it was well behind Windows Mobile, Palm OS, Symbian, and even BlackBerry in a strict feature-to-feature comparison. Yet the first device using iOS, Apple's iPhone, experienced strong sales.

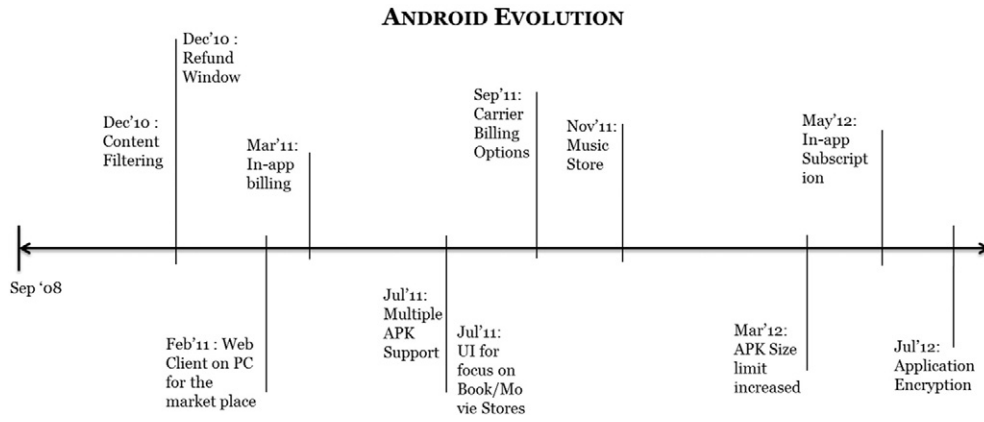


Fig. 3. Major events during evolution of Android.

While the rest of the mobile handset market competed on technical specifications and features, iOS focused on the core user experience, on speed, consistency between apps, and on making a few features (such as browser, maps, itunes ecosystem, keyboard, touch interface) significantly better than anything available at that time (Bohn et al., 2013). Some of the major events accompanying iOS evolution are presented in Fig. 5.

Table 3 shows changes in characteristics of iOS handsets. Note that the user rating does not fall below 3, indicating high user satisfaction (JDPower, 2013). Fig. 6 shows the pattern of Web searches based on the keywords ‘iOS issues’ and ‘iOS problems’. The frequency of inquiries about iOS spiked just before the release of new versions, but also drops sharply in a month or two thereafter. This is in contrast to Fig. 4 where one also sees spikes in web searches around the time of the release of

new Android versions, but the search level does not die down, indicating continuing issues and problems. One milestone in iOS’ evolution was the introduction of the App-store for iOS developers. It opened in July 2008 with 552 Apps and, as of September 9, 2014, it hosted 1,300,000 Apps which together experienced 75,000,000 downloads and average 62,500 downloads per app. Apple earns 30% of revenue generated from app sales.

In summary, we can see that the growth of Android and iOS handsets has occurred through complex interaction among consumers, mOS builders (Apple and Google), handset manufacturers, and app developers. Understanding mOS based handset diffusion therefore requires models that take this interaction into account and accommodates the impact of abrupt events that accompany their evolution.

#### 4. Modelling mOS handset diffusion

We use a two-phase approach, where the first phase consists of identifying the presence and timing of structural breaks in the diffusion pattern. The second phase consists of developing a causal model that represents the mechanics of diffusion and uses the structural break information from phase one for estimating parameters and identifying changes in causal mechanisms.

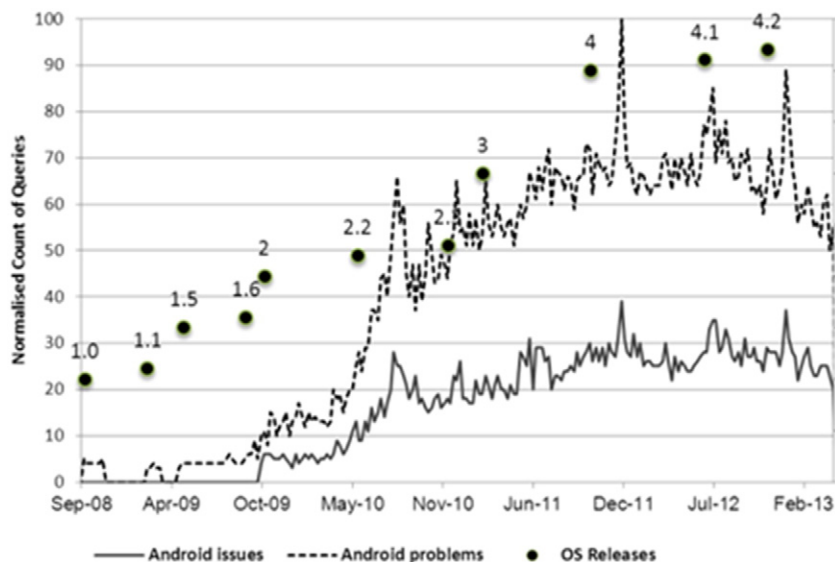
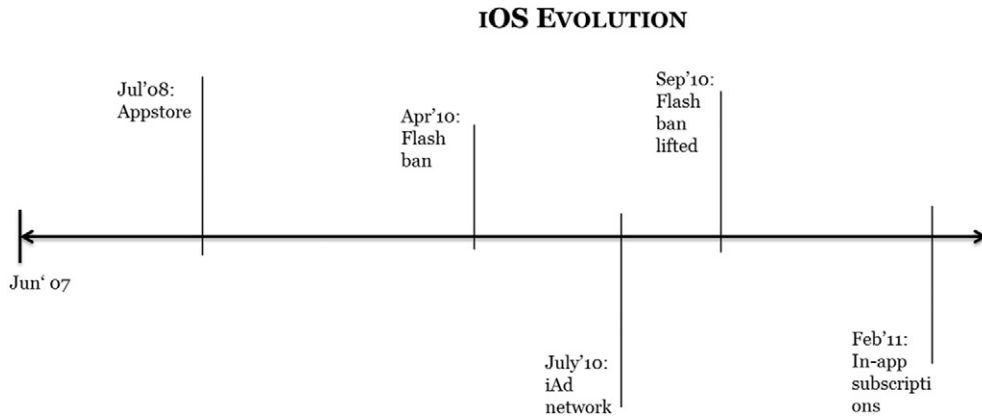


Fig. 4. Web Search interest in Android Issues and Problems (Data from Google Trends).

**Table 2**  
Changes in Android handset model characteristics.

Month-year	Number of models	Min user rating	Min MSRP	Max talk time minute	Min weight	Max screen size (inch)
Oct-08	1	3.50	330	300.00	5.60	3.2
Oct-09	6	2.50	179	385.00	5.70	3.7
Mar-10	12	2.00	100	350.00	4.70	3.1
Nov-10	54	1.00	30	540.00	3.60	3.8
Jul-11	95	2.00	129	624.00	3.88	4.3



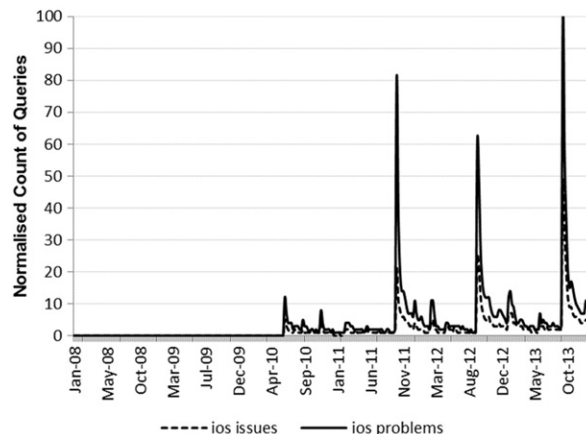
**Fig. 5.** Major policy changes during evolution of iOS.

**Table 3**  
Changes in iOS handset model characteristics.

Month-year	Handset model	User rating	Talk time (hours)	Weight (grams)	Diagonal screen size (inch)
Jun-07	iPhone	3.50	8	135	3.5
Jul-08	iPhone 3G	3.00	5	133	3.5
Jun-09	iPhone 3GS	3.50	5	135	3.5
Jun-10	iPhone 4	3.50	7	137	3.5
Oct-11	iPhone 4S	4.00	8	140	3.5
Sep-12	iPhone 5	4.00	8	112	4.0
Sep-13	iPhone 5C	2.50	10	132	4.0
Sep-13	iPhone 5S	3.50	10	112	4.0

4.1. Testing for structural breaks

If the parameters of a time series model are not constant over time, structural changes are said to be present (Gujarati, 2004), and there are several methods for detection of them (Andrews, 2003; Bai and Perron, 2003; Chow, 1960). We choose to use the Bai-Perron test as it can find the optimal number of break points based on rigorous criteria. The test uses either Akaike's Information Criterion (AIC) or the Bayesian Information Criterion (BIC) scheme (Burnham and Anderson, 2004) to appropriately address the trade-off between complexity and goodness of fit. The model with the



**Fig. 6.** Web search interest in iOS issues and problems.

**Table 4**  
BIC values for different number of breaks.

No. of breaks	BIC values (Android sales)	BIC values (iOS sales)
0	29.04	29.85
1	28.61	28.97
2	28.46	28.68
3	28.08	28.84
4	28.02	29.00
5	28.26	29.89

lowest AIC or BIC is chosen. Researchers often prefer BIC over AIC (Burnham and Anderson, 2004) on the grounds that AIC is liberal in its criteria and frequently prefers a more complex, wrong model over a simpler, true model. So we used the Bai-Perron test with BIC to identify structural breaks. Data was obtained from a variety of sources and included handset sales, user ratings surveys, as well as narratives on various events associated with the handset diffusion. These sources are summarized in the section on model calibration. For the time series analysis that follows, we used monthly sales data (in number of units) for iOS and Android spanning January '09–April '12.

