Forking, Fragmentation, and Splintering

Timothy Simcoe
Boston University, Questrom School of Business and NBER

Jeremy Watson
Boston University, Questrom School of Business

September 27, 2017

Forking, Fragmentation, and Splintering

Abstract

Although economic theory suggests that markets may "tip" towards a dominant platform or standard, there are many prominent examples of persistent incompatibility, inter-platform competition and standards proliferation. This paper examines the phenomena of forking, fragmentation and splintering in markets with network effects. We illustrate several causes of mis-coordination, as well as the tools that firms and industries use to fight it, through short cases of standardization in railroad gauges, modems, operating systems, instant messaging and Internet browsers. We conclude by discussing managerial implications and the potential welfare effects of efforts to promote inter-operability.

Keywords: Compatibility, standards, network effects, platforms, forking.

JEL Codes: L15, Q55, Q58

Google, Inc. provided financial support for this research. All views expressed herein are solely those of the authors. We thank Daniel Gross, Rebecca Karp, Michael Schrage and Joel West for helpful comments and discussion. Contact: tsimcoe@bu.edu, jwats@bu.edu. Boston University Questrom School of Business, 595 Commonwealth Avenue, Boston, MA 02215.

1 Introduction

Standards, platforms and protocols are defining features of the digital economy. By adhering to pre-defined rules such as file formats, communications protocols or programming languages, independently designed products can work together well. The resulting interoperability promotes communication amongst a large network of users, and access to a wide range of complementary products and services. Interoperability can also produce complex patterns of technological interdependence, leading some scholars to liken the information technology sector to a natural ecosystem where firms in distinct niches both cooperate and compete with one another (Adner and Kapoor, 2010; Gawer and Cusumano, 2014; Parker et al., 2017)

In economic models, interoperability benefits can create network effects that cause markets to "tip" towards a dominant standard (Katz and Shapiro, 1985; Farrell and Saloner, 1986; Arthur, 1989). But in practice, there are many cases of persistent incompatibility, inter-platform competition and standards proliferation. For example, U.S. cell phones would not work in Europe for many years because European carriers adopted different transmission standards. Similarly, modern web browsers support dozens of audio and video file formats, while smartphone users can choose among incompatible platforms for ride sharing, instant messaging and music streaming.

Technologists use terms like forking, fragmentation and splintering to describe markets characterized by a variety of competing standards or platforms. Advocates for widespread inter-operability often view "failure to tip" as evidence of inefficiency (Farrell, 2007). That is, they believe that having fewer standards is generally better, but markets often fail to produce an interoperable outcome. Meanwhile, advocates for laissez-faire standards policy argue that – with a dose of help from platform leaders and standards organizations – markets tend to get the balance between variety and compatibility about right (Leibowitz and Margolis, 1990; Tsai and Wright, 2015).

Large firms can find themselves on both sides of the debate between those who favor and oppose intervention in support of interoperability. For example, Google has recently been criticized (and sued) for forking Java to create the Android operating system, and at the same time, is under investigation for including anti-forking provisions in its Android licensing agreements.[1]

---

[1] For a contemporaneous account of the Java-Android forking dispute see, for example, `http://www.`

This paper explores the causes of "failure to tip" in markets with network effects. Our main contribution is conceptual: we propose a classification scheme that distinguishes between forking, fragmentation and splintering as the root cause of mis-coordination. The first part of the paper uses a very simple and stylized economic model to define these three different types of coordination failure, to explain why each one may persist (or not), and to describe the conditions that make them more likely. The second part of the paper uses several short case studies to illustrate the causes of forking, fragmentation and splintering, as well as the tools that firms and industries use to combat mis-coordination. The cases examine standardization and platform competition in railroad gauges, modems, mobile operating systems, instant messaging and Internet browsers. The paper's final section suggests some managerial implications, and considers the difficult question of how to assess welfare effects of efforts to promote interoperability.

2 Forking, Fragmentation and Splintering

Scholars and business people use the terms forking, fragmentation and splintering to describe incompatibility in the presence of network effects. However, it is not clear whether these terms are merely synonyms, or refer to subtly different phenomena. In this section, we propose a classification scheme that distinguishes amongst different types of incompatibility. Table 1 provides an overview.