Nonlinear time series regression equations were fitted using handset sales at time  $t$  as the dependent variable and powers of time ( $t, t^2, t^3$  etc.) as the independent variables. A quadratic equation was found to have the best fit, yielding  $Y_t = \beta_1 t + \beta_2 t^2 + \varepsilon$  where  $Y_t$  is the handset sales at time  $t$ . Table 4 gives the BIC values for different number of breaks and the lowest BIC for Android is obtained with 4 breaks. For iOS, the lowest BIC occurs with 2 breaks. The timing of those breaks is at 14, 20, 26 and 34 months for Android and at 16 and 32 months for iOS handset sales.

Confirmation of structural breaks suggests that there are events that are likely reasons behind these changes. In trying to identify such events, we choose as a first approximation, those that most closely precede the break dates. We examined App store–developer license agreements, company press releases and searched other publicly available information. Articles on technology websites like Wired, Endgadget, Mashable, and CNET were particularly informative. The significant Android-related events that temporally precede its breakpoints are launch of OS version 2.1 (12/01/2010), 2.3.3 (09/02/2011) and 4.0 (19/10/2011), SDK upgrades released on 20/05/2010, and introduction of in-app billing system in March 2011. One would expect a break date close to an OS version release as it impacts both handset sales and complementary innovation by developers. Similarly, SDK improvements significantly impact the developer community by reducing the effort to build Apps. In-app billing systems have significantly impacted the magazine/news genre of Apps. Instead of a one-time charge for purchasing an app, users can now install for free and subscribe to the features they need.

For iOS, the break points coincide with two major events. In April 2010 Apple banned use of Flash on the iPhone compiler as a result of the ongoing battle with Adobe over Flash on iOS. There was substantial resentment from the developer community along with demands to reverse the ban. Over the next few months there was a significant drop in the rate of additions to the App-store, and Apple retracted the restriction on usage of any private APIs. We surmise the first break in month 16 points to effects of this policy change. The second break is in month 32 (Sep 2011) and coincides with the release of iOS 5 in June 2011. Fig. 6 shows a steep increase in iOS related queries around that time.

The temporal association between structural breaks in the diffusion patterns of iOS and Android handsets and events related to Android and iOS evolution discussed in the preceding paragraphs suggests an underlying web of cause–effect relationships. In the next section, we develop a simple causal model of the underlying mechanics that captures these interactions.

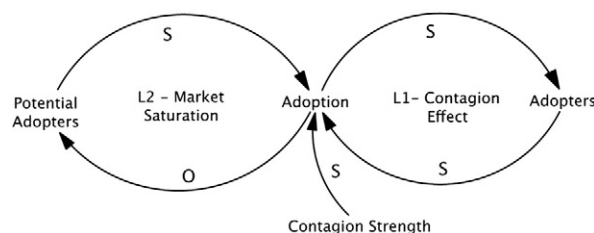
#### 4.2. A causal model of Android and iOS growth

To develop a causal model of the handset diffusion process, we used the system dynamics methodology (SD) (Forrester, 1961). It has been successfully applied to study the temporal behavior of systems in a wide variety of application domains (Forrester, 1994) including supply chain and inventory management (Morecroft, 1983), environmental policy making, managing research and development strategies (Garcia et al., 2003), project management, collusion in auctions (Padhi et al., 2016) etc. For brevity, the conceptual building blocks of SD are presented in parallel with the development of the models themselves. Additional technical details can be found in (Sternan, 2000). We present the model in a top-down fashion beginning with a 'sectoral' model which shows the major interacting causal components. That is followed by a detailed stock-flow model that was used for parameter estimation and evaluation. Such top-down presentation of system dynamics models is not uncommon in the literature (Soydan and Oner, 2012).

Fig. 7 shows a Causal Loop Diagram (CLD) of the cause–effect relationships in the basic contagion mechanism that drives the diffusion process. This structure will form the foundation for the sectoral model and hence deserves some elaboration.

A CLD consists of variables connected by cause–effect relationships, the latter being represented by a signed arrow. Positive links (marked as 'S') imply that the cause and effect move in the same direction (more Adopters causes more Adoption through contagion) while negative links (marked as 'O') indicate that cause and effect move in opposite directions (more Adoption reduces Potential Adopters and the market saturates).

A closed sequence of links yields a feedback loop. Positive feedback loops have an even number of negative links, while negative loops have an odd number of negative links. Negative (balancing) loops are goal seeking in that they try to move the behavior back to a steady state if any deviation from



**Fig. 7.** CLD for basic contagion mechanism.

that state is observed. Positive (reinforcing) loops are unstable in that once initiated, they move the system further and further away from the initial state. The structure in Fig. 7 shows two feedback loops:

1. Reinforcing loop L1 on the right (Contagion Effect):  $Adoption \rightarrow + Adopters \rightarrow + Adoption$
2. Balancing loop on the left L2 (market saturation):  $Adoption \rightarrow - Potential Adopters \rightarrow + Adoption$ .

The parameter *Contagion Strength* modulates the adoption process. L1 reflects the reinforcing Contagion Effect stemming from word of mouth. L2 represents the throttling mechanism of Market Saturation. The exogenous representation of *Contagion Strength* reflects an assumption that the parameter is unaffected by the diffusion itself. Of course, we know from the preceding review of the evolution of iOS and Android, that the diffusion of both these platforms exhibit changes in patterns at different points in time, indicating that the Contagion Strength parameter is itself affected by the diffusion process – i.e. it is endogenous. The sectoral model, shown in Fig. 8, recognizes this reality and thus modifies and builds on the basic structure of Fig. 7 in two ways. First, it shows the causation structure for the individual sectors and second, it shows the interconnections across the sectors.

There is one sector for each player – iOS and Android – and the two players draw from the same pool of *Potential Adopters* which constitutes the third sector which is the full market for mOS based handsets. Since the sector structure for each player is the same, we will explain the two sectors in parallel. The two sectors in Fig. 8, for Android and iOS respectively, build on the basic contagion and market saturation feedback loops mentioned earlier, using additional loops that capture influences present in the context of mOS handset diffusion. These additional loops capture the feedback effects of the diffusion process itself, on the contagion strength, resulting in this parameter becoming endogenous as suggested above.

**L11** (*Android Adoption, Android Adopters*), **L21** (*iOS Adoption, iOS Adopters*): These two loops model the basic contagion mechanism shown earlier in Fig. 7, but for Android and iOS separately.

**L13** (*Android Adopters, Positive Effects of Android Adopters on Attractiveness, Android Contagion Strength, Android Adoption*), **L23** (*iOS Adopters, Positive Effects of iOS Adopters on Attractiveness, iOS Contagion Strength, iOS Adoption*): The diffusion of Android or iOS, which are two-sided platforms, depends on network effects (Evans and Schmalensee, 2010) among end-users and application developers. The direct network effect occurs as new end-users join the platform, increasing its value for existing users as well (Economides, 1996). The increasing adopter pool also invites application developers who derive value by selling goods that enhance value of the mOS (Park, 2004; Werden, 2001). The positive contribution of the network effects is built into the variable *Positive Effects of Android Adopters on Attractiveness* and *Positive Effects of iOS Adopters on Attractiveness*. These two

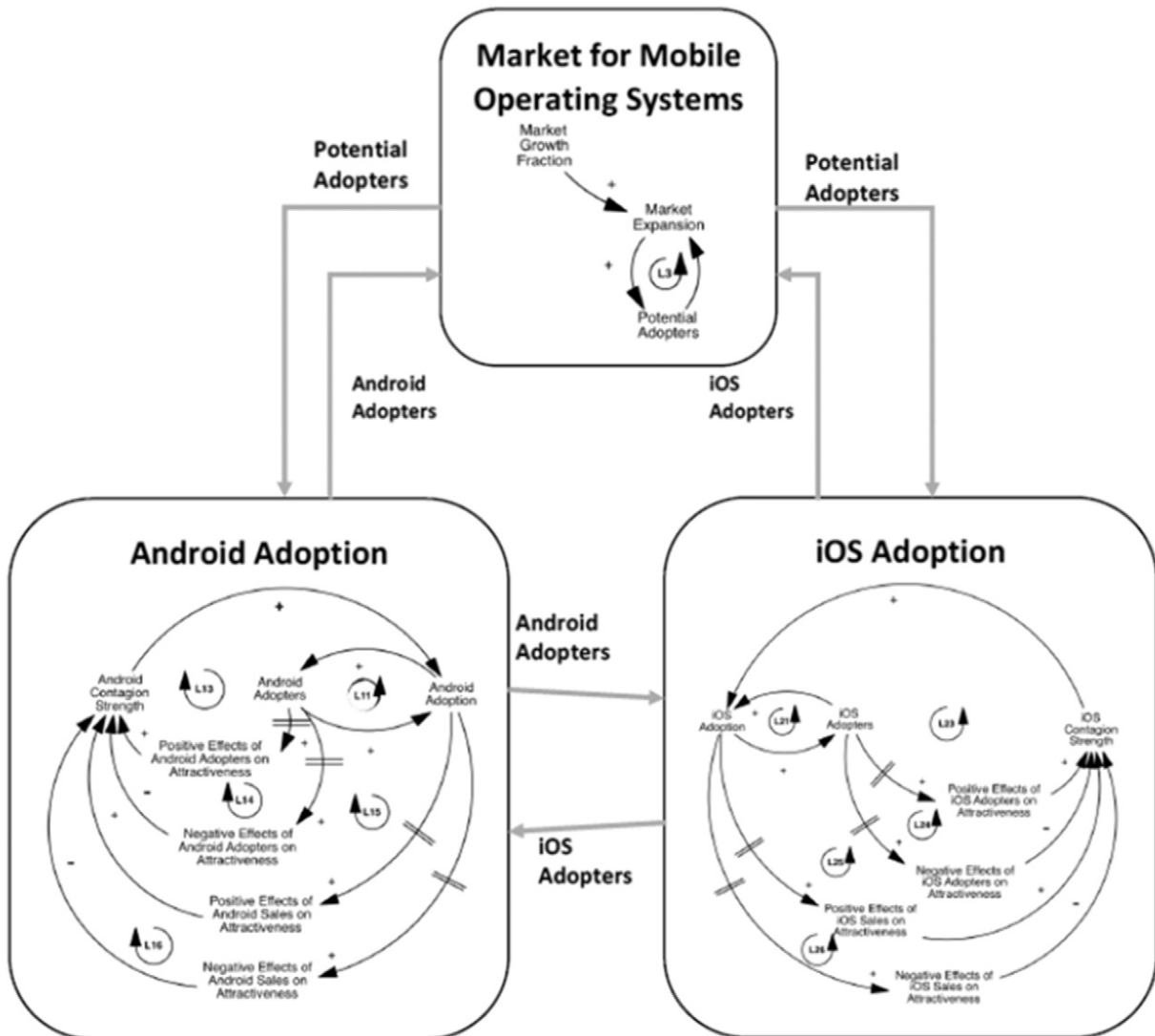


Fig. 8. Sectoral model of Android/iOS handset diffusion.

loops model how growth in handset adopters endogenously strengthens their respective contagion parameters instead of being held constant exogenously.