Table 1: Types of Incompatibility

	Root Conflict	Game	Equilibrium	Example
Forking (stable)	Compatibility	Deadlock	Dominant	Unix
Forking (contested)	Compatibility	Pesky Little Brother	Mixed	Java
Fragmentation	Technology	Battle of Sexes ($n = 2$)	Mixed	56k
Splintering	Technology	Battle of Sexes ($n > 2$)	Pure	SAE

`cnet.com/news/googles-android-parts-ways-with-java-industry-group/`. The European Commission's preliminary views on anti-fragmentation provisions in Android licenses are at `http://europa.eu/rapid/press-release_IP-16-1492_en.htm`

2.1 A Classification Scheme for Coordination Failure

To highlight the key distinctions between forking, fragmentation and splintering, we employ a stylized game-theoretic framework. This framework is meant to illustrate the incentives and strategies of technology suppliers. The cases and discussion below consider more realistic environments, along with implications for customers and complementers.

To keep things as simple as possible, consider a game with two-players $i \in \{1, 2\}$ who must choose between two technologies $j \in \{1, 2\}$. Player i's choice is denoted by a_i. Each player receives a private benefit $b > 0$ if they choose their preferred technology ($a_i = i$) and a coordination benefit c_i if they both choose the same technology ($a_1 = a_2$). Thus, player 1's payoffs are $b + c_1$ if both choose $j = 1$; b if each chooses their own technology; c_1 if they coordinate on $j = 2$; and zero if they each choose the other's preferred technology. We associate forking, fragmentation and splintering with specific Nash equilibria in the simultaneous move complete information version of this game, for different configurations of (b, c_i).

2.1.1 Forking

Forking refers to the creation of a new version of a standard or application that fails to maintain backwards compatibility. The term is widely used in software development, where the practice is common. The Unix operating system has been forked many times. In the 1990s Microsoft tried to fork Java, and more recently Google has been accused of forking Java to create Android. Amazon forked Android to create the Kindle Fire. In our game-theoretic framework, forking arises when players have divergent preferences over compatibility, which we model as $c_1 > 0 > c_2$, so player 1 wants to coordinate while 2 prefers incompatibility.

Forks come in two flavors. Firms or developers may "agree to disagree" and independently pursue separate paths. We call this a stable fork. Alternatively, proponents of the original standard may question the legitimacy of a fork and seek to preserve compatibility. We refer to this second scenario as a contested fork.

A stable fork occurs when $b > |c_i|$. This produces a game with a single equilibrium where each player's dominant strategy is to select its preferred technology.[2] Game theorists sometimes refer to this game as "deadlock." Although the strategies are not very interesting, it provides

---

[2] A dominant strategy is a choice that yield a higher payoff regardless of other players' actions.

a useful reminder that compatibility need not be an efficient outcome simply because one of the players prefers it (conditional on selecting its preferred technology).

Contested forking occurs when $b < |c_i|$. In that case, the players' preference for (in)compatibility exceeds their desire to select a particular technology, creating a game of "pesky little brother" whose only Nash equilibrium is in mixed strategies.[3] When $c_1 = -c_2$, both players choose technology 1 with probability $\frac{b+c}{2c}$ in the mixed-strategy equilibrium. Thus, the probability of coordination is higher when technological preferences are large relative to the coordination payoff ($b \approx c$), but as compatibility issues become more salient, the strategies converge towards a coin toss, and coordination will occur only half of the time.

What do these simple models reveal about actual forking? First, forking can be efficient. In the case of a stable fork, the benefits of variety outweigh the costs of forgone compatibility. Second, when there are strong disagreements over compatibility, contested forking may generate "cat and mouse" games that resemble the unstable dynamics of a mixed-strategy equilibrium, where one actor (or group of actors) seeks to differentiate its offerings while another works to restore compatibility. This can result in a state of partial or intermittent inter-operability, as described below for the case of early HTML standardization. Third, any resolution to these mixed strategy cat-and-mouse games requires a change in payoffs, so the players will either agree to remain compatible or not. In some cases, payoffs change because a court decides to enforce compatibility. For example, the Carterphone decision forced AT&T to open its network to independent device makers. In other cases, such as the Unix wars described below, market developments overtake the compatibility disputes, and the players simply move on.

2.1.2 Fragmentation

Fragmentation occurs when all parties would like to adopt a common standard, but can't agree what it should be. In practice, fragmentation often occurs at the point of upgrade to a standard or platform, when different parties bring their own technology to the table and push for its adoption. For example, when Digital Video Disc (DVD) standards were revised, suppliers fragmented into two camps supporting the incompatible Blu-ray and HD-DVD formats. Greenstein and Rysman (2007) describe a similar episode in the transition from

---

[3] The equilibrium will be familiar to anyone who has played "Odds and Evens" where two players each hold out either one or two fingers: one player wins if they make the same choice, and the other wins if the choices are different.

33K to 56K modem standards. Postrel (1990) studies fragmentation in the development of quadrophonic sound, with CBS, JVC and RCA each sponsoring a different standard, leading to weak availability of complements (i.e. recorded music) and slow end-user adoption.

In our framework, incentives to fragment occur when $c_1 = c_2 > b$. This leads to a "battle of the sexes" coordination game with three Nash equilibria.[4] We set aside (for now) the two pure-strategy equilibria where both players choose the same technology, and focus on the mixed-strategy outcome where each player chooses its own technology with probability $\frac{b+c}{2c}$.

What does this simple model reveal about actual fragmentation? First, note that as b approaches c, both player's are increasingly likely to choose their own technology, and the probability of coordination falls to zero. This suggest that technological preferences play an important role in fragmentation. Farrell and Simcoe (2012b) discuss some of the reasons why firms may have "vested interests" in a particular technology, including sunk research and development costs, intellectual property rights, development lead-times, and the existence of proprietary complements.

Second, note that the two pure-strategy equilibria to this game Pareto dominate fragmentation. In particular, the "loser" in pure strategies receives c, which is larger than the expected payoff $\frac{b+c}{2}$ in a fragmented equilibrium. Given the choice, both players would prefer either one of the pure-strategy outcomes. Thus, when b is small relative to c sophisticated firms can often avoid a fragmented equilibrium. In the limit as b approaches zero, cheap talk (Farrell and Rabin, 1996) should suffice to ensure coordination on one of the two technologies. However, when b is large and rooted in sunk investments, it can be hard to resolve technological conflicts via negotiation. Thus, even when choices are few and players are sophisticated, "accidental" fragmentation can emerge from brinkmanship, occasionally leading to an all-out standards war.

2.1.3 Splintering

Splintering occurs when decentralized technology adoption leads to excessive product variety. For example, Thompson (1954) describes how early automotive component manufacturers each assembled parts to their own specifications. As a result, tire manufacturers had to accommodate a wide variety of wheels, wheel manufacturers had to adapt to a host of axle sizes, axle

---

[4] See Farrell and Saloner (1988) for an extended analysis and discussion of this type of coordination game.

manufacturers had to fit a variety of springs and so forth. Gabel (1994) describes how the regional networks formed to connect independent local telephone exchanges splintered, leading to incompatibilities that persisted from 1894 through 1910.[5] The literature on industry life-cycles contains many similar examples, typically associated with the era of technological ferment that often precedes emergence of a dominant new-product design (Suarez and Utterback, 1995; Klepper and Graddy, 1990).

To model splintering, we retain the battle-of-the-sexes payoff structure ($c_1 = c_2 > b$), while adding a third player and assuming that coordination benefits are only realized if *all* firms adopt the same technology.[6] As in the two-player game, it is a Nash equilibrium for all players to adopt a single technology. However, there is now an uncoordinated *pure-strategy* equilibrium where each player selects its own technology. In this splintered equilibrium, any unilateral deviation yields a zero payoff, while sticking to one's own technology yields b.

The key insight provided by our simple model of splintering is that it takes *coordinated* action to escape from a splintered equilibrium.[7] Coordination sometimes occurs through multilateral institutions. For example, the costs of managing a wide variety of incompatible autocomponents ultimately led to the creation of the Society of Automotive Engineers (SAE), whose early technical standardization work focused on reducing component variety. In other cases, coordination can occur through the actions of a dominant platform leader, such as AT&T's acquiring or displacing regional competitors to establish universal long-distance.

2.2 Coordination Costs and Benefits

In practice, the lines between forking, fragmentation and splintering can be blurry. However, all three are examples of what Farrell (2007) calls *horizontal* incompatibility: complements for one system cannot be used with a rival standard or platform. For example, horizontal incompatibility in the market for video game consoles implies that software developed for the

---

[5] This episode took place in the shadow of AT&T's refusal to inter-connect, which would classify as forking in our ontology.

[6] This strong complementarity assumption can be relaxed. For instance, it is easy to verify that splintering is a Nash equilibrium in the N player battle of the sexes where player i's payoff equals $b + cn$ (where $n \leq N$ is the number of other players who choose the same technology as i), so long as $b < c$.

[7] In game-theoretic terms, splintering is a non-coalition proof pure-strategy Nash equilibrium. The coalition-proof Nash concept involves players in a non-cooperative environment that are able to openly discuss strategies, but are unable to make binding agreements. Hence, any meaningful agreement must be self-enforcing. See Bernheim et al. (1987) for a formal definition.