**L14** (*Android Adopters, Negative Effects of Android Adopters on Attractiveness, Android Contagion Strength, Android Adoption*), **L24** (*iOS Adopters, Negative Effects of iOS Adopters on Attractiveness, iOS Contagion Strength, iOS Adoption*): As noted in the section on iOS and Android evolution, adoption experience is not always positive. For instance, as the adopter pool grows large, maintaining service levels becomes more difficult, causing deterioration in user experience and weakening the contagion strength. The indirect network effect could also experience unanticipated side effects similar to the Atari experience (Coughlan, 2004). Initially, there was an influx of high quality games driven by the increasing user base. However, in later periods, lower quality games, and even some games deemed objectionable by segments of the user population, appeared, leading to negative network effects. Halaburda et al. (2010) also offer evidence that the increasing variety of Android handset models has made App development difficult and may have weakened the indirect network effect through dissatisfaction among developers. L14 and L24 model the negative outcomes that erode attractiveness and weaken the contagion process.

**L15** (*Android Adoption, Positive Effects of Android Sales on Attractiveness, Android Contagion Strength*), **L25** (*iOS Adoption, Positive Effects of iOS Sales on Attractiveness, iOS Contagion Strength*): Attractiveness improves with growing sales, which increases product awareness; falling handset price from economies of scale and learning effects; improvement in handset features and quality (Tables 1 and 2). L15 and L25 model these positive effects.

**L16** (*Android Adoption, Negative Effects of Android Sales on Attractiveness, Android Contagion Strength*), **L26** (*iOS Adoption, Negative Effects of iOS Sales on Attractiveness, iOS Contagion Strength*): However, Tables 1 and 2 shown earlier indicate that growing sales have not been without negative impacts on user satisfaction, albeit to different degrees for Android and iOS. With these two feedback loops we have modeled the negative effects of growing sales that can weaken the contagion process.

**L3** (*Potential Adopters, Market Expansion*): The market for smart phones expands due to general economic and demographic factors. This loop models that effect and constitutes the third sector named Market for Mobile Operating Systems.

Intersector linkages: The three sectors are linked among themselves as follows. The pair of links between Android Adoption and Market for Mobile Operating Systems represents the linkage between the variables Android Adoption and Potential Adopters. This loop models how the *Potential Adopters* pool is depleted as people buy Android handsets and join the pool of adopters. Similarly, a pair of links connects the sector iOS Adoption to Market for Mobile Operating Systems. Both players draw from the same pool of *Potential Adopters*. There is also a pair of links between the iOS and Android adoption sectors. These two linkages reflect the fact that the likelihood of an Android adopter influencing a potential adopter to buy an Android handset depends on the number of iOS adopters as well, and vice versa because both platforms are drawing on the same pool of potential adopters.

In summary, the sectoral model shows that Android and iOS diffusion is driven endogenously by both positive and negative influences, and they affect each other's growth by drawing on a common pool of potential customers. We proceed now to present a detailed stock-flow model corresponding the sectoral model shown above.

### 4.3. Simulation model construction

The sectoral model of Fig. 8 was converted into a stock-flow diagram that appears in Fig. 9. A simulation model was then generated by developing appropriate functional expressions for the different cause-effect relationships. Following the discussion in the sectoral model, the two stocks variables *iOS Adopters* and *Android Adopters* draw from a common stock variable called *Potential Adopters*, through two flow variables *iOS Adoption*

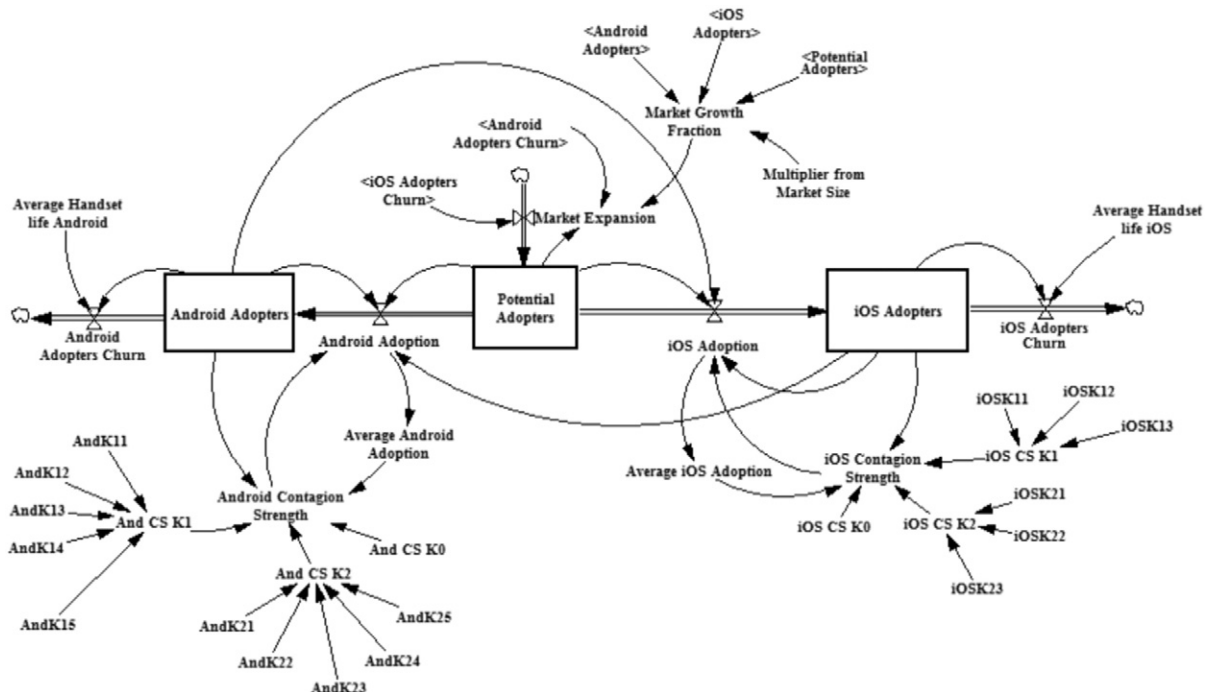


Fig. 9. Stock flow model of Android and iOS diffusion.



and *Android Adoption*, respectively. This structure can be represented by the following equations:

$$\frac{d(\text{Android Adopters})}{dt} = \text{Android Adoption} - \text{Android Adopters Churn} \quad (1)$$

$$\frac{d(\text{Potential Adopters})}{dt} = -\text{Android Adoption} - \text{iOS Adoption} + \text{Market Expansion} \quad (2)$$

An equation similar to Eq. (1) is associated with iOS adoption, and the variable *Market Expansion* in Eq. (2) captures the expansion referred to earlier due to demographic factors. The flow variable *Android Adoption* is the fraction of *Potential Adopters* converting to *Android Adopters* through the contagion process. This fraction will depend on the likelihood that a *Potential Adopter* meets an *Android Adopter* and the likelihood that this encounter will lead to *Adoption*. Eq. (3) below captures that mechanism. Market Expansion, as has been written in Eq. (4), aggregates adopters rejecting their respective platforms and the expansion of the market happening over time. Eq. (5) represents a simple exponential decay process in which a fraction of Android adopters reject the Android platform.

$$\text{Android Adoption} = \text{Android Contagion Strength} * \text{Potential Adopters} * \frac{\text{Android Adopters}}{\text{Potential Adopters} + \text{Android Adopters} + \text{iOS Adopters}} \quad (3)$$

$$\text{Market Expansion} = \text{Potential Adopters} * \text{Market Growth Fraction} + \text{iOS Adopters Churn} + \text{Android Adopters Churn} \quad (4)$$

$$\text{Android Adopters Churn} = \frac{\text{Android Adopters}}{\text{Average Handset life Android}} \quad (5)$$

The sectoral model in Fig. 8 was simplified by merging four pairs of loops (L13, L14), (L15, L16), (L23, L24) and (L25, L26) into four loops by writing each pair as a single equation. Each pair represents the same relationship although the direction of causation of the first (e.g. L13) was positive while that of the second (e.g. L14) was negative. Since we are interested in the ‘net effect’ of the positive and negative externalities on attractiveness we allowed the polarity to be determined empirically as shown in Fig. 9.

Thus, to capture both negative and positive externalities, *Android Contagion Strength* was first rewritten as:

$$\text{Android Contagion Strength} = K_{\text{android},0} * e^{(K_{\text{android},1} * \text{Android Adopters} + K_{\text{android},2} * \text{Android Adoption})} \quad (6)$$

$K_{\text{android},0}$  is a constant. Initial model testing showed that both iOS Contagion Strength and Android Contagion Strength reached a steady state value asymptotically. We therefore chose an exponential form with polarity to be determined from the data. Furthermore,  $K_{\text{android},1}$ ,  $K_{\text{android},2}$  were expressed in a form that incorporated knowledge about structural breaks identified earlier. Pulse (start-time, duration) functions were used to inject disruptions at break points, as in Eqs. (7) and (8).

$$K_{\text{android},1} = K_{\text{android},11} * \text{Pulse}(0, 14) + K_{\text{android},12} * \text{Pulse}(14, 6) + K_{\text{android},13} * \text{Pulse}(20, 6) + K_{\text{android},14} * \text{Pulse}(26, 8) + K_{\text{android},15} * \text{Pulse}(34, 6) \quad (7)$$

$$K_{\text{android},2} = K_{\text{android},21} * \text{Pulse}(0, 14) + K_{\text{android},22} * \text{Pulse}(14, 6) + K_{\text{android},23} * \text{Pulse}(20, 6) + K_{\text{android},24} * \text{Pulse}(26, 8) + K_{\text{android},25} * \text{Pulse}(34, 6) \quad (8)$$

Use of Pulse functions enabled  $K_{\text{android},1}$ ,  $K_{\text{android},2}$  to take different values across break points. Thus  $K_{\text{android},1}$  took the value of  $K_{\text{android},11}$  between  $t = 0$  and  $t = 14$ ,  $K_{\text{android},2}$  took the value of  $K_{\text{android},25}$  between  $t = 34$  and  $t = 39$  and so on. This method of writing  $K_{\text{android},1}$ ,  $K_{\text{android},2}$  also enables the causal structure to generate exponential growth (positive exponent) or logarithmic growth (negative exponent) in different time segments, depending on the relative magnitude of the two terms  $K_{\text{android},1} * \text{Android Adopters}$  and  $K_{\text{android},2} * \text{Android Adoption}$ . Equations for iOS Contagion Strength were written similarly to incorporate its breakpoints.

$$\text{iOS Contagion Strength} = K_{\text{ios},0} * e^{(K_{\text{ios},1} * \text{iOS Adopters} + K_{\text{ios},2} * \text{iOS Adoption})} \quad (9)$$

$$K_{\text{ios},1} = K_{\text{ios},11} * \text{Pulse}(0, 16) + K_{\text{ios},12} * \text{Pulse}(16, 16) + K_{\text{ios},13} * \text{Pulse}(32, 8) \quad (10)$$

$$K_{\text{ios},2} = K_{\text{ios},21} * \text{Pulse}(0, 16) + K_{\text{ios},22} * \text{Pulse}(16, 16) + K_{\text{ios},23} * \text{Pulse}(32, 8) \quad (11)$$

Data for estimating model parameters came from secondary sources shown in Table 5. A large segment of the data was purchased from a German mobile analytics firm that maintains platform-wise databases. Policy changes were identified through extensive Internet based search on official

**Table 5**  
Sources of data for model calibration.

Description	Periodicity	Period	Source
Android sales	Monthly	Jan'09–Jul'12	Research2Guidance, Mobile Analytics Firm – Paid Source
Apps -Google Play (Android MARKET)	Monthly	Jan'09–Jul'12	
iOS sales	Monthly	Apr'07–Jul'12	
Appstore Apps	Monthly	Jul'08–Jul'12	
Apps (category wise) (Android/iOS)	Monthly	Dec'08–Jun'12	
Downloads (platform wise)	Daily	Dec'08–Jun'12	
Policy changes (Android and iOS)	NA	Till Jun'12	Android official developer blog, iOS developer license agreements, CNET, Mashable, Wired, Techcrunch
Handset specifications	NA	Till Jun'12	GSMARena.com
OS version ratings	NA	Till Jun'12	CNET
String search data	Daily	Till Jun'12	Google Analytics

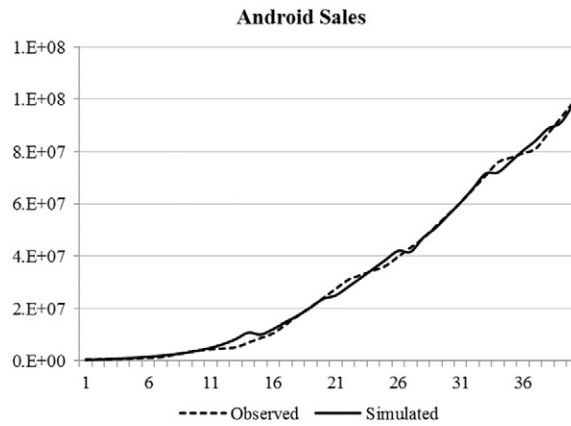


Fig. 10. Observed and simulated values of Android adoption.

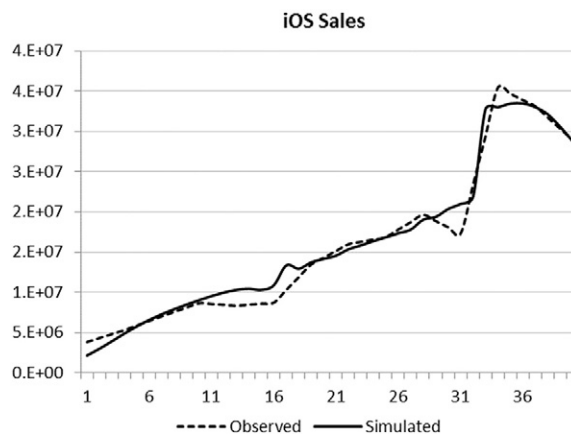


Fig. 11. Observed and simulated values of iOS adoption.

Android/iOS blogs, websites, developer agreements and popular tech websites. We sifted through 200+ articles on various tech websites to search, validate and understand all the policy changes that took place in the evolution of Android and iOS. For the user experience of different versions of Android and iOS, we used the user ratings available on CNET as a proxy for the same.

To calibrate the model, we used the automatic calibration feature of the Vensim® package,<sup>2</sup> and used the three-point heuristic proposed by Oliva (2003) to guide the process.

- I. Include all available knowledge about system parameters in the calibration process.
- II. Apply automatic calibration to the simplest possible calibration problem.
- III. Use automatic calibration to test the hypothesis “The estimated parameter matches the observable structure of the system.”

The model was calibrated by minimising two cumulative error terms (i) the sum of absolute differences between actual and simulated Android Adoption values over forty months and (ii) sum of absolute differences between actual and simulated iOS Adoption over the same forty data points. The parameters to be calibrated were  $K_{\text{android},0}$ ,  $K_{\text{android},ij}$  ( $i = 1, 2; j = 1 \dots 5$ ),  $K_{\text{ios},0}$ ,  $K_{\text{ios},km}$  ( $k = 1, 2; m = 1 \dots 3$ ), Market Growth Fraction and initial value of Potential Adopters. The optimization feature of Vensim Professional was used during calibration. It uses a hill climbing algorithm and completed the calibration in 15,794 runs. A mean absolute percentage error (MAPE) of 8.92% and 9.84% was obtained for Android and iOS sales, respectively. Figs. 10 and 11 show post-calibration simulated values of Android Adoption and iOS Adoption against their respective observed handset sales values.

#### 4.4. Evaluation of calibrated model

To evaluate the calibrated model, we ran it for a fairly long span from May 2012 to April 2016 – i.e. period 40 to 87 – and compared the simulated results to actual monthly sales data for Android and iOS handsets for that period. During this period, breakpoints were not identified and consequently that information was not used in altering any model parameters. The results are shown in Figs. 12 and 13, and the equations for the calibrated model are shown in Appendix A - Table 11.

<sup>2</sup> Vensim is a simulation software created and distributed by Ventana Systems Inc., [www.vensim.com](http://www.vensim.com). The calibration feature is available in Vensim Professional and Vensim DSS.

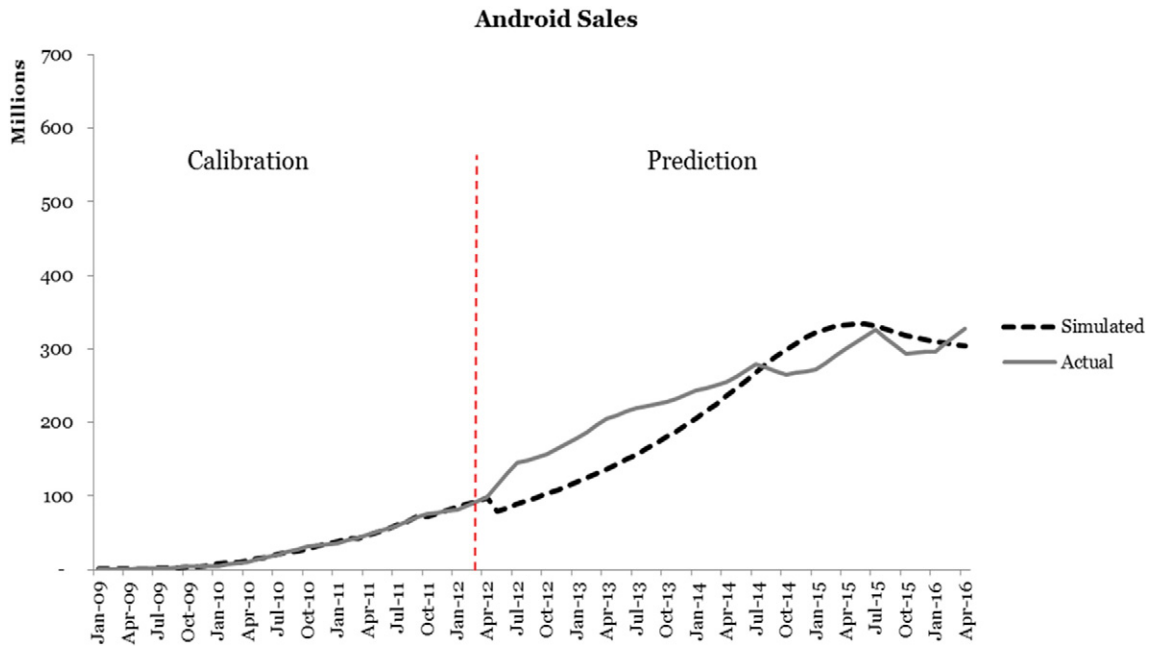


Fig. 12. Observed and simulated values of Android adoption (in calibration and prediction phase).

The MAPE for the predicted values are shown in Table 6 where they are also compared with the calibration phase. Not unexpectedly, error performance is poorer in the prediction phase compared to the calibration phase. However, two interesting observations can be made regarding the simulated versus actual values graphs for May 2012 to April 2016. First, by comparing the MAPE of the calibration and prediction phases, the significance of using structural break information in modeling IT diffusion becomes clear. We deliberately did not use this information during the prediction phase because the significance of events associated with structural breaks often becomes evident only after some passage of time. Second, the causal model was able to predict the general slowdown and decline in sales pattern for both platforms – note the dip after about January 2015. While statistical models, such as moving average, can quantitatively fit these patterns, they do not give us much insight as to what caused them. However, our causal model is able to capture changes in the underlying mechanisms over time as elaborated on in the next section.

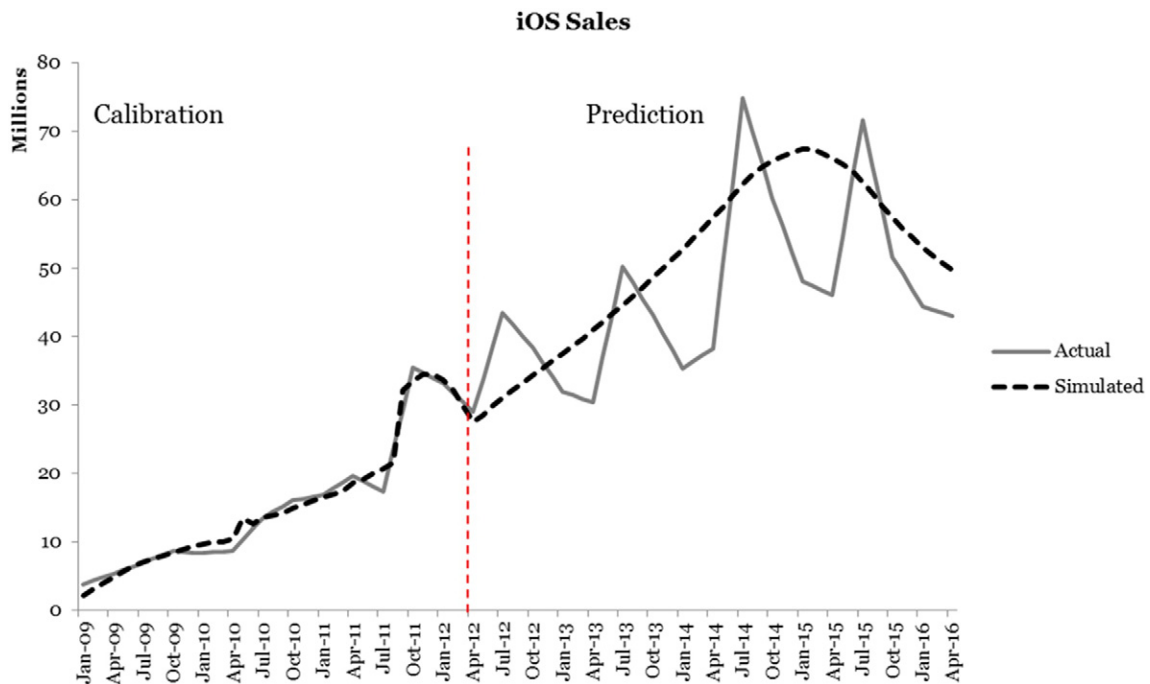


Fig. 13. Observed and simulated values of iOS adoption (in calibration and prediction phase).

**Table 6**  
Comparison of fit for Android and iOS across the two phases.

	Android	iOS
Calibration	MAPE: 8.92% Correlation: 0.999	MAPE: 9.19% Correlation: 0.991
Prediction	MAPE: 23.41% Correlation: 0.959	MAPE: 19.93% Correlation: 0.684

**5. Deconstructing the mechanics of handset diffusion**

The causal model presented in Fig. 8 posited that the diffusion of Android and iOS handsets can be explained with a set of mechanisms where the following held true.

- a) Contagion mechanism is at the core of diffusion. A section of *Potential Adopters* join the pool of *Adopters* as a result of information exchange between these two groups.
- b) The time rate of change of *Adopters* pool for Android or iOS is proportional to the number of *Potential Adopters* and *Adopters* for the corresponding mOS. The proportionality constant *Contagion Strength*, modulates the pace of change.
- c) *Contagion Strength* is a time-varying parameter whose magnitude is moderated by Network Effects (L13–14 & L23–24) as well as the Supply Effects (L15–16 & L25–26). Recall that these loops model the mutual dependence of *Adopters*, *Adoption Rate* and *Contagion Strength*, and can result in either exponential or logarithmic diffusion patterns over different segments of the period of analysis, depending on the parameter values.

To gain a deeper understanding of the shifting patterns of diffusion of Android and iOS handsets we would need to extend the core contagion mechanism and include the feedback mechanisms that result in changes in *Contagion Strength* over time. The calibrated simulation model provides empirical evidence of these temporal changes in parameter values, which can then be used to get a more detailed understanding of changes in the underlying *mechanisms* that drive diffusion behavior. This analysis is presented in the next two sections.

**5.1. Characterization of feedback loops**

The simplified causal model in Fig. 8 had left the polarities of L13–14, L23–24, L15–16 and L25–26 ambiguous since their net effect could impact diffusion in both positive and negative ways. The simulation model was built such that signs of associated parameters would determine the polarity of respective loops. The calibration process described above resulted in the loop polarities shown in Tables 7 and 8. The signs of the parameters  $K_{Android,1}$ ,  $K_{Android,2}$ ,  $K_{iOS,1}$ ,  $K_{iOS,2}$  during different time intervals are shown alongside.

Tables 7 and 8 show how the nature of Network Effects (L13–14 & L23–24) and Supply Effects (L15–16 & L25–26) changed during different time intervals. The changing polarities of the loops – reinforcing or

**Table 7**  
Polarity of L13–14 and L15–16 across time intervals.

Interval	$K_{Android,1}$	Polarity of L13–14	$K_{Android,2}$	Nature of L15–16
$0 \leq t < 14$	Positive	Reinforcing	Negative	Balancing
$14 \leq t < 20$		Reinforcing	Negative	Balancing
$20 \leq t < 26$	Positive	Reinforcing	Negative	Balancing
$26 \leq t < 34$		Reinforcing	Positive	Reinforcing
$34 \leq t$	Negative	Balancing	Negative	Balancing
	Positive	Reinforcing		

**Table 8**  
Polarity of L23–24 and L25–26 across time intervals.

Interval	$K_{iOS,1}$	Nature of L23–24	$K_{iOS,2}$	Nature of L25–26
$0 \leq t < 16$	Negative	Balancing	Negative	Balancing
$16 \leq t < 32$		Balancing	Positive	Reinforcing
$32 \leq t$	Negative	Balancing	Positive	Reinforcing

balancing – offers insight into changes in underlying mechanisms that occurred over the course of the diffusion. For example, in Table 7 we see that during the first 14 months, L13–14 behaved as a reinforcing loop implying that positive effects of the expanding pool of *Android Adopters* were more pronounced than negative effects. In contrast, during interval 14–20, the same loop behaved as a balancing loop, indicating the opposite net effect. During the interval 0–14, L15–16 behaved like a balancing loop implying that the negative effects of increasing *Android Adoption* (e.g. variability across handset models) were more pronounced than the positive effects (falling price, feature improvement). In the next section, we combine this loop characterization with loop dominance patterns to get a more complete picture of the diffusion mechanisms.

**5.2. Analysis of loop dominance**

The basic structural component in an SD model is the feedback loop, and a model typically consists of numerous loops. To get a parsimonious understanding of the mechanics, it is useful to identify a limited subset of loops which have the most impact on behaviour. We use the formal method of loop dominance analysis (LDA) for this purpose. Theoretical underpinnings of this technique may be found in the literature (Ford, 1999; Kampmann, 1996; Mojtahedzadeh, 1997; Richardson, 1996). We adopted Ford’s technique because of its intuitive appeal and ease of implementation. Ford’s method determines loop dominance by designating one model variable as the Variable of Interest (VoI). A feedback loop is said to dominate if deactivating it, without changing model structure and other system parameters, alters the pattern of VoI’s behavior in any time interval. The behavior of a system variable can be viewed as a sequence of atomic behaviour patterns, which can be one of three types namely Linear (VoI grows or decays at a constant rate); Exponential (VoI grows or decays at an increasing rate); Logarithmic (VoI grows or decays at a decreasing rate).

We carried out loop dominance analysis twice, with *Android Adoption* and *iOS Adoption* as the VoI, respectively. For *Android Adoption* we checked dominance of L11, L12, L13–14 and L15–16 while for *iOS Adoption* loops L21, L22, L23–24 and L25–26 were checked. Appendix B – Fig. 14 shows results of the analysis with *Android Adoption* as the VoI. For completeness, we also show polarities of L13–14 and L15–16 presented earlier in Table 7. Dominance results for *iOS Adoption* appear in Appendix B – Fig. 15 along with polarities of L23–24 and L25–26.<sup>3</sup>

Ideally a platform owner would like growth to be dominated by reinforcing loop(s). However, external events can switch dominance to balancing loop(s) that slow down growth to a logarithmic form. By sensing such occurrences, one can intervene appropriately such that dominance switches back to reinforcing loop(s). The growth for Android was exponential until Sep '10 during which two reinforcing loops, L11 the Word-of-Mouth loop, and L13–14, the network externality loop, took turns in boosting growth. Loop L13–14, however, behaved as a balancing loop from Jan '10. In October '10, Android experienced a

<sup>3</sup> Since these two figures require a different page orientation these have been presented as Appendices.

**Table 9**  
Growth type and dominant loop in Android diffusion.

	Growth type	
	Exponential	Logarithmic
Proportion of periods showing growth type	79.49%	20.51%
(a) Network effect dominant within growth type	25.81%	37.50%
(b) Supply effect dominant within growth type	12.90%	62.50%
(c) Word-of-mouth effect dominant within growth type	61.29%	0.00%
(d) Market saturation effect dominant within growth type	0.00%	0.00%
Total of (a), (b), (c) and (d)	100.00%	100.00%

Malware Attack and the growth pattern immediately became logarithmic. Our analysis shows that the network externality loop was dominating Android diffusion at this time. This could mean that potential buyers perceived the large installed base to be a strong attractor for hackers, resulting in a negative effect on the buying decision. Developers also had similar perception about the increasing installed base while choosing the platform to develop apps. The loop at this point was a balancing one, implying the negative effect of increasing installed base. Thereafter, the growth started stagnating.

The next disruption in diffusion happened in April 2011, which was about the time Apple went to court against Samsung. Dominance switched to the L15–16 loop and the growth pattern became logarithmic. Google released a few improvements in May '11. Dominance switched to L13–15 and the growth pattern became exponential again. In Oct 2011, Android announced the Ice Cream Sandwich (4.0.0) operating system. The market was euphoric but there were concerns regarding compatibility (Nickinson, 2011). During October '11–March '12 Google released three updates for Ice Cream Sandwich (4.0.2–4.0.4) to incorporate bug fixes, performance improvements and compatibility with manufacturer specific handsets (e.g. minor bug fixes for Samsung Nexus in Oct '11 and minor bug fixes for Verizon Galaxy Nexus in Nov '11).<sup>4</sup> Of and on, during this period, concerns about the new OS shifted dominance to the loop L15–16. But Google was quick to launch several upgrades which shifted dominance to L13–14.

In the case of iOS, interventions by Apple that resulted in dominance shifting to reinforcing loop(s) have mostly been short lived, spanning one or two months. We surmise this has occurred as a result of the constraint that the pool of potential iPhone/iPad buyers has imposed on the diffusion of iOS. iPhone was initially available in only a few countries and its growth until May 2010 had been logarithmic, dominated throughout by balancing loop(s). However, when Apple launched the products in other countries (Sep '10, Mar '11, Nov '11), dominance shifted to reinforcing loops, albeit for a short while. Similar shift also occurred when features were added to iOS (Jun '11). Notably, in the case of intervention leading to market expansion, the shift happened almost immediately, whereas in the case of feature enhancement the effect came after some time.<sup>5,6</sup>

Table 9 summarizes the growth type and loop dominance observed for Android diffusion. It shows that in about 79% of the periods under study, *Android Adoption* grew at an increasing rate. In 61% of such exponential growth periods, growth came from word-of-mouth effect. In about 21% of the periods the growth was logarithmic (increasing at a decreasing rate) and Supply Effect dominated the system behaviour. Interestingly, system behaviour was never dominated by Market Saturation effect (L12), which leads us to conclude that up until the end of the study period, the Android Market is yet to be saturated.

<sup>4</sup> Wikipedia Page on Android Version History, [https://en.wikipedia.org/wiki/Android\\_version\\_history](https://en.wikipedia.org/wiki/Android_version_history) accessed 18th January 2016.

<sup>5</sup> iOS Version History Wikipedia page, [https://en.wikipedia.org/wiki/iOS\\_version\\_history](https://en.wikipedia.org/wiki/iOS_version_history) accessed 18th January 2016.

<sup>6</sup> Apple Press Releases, <http://www.apple.com/pr/library/2011/>, accessed 18th January 2016.

**Table 10**  
Growth type and dominant loop in iOS diffusion.

	Growth type	
	Exponential	Logarithmic
Proportion of periods showing growth type	38.46%	61.54%
Network effect dominant within growth type	20.00%	16.67%
Supply effect dominant within growth type	6.67%	41.67%
Word-of-mouth effect dominant within growth type	73.33%	0.00%
Market saturation effect dominant within growth type	0.00%	41.76%
Total of (a), (b) (c) and (d)	100.00%	100.00%

One noteworthy observation in the case of iOS diffusion, compared to Android, is the shorter spells of dominance. The average span of dominance (number of time intervals/number of change overs) is 1.82 periods per change-over in comparison to 3.33 Android. Table 10 summarises the Loop Dominance findings for iOS diffusion. Growth in *iOS Adoption* was predominantly logarithmic (increasing at a decreasing rate), since about 61% of the periods exhibited this behaviour. During these periods of logarithmic growth, Supply Effect dominated system behaviour about 42% of the time, and Market Saturation about 42% of the time. Supply Effect did not dominate the periods of exponential growth in any significant way. The periods of exponential growth were mostly (73%) dominated by positive word-of-mouth. Network Effect had limited influence in energising exponential growth pattern as indicated by the 20% number for the proportion of periods in which it was dominant. The fact that exponential growth pattern accounted for only 38% of the periods analysed, points to more sluggish adoption of iOS compared to Android.

Comparison of the loop dominance patterns of *Android Adoption* and *iOS Adoption* offers insight into why, in spite of being the early mover, iOS was overtaken by Android.

- iOS diffusion, in the beginning, had a short spell (3 periods) of exponential growth followed by a longer spell (14 periods) of slower logarithmic growth. But Android experienced a long initial spell (23 periods) of exponential growth followed by a short spell (3 periods) of slower logarithmic growth. This helped it to catch up and overtake iOS handset sales.
- Supply Effect acted both as an inhibitor (supply shortage, usage issues) and an enabler (price cuts, feature enhancement). The inhibitor effect was relatively more pronounced compared to the enabler effect for both platforms, as is evident from Tables 9 and 10. Popular wisdom held that Android diffusion was spurred mainly by handset price cuts and feature enhancements (Seltzer et al., 2015) – i.e. Supplier effect. But the results in Tables 9 and 10 show that was not the case and word of mouth was the dominant driving mechanism for the exponential growth in both platforms.
- Network Effects also helped drive exponential growth in both platforms. It therefore appears that the growing pool of complementary goods (Apps) facilitated *Adoption* although the impact was more pronounced (12% of periods in exponential growth) in the case of *Android Adoption* compared to *iOS adoption* (8%).
- Market Saturation constrained growth only in the case of *iOS Adoption*.

Overall, this analysis of loop dominance leads us to infer that iOS exploited its market in bursts while Android exploited the market in a more sustained manner. For both platforms, the pattern changed to pairs of short spell of exponential growth followed by short spell of logarithmic growth reflecting a series of S-shaped diffusion patterns. However, an initial sustained spell of exponential growth boosted *Adoption* towards building a very large pool of *Android Adopters* and subsequent sustained market dominance.

6. Conclusion

Despite a rich body of literature, IT diffusion continues to interest researchers (Kauffman and Techatassanasoontorn, 2012). One thread in this body of work focuses on modeling the temporal pattern of diffusion. Our study makes a contribution by addressing two issues which have received insufficient attention in this stream of work. First, due to the nature of technology innovation, the parameters driving an IT diffusion phenomenon need not remain constant over the duration of analysis as is commonly assumed. Hence, one needs to detect and allow for possible shifts in these driving parameters. Second, there are circumstances when the descriptive approach of characterizing the temporal diffusion pattern as  $y(t) = f(t)$ , however elaborate  $f(t)$  might be, is not enough. Rather, in situations where managerial interventions are contemplated to deliberately affect the diffusion process, one needs to know the underlying mechanics of the diffusion process. The above approach was demonstrated by studying the diffusion of handsets based on two popular mobile operating systems platforms, Android and iOS.

We addressed these two issues with a 2-phase approach, first detecting statistically significant breakpoints, followed by a second phase where a systems dynamics based model was developed and calibrated using the breakpoints determined in the previous phase. Loop dominance analysis of the model provided insights into the mechanics of diffusion and how it changes over time. One revealing illustration of the benefits of a causal approach could be seen during the model evaluation shown in Figs. 12 and 13. There, the simulated graphs for both iOS and

Android show an exponential growth pattern for a long time followed by saturation and decline. Examination of the feedback loops shows that the saturation and decline occurred due to slowdown in market expansion. Once we know this cause, interventions can be contemplated, such as developing ultra-cheap handsets with very simple functionality (Economist, 2014), that could reach people at the 'bottom of the pyramid', which in turn would expand the market considerably along with the pursuit of new growth opportunities that come from expanding the mOS reach to other devices including wearables (Gartner, 2017).

Both strengths and limitations of our study are a consequence of characteristics of the System Dynamics method. It offers a powerful conceptual construct – the feedback loop – to identify mechanisms underlying a temporal phenomenon such as diffusion. Understanding these mechanisms can help engineer purposeful interventions to change the observed diffusion behaviour. However, SD models are weak at capturing heterogeneity. For example, adopters and handset manufacturers will exhibit heterogeneity in their defining characteristics. In capturing the complexity of interactions among different factors, the richness with which each factor is characterized is simplified. Other techniques such as agent based modelling are capable of modelling heterogeneity (Kiesling et al., 2011), but do not offer constructs to capture the underlying causal structure that is generating observed behaviour. We surmise that an SD based approach will be applicable to other technology diffusion settings. One that comes to mind is adoption of cloud computing, where potential cost reductions act as reinforcing influences while security concerns act in the opposite direction.

Appendix A

Table 11  
Equations of calibrated model.

And CS $K_0$	0.0490719
And CS $K_1$	$(AndK_{11} * Pulse(0,14) + AndK_{12} * Pulse(14,6) + AndK_{13} * Pulse(20,6) + AndK_{14} * Pulse(26,8) + AndK_{15} * Pulse(34,6))$
And CS $K_2$	$(AndK_{21} * Pulse(0,14) + AndK_{22} * Pulse(14,6) + AndK_{23} * Pulse(20,6) + AndK_{24} * Pulse(26,8) + AndK_{25} * Pulse(34,6))$
And $K_{11}$	32.8508
And $K_{12}$	-0.225999
And $K_{13}$	-0.613815
And $K_{14}$	0.32844
And $K_{15}$	2.23803
And $K_{21}$	-1041.11
And $K_{22}$	61.1013
And $K_{23}$	36.5481
And $K_{24}$	-46.8698
And $K_{25}$	-216.549
Android adopters churn	Android adopters/average handset life Android
Android adoption	$Android\ Contagion\ Strength * Android\ Adopters * Potential\ Adopters / (Android\ Adopters + Potential\ Adopters + iOS\ Adopters)$
Android contagion strength	$EXP(And\ CS\ K_0 + Android\ Adopters / 1e + 09 * And\ CS\ K_1 + Average\ Android\ Adoption/1e + 10 * And\ CS\ K_2)$
Average android adoption	$SMOOTH(Android\ Adoption, 1,575,300)$
Android adopters	$INTEG(Android\ Adoption - Android\ Adopters\ Churn, 575,300)$
Average handset life Android	33.1285
Average handset life iOS	19.6363
Average iOS adoption	$SMOOTH(iOS\ Adoption, 2.385e + 06)$
Initial potential adopters	$6.60773e + 06$
iOS adopters	$INTEG(iOS\ Adoption - iOS\ Adopters\ Churn, 3.8481e + 06)$
iOS adopters churn	$iOS\ Adopters / Average\ Handset\ life\ iOS$
iOS adoption	$iOS\ Contagion\ Strength * iOS\ Adopters * Potential\ Adopters / (iOS\ Adopters + Potential\ Adopters + Android\ Adopters)$
iOS contagion strength	$EXP(iOS\ CS\ K_0 + iOS\ Adopters / 1e + 10 * iOS\ CS\ K_1 + Average\ iOS\ Adoption/1e + 10 * iOS\ CS\ K_2)$
iOS $K_{11}$	-61.9317
iOS $K_{12}$	79.992
iOS $K_{13}$	-12.4662
iOS $K_{21}$	-66.5437
iOS $K_{22}$	-1151.53
iOS $K_{23}$	108.52
Market expansion	$iOS\ Adopters\ Churn + Android\ Adopters\ Churn + Market\ Growth\ Fraction * Potential\ Adopters$
Market growth fraction	$0.589785 * Multiplier\ from\ Market\ Size / ((Android\ Adopters + iOS\ Adopters + Potential\ Adopters) / 9.0E09)$
Potential adopters	$INTEG((Market\ Expansion - Android\ Adoption - iOS\ Adoption), Initial\ Potential\ Adopters)$

Appendix B

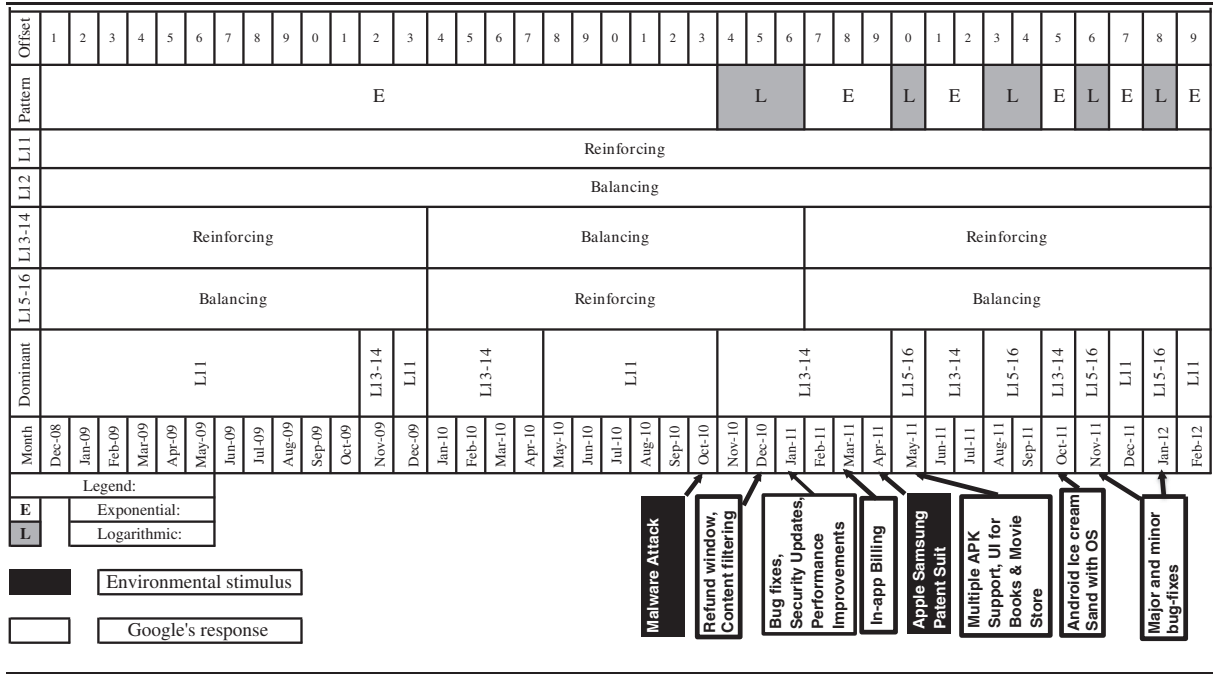


Fig. 14. Loop dominance in Android adoption.

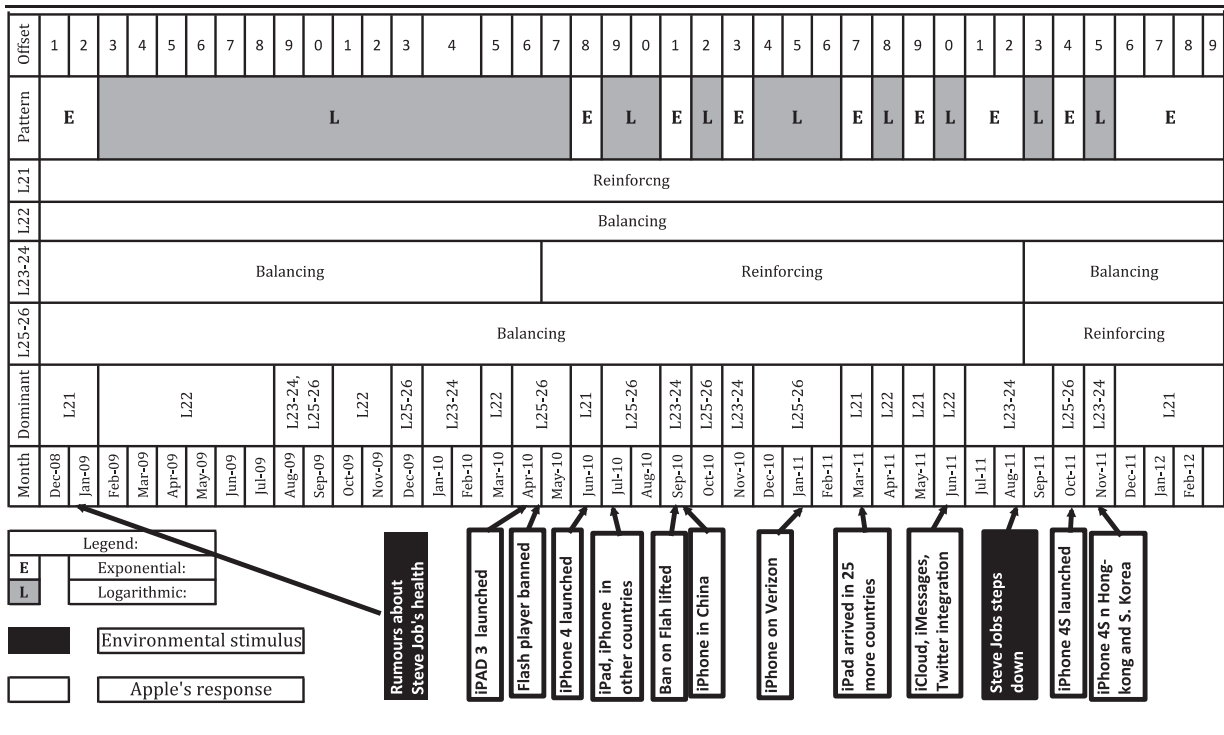


Fig. 15. Loop dominance in iOS adoption.

## References

- Andrews, D.W.K., 2003. Tests for parameter instability and structural change with unknown change point: a corollary. *Econometrica* 71 (1):395–397. <http://dx.doi.org/10.1111/1468-0262.00405>.
- Bai, J., Perron, P., 2003. Critical values for multiple structural change tests. *Econ. J.* 6 (1): 72–78. <http://dx.doi.org/10.1111/1368-423X.00102>.
- Baskerville, R., Pries-Heje, J., 1998. Information technology diffusion: building positive barriers. *Eur. J. Inf. Syst.* 7 (1):17–28. <http://doi.org/Article>.
- Bass, F.M., 1969. A new product growth for model consumer durables. *Manag. Sci.* 15 (5): 215–227. <http://dx.doi.org/10.1287/mnsc.1040.0264>.
- Bohn, D., Souppouris, A., Seifert, D., 2013. iOS: A Visual History. Retrieved January 18, 2016, from <http://www.theverge.com/2011/12/13/2612736/ios-history-iphone-ipad>.
- Burgelman, R., Silverman, A., Wittig, C., Hoyt, D., 2009. Google's Android: Will It Shake Up the Wireless Industry in 2009 and Beyond? Retrieved January 18, 2016, from <http://www.gsb.stanford.edu/faculty-research/case-studies/googles-android-will-it-shake-wireless-industry-2009-beyond>.
- Burnham, K.P., Anderson, D.R., 2004. Multimodel inference: understanding AIC and BIC in model selection. *Sociol. Methods Res.* 33 (2), 261–304.
- Casey, T.R., Töyli, J., 2012. Mobile voice diffusion and service competition: a system dynamic analysis of regulatory policy. *Telecommun. Policy* 36 (3):162–174. <http://dx.doi.org/10.1016/j.telpol.2011.07.002>.
- Ceccagnoli, M., Forman, C., Huang, P., Wu, D.J., 2012. Cocreation of value in a platform ecosystem: the case of enterprise software. *MIS Q.* 36 (1), 263–290.
- Cho, S., Choi, Y.-J., 2015. Effect of leased line price on broadband penetration in OECD countries with implications on network neutrality. *Int. Telecommun. Pol. Rev.* 22 (2), 109–128.
- Choi, J., Nazareth, D.L., Jain, H.K., 2010. Implementing service-oriented architecture in organizations. *J. Manag. Inf. Syst.* 26 (4):253–286. <http://dx.doi.org/10.2753/MIS0742-1222260409>.
- Chow, G.C., 1960. Tests of equality between sets of coefficients in two linear regressions. *Econometrica* 28 (3):591–605. <http://dx.doi.org/10.2307/1910133>.
- Coughlan, P., 2004. The Golden Age of Home Video Games: From the Reign of Atari to the Rise of Nintendo. Retrieved January 18, 2016, from <https://hbr.org/product/the-golden-age-of-home-video-games-from-the-reign-of-atari-to-the-rise-of-nintendo/704487-PDF-ENG>.
- Daim, T., Rueda, G., Martin, H., Gerdri, P., 2006. Forecasting emerging technologies: use of bibliometrics and patent analysis. *Technol. Forecast. Soc. Chang.* 73 (8), 981–1012.
- Dutta, A., Roy, R., 2005. Offshore outsourcing: a dynamic causal model of counteracting forces. *J. Manag. Inf. Syst.* 22 (2), 15–35.
- Economides, N., 1996. Network externalities, complementarities, and invitations to enter. *Eur. J. Polit. Econ.* 12 (2):211–233. [http://dx.doi.org/10.1016/0176-2680\(95\)00014-3](http://dx.doi.org/10.1016/0176-2680(95)00014-3).
- Economist, 2014. The Rise of the Cheap Smartphone. Retrieved January 12, 2017, from <http://www.economist.com/news/business/21600134-smartphones-reach-masses-host-vendors-are-eager-serve-them-rise-cheap>.
- Evans, D.S., Schmalensee, R., 2010. Failure to launch: critical mass in platform businesses. *Rev. Netw. Econ.* 9 (4):1–26. <http://dx.doi.org/10.2202/1446-9022.1256>.
- Fichman, R., 1992. Information technology diffusion: a review of empirical research. *Proceedings of the 1992 International Conference on Information Systems* (p. Paper 39). Dallas, TX Retrieved from <http://aisel.isnet.org/cgi/viewcontent.cgi?article=1021&context=icis1992>.
- Fichman, R., 2004. Going beyond the dominant paradigm for information technology innovation research: emerging concepts and methods. *J. Assoc. Inf. Syst.* 5 (8), 314–355.
- Ford, D., 1999. A behavioral approach to feedback loop dominance analysis. *Syst. Dyn. Rev.* 15 (1):3–36. [http://dx.doi.org/10.1002/\(SICI\)1099-1727\(199921\)15:1<3::AID-SDR159>3.0.CO;2-P](http://dx.doi.org/10.1002/(SICI)1099-1727(199921)15:1<3::AID-SDR159>3.0.CO;2-P).
- Forrester, J.W., 1961. *Industrial Dynamics*. MIT Press, Cambridge, Massachusetts.
- Forrester, J.W., 1994. System dynamics, systems thinking, and soft OR. *Syst. Dyn. Rev.* 10 (2–3), 245–356.
- Garcia, R., Calantone, R., Levine, R., 2003. The role of knowledge in resource allocation to exploration versus exploitation in technologically oriented organizations. *Decis. Sci.* 34 (2), 323–349.
- Gartner, 2017. Gartner Smart Phone Marketshare. Retrieved January 12, 2017, from <http://www.gartner.com> (Multiple Pages).
- Gary, M.S., Dosi, G., Lovallo, D., 2008. Boom and bust behavior: on the persistence of strategic decision biases. In: Hodgkinson, G.P., Starbuck, W.H. (Eds.), *The Oxford Handbook of Organizational Decision Making*. Oxford University Press, Oxford:pp. 33–55 <http://dx.doi.org/10.2139/ssrn.1015817>.
- German, K., 2011. A Brief History of Android Phones. Retrieved January 18, 2016, from <http://www.cnet.com/news/a-brief-history-of-android-phones/>.
- Greenan, C.C., 2015. Diffusion of innovations in dynamic networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 178 (1):147–166. <http://dx.doi.org/10.1111/rssa.12054>.
- Gujarati, D.N., 2004. *Basic Econometrics*, fourth ed. Tata McGraw Hill, New Delhi, India.
- Gupta, R., Jain, K., 2014. Adoption of mobile telephony in rural India: an empirical study. *Decis. Sci.* 45 (2), 281–307.
- Gurbaxani, V., 1990. Diffusion in computing networks: the case of BITNET. *Commun. ACM* 33 (12), 65–75.
- Halaburda, H., Gans, J., Burbank, N., 2010. Developing an App for that: Mobile Application Strategy. Retrieved January 18, 2016, from <https://hbr.org/product/developing-an-app-for-that/711415-PDF-ENG>.
- Hsu, C., Lu, H., Hsu, H., 2007. Adoption of the mobile Internet: an empirical study of multimedia message service (MMS). *Omega* 35 (6), 715–726.
- JDPower, 2013. U.S. Wireless Smartphone Satisfaction Study—Volume 2. Retrieved January 18, 2016, from <http://www.jdpower.com/press-releases/2013-us-wireless-smartphone-satisfaction-study-volume-2>.
- Kampmann, C., 1996. Feedback loop gains and system behavior. *Syst. Dyn. Rev.* 28 (4), 370–395.
- Kauffman, R., Techatassanasontorn, A., 2009. Understanding early diffusion of digital wireless phones. *Telecommun. Policy* 33 (8), 432–450.
- Kauffman, R., Techatassanasontorn, A., 2012. Guest editors' introduction: methods innovations for the empirical study of technology adoption and diffusion. *Inf. Technol. Manag.* 13 (3), 113–114.
- Kiesling, E., Günther, M., Stummer, C., Wakolbinger, L.M., 2011. Agent-based simulation of innovation diffusion: a review. *CEJOR* 20 (2):183–230. <http://dx.doi.org/10.1007/s10100-011-0210-y>.
- Koch, S., Kerschbaum, M., 2014. Joining a smartphone ecosystem: application developers' motivations and decision criteria. *Inf. Softw. Technol.* 56 (11):1423–1435. <http://dx.doi.org/10.1016/j.infsof.2014.03.010>.
- Lakka, S., Michalakelis, C., 2012. Exploring the determinants of the OSS market potential: the case of the Apache web server. *Telecommun. Policy* 36 (1), 51–68.
- Lakka, S., Michalakelis, C., Varoutas, D., Martakos, D., 2013. Competitive dynamics in the operating systems market: modeling and policy implications. *Technol. Forecast. Soc. Chang.* 80 (1):88–105. <http://dx.doi.org/10.1016/j.techfore.2012.06.011>.
- Levin, S.G., Stephan, P.E., Winkler, A.E., 2012. Innovation in academe: the diffusion of information technologies. *Appl. Econ.* 44 (14):1765–1782. <http://dx.doi.org/10.1080/00036846.2011.554375>.
- Loch, C.H., Huberman, B.A., 1999. A punctuated-equilibrium model of technology diffusion. *Manag. Sci.* 45 (2):160–177. <http://dx.doi.org/10.1287/mnsc.45.2.160>.
- Meade, N., Islam, T., 2006. Modelling and forecasting the diffusion of innovation — a 25-year review. *Int. J. Forecast.* 22 (3), 519–545.
- Mojtahedzadeh, M.T., 1997. *A Path Taken: Computer-assisted Heuristics for Understanding Dynamic Systems*. State University of New York, Albany (January).
- Morecroft, J.D.W., 1983. A systems perspective on materials requirement planning. *Decis. Sci.* 14 (1), 1–18.
- Nickinson, P., 2011. The Top Android News of 2011. Retrieved January 18, 2016, from <http://www.androidcentral.com/top-android-news-2011>.
- Oliva, R., 2003. Model calibration as a testing strategy for system dynamics models. *Eur. J. Oper. Res.* 151 (3), 552–568.
- Padhi, S.S., Wagner, S.M., Mohapatra, P.K.J., 2016. Design of auction parameters to reduce the effect of collusion. *Decision Sciences* 47 (6):1016–1047. <http://dx.doi.org/10.1111/dec.12159>.
- Paich, M., Sterman, J.D., 1993. Boom, bust, and failures to learn in experimental markets. *Manag. Sci.* 39 (12):1439–1458. <http://dx.doi.org/10.1287/mnsc.39.12.1439>.
- Park, S., 2004. Quantitative analysis of network externalities in competing technologies: the VCR case. *Rev. Econ. Stat.* 86 (4):937–945. <http://dx.doi.org/10.1162/0034653043125275>.
- Rai, A., Ravichandran, T., Samaddar, S., 1998. How to anticipate the Internet's global diffusion. *Commun. ACM* 41 (10), 97–106.
- Richardson, G., 1996. Problems for the future of system dynamics. *Syst. Dyn. Rev.* 12 (2), 141–157.
- Rogers, E.M., 1962. *Diffusion of innovations*. The Free Press, New York.
- Scaglione, M., Giovannetti, E., Hamoudia, M., 2015. The diffusion of mobile social networking: exploring adoption externalities in four G7 countries. *Int. J. Forecast.* 31 (4):1159–1170. <http://dx.doi.org/10.1016/j.ijforecast.2015.03.005>.
- Seltzer, M., Siegel, R., Burgelman, R., 2015. Google and Android in 2015 (C): Looking Towards the Future. SM-176-C. Stanford University Case.
- Soydan, A.I., Oner, M.A., 2012. Timely resource allocation between R&D and marketing: a system dynamics view. *Int. J. Innov. Technol. Manag.* 9 (2).
- Statista, 2016. Global Smartphones Installed Base 2008–2017. Retrieved January 18, 2016, from <http://www.statista.com/statistics/371889/smartphone-worldwide-installed-base/>.
- Sterman, J., 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. McGraw Hill, Irwin, CA.
- Susarla, A., Oh, J., Tan, Y., 2012. Social networks and the diffusion of user-generated content: evidence from YouTube. *Inf. Syst. Res.* 23 (1), 23–41.
- Watanabe, C., Moriyama, K., Shin, J.-H., 2009. Functionality development dynamism in a diffusion trajectory: a case of Japan's mobile phones development. *Technol. Forecast. Soc. Chang.* 76 (6):737–753. <http://dx.doi.org/10.1016/j.techfore.2008.06.001>.
- Werden, G., 2001. Network effects and conditions of entry: lessons from the Microsoft case. *Antitrust Law J.* 69 (1), 87–111.

**Amitava Dutta** is the LeRoy Eakin chair and professor of information systems and operations management at the School of Business at the George Mason University, Fairfax Virginia; **Abhinav Puvvala** is a fellow of Indian Institute of Management (IIM) Calcutta and is currently with Tata Consultancy Services; **Rahul Roy** and **Priya Seetharaman** are faculty members with the MIS Group at IIM Calcutta. All four authors have an interest in using the system dynamics methodology to model time dependent phenomena that involve an interaction between technology and individuals or organizations.