Xbox will not run on the PlayStation. This is distinct from the "vertical" question of whether game developers can access the installed base for a particular console without first gaining permission from the platform sponsor.

Questions of vertical and horizontal compatibility are often related. Indeed, the famous *Unites States vs. Microsoft* (2001) antitrust case focused on whether Microsoft could legally degrade Netscape's vertical access to the Windows platform, given that Internet browsers might lead to increased horizontal compatibility in other application markets.[8] As this example suggests, the key vertical question is often whether it is necessary to regulate the access polices of a platform leader. The key horizontal question, on the other hand, is whether decentralized technology adoption produces the right balance between variety and compatibility, and if not, what can be done about it?

To properly answer the horizontal question, one must move beyond the simple games described above, which focus only on the incentives of platform sponsors, and consider the costs and benefits of interoperability to consumers and complementers. Broadly speaking, users and complementers will favor compatibility when network effects are strong, when users have similar tastes (so there is limited scope for product differentiation), and when switching costs are large, making it is difficult to achieve compatibility *ex post* through converters, multi-homing or a coordinated platform switch. These factors increase c relative to b (at least to the extent that technology sponsors internalize the preferences of platform users) and raise the cost of adopting multiple standards. Conversely, as c/b shrinks or it becomes easy to adopt both technologies, users favor variety, as in the case of a stable fork.

Variety is valuable when users have diverse needs or tastes. Thus, we often find less compatibility in settings where there are opportunities for specialization. For example, computer programming languages are often well suited to specific tasks, such as text processing (Python, Perl), speedy computation (C++) or database manipulation (SQL), even though code-sharing and portability concerns give rise to network effects. A similar situation exists with audio, video and image file formats, where various standards are tailored to applications that prioritize compression, resolution, ease of transmission or security and rights management, even though compatibility makes it easier to share files. In this context, it is worth noting that although incompatibility tends to imply increased variety, the converse is not always true. For systems goods, such as personal computers and audio-visual electronics, component-level in-

---

[8] This example is due to Farrell (2007), who offers a longer discussion. See also Bresnahan (2002).

teroperability can increase system-level variety by allowing users to mix-and-match parts from diverse vendors (Matutes and Regibeau, 1988).

The costs of incompatibility are also small when it easy for users to adopt several standards or join multiple platforms. For example, many computer applications allow a user to read and write to a wide variety of different file formats. This is made simpler through the use of converter technologies, such as the "container" standards for audio and video files that help web browsers work with a wide-variety of different formats. Similarly, computer programmers often use integrated development environments that facilitate collaboration even among coders writing in different languages (O'Mahony and Karp, 2017). In some cases, individual users can multi-home by adopting multiple platforms. For example, many people have both Visa and Mastercard, and merchant terminals typically accept a card from either platform. Software developers often port their applications to numerous platforms (Corts and Lederman, 2009). And smartphone users can install competing apps, such as Lyft and Uber, making the costs of incompatibility among such services relatively small.

Consumers and complementers might also trade compatibility for increased competition and innovation. Unfortunately, it is not clear *a priori* whether forking, fragmentation and splintering will deliver those benefits. For example, many papers highlight the idea that when network effects are strong, incompatible competition "for the market" can be intense (Fudenberg and Tirole, 1987; Evans and Schmalensee, 2002). But *ex ante* competition need not take the form of transferring surplus to platform users, and its intensity is directly proportional to the anticipated success of "bargain then rip-off" pricing strategies that exploit *ex post* lock-in. Compatible competition (by definition) reduces differentiation, which may increase *ex post* competition without the need for a costly standards war.

Similar trade-offs appear in the relationship between compatibility and innovation. Because standards are public goods, there might be under-investment in shared innovation (Weiss and Sirbu, 1990; Cabral and Salant, 2014). However, even with shared standards, firms have tools to appropriate the value produced by proprietary innovations. When the benefits of having a preferred technology become the industry-wide solution are large, incentives to innovate might even be excessive (Simcoe, 2012).

Given the complexity of these trade-offs, it is natural to ask whether the market, left to to itself, is likely to supply an appropriate level of compatibility? Within our simple theoretical framework, fragmentation and splintering are Pareto-dominated outcomes, leading many game-theorists to favor alternative equilibria where players adopt a common standard. More generally, platform sponsors and user groups have an incentive to coordinate when incompatibility would leave some surplus on the table. Thus, in order to understand the persistence of forking, fragmentation and splintering, we must consider the *coordination costs* associated with various paths to compatibility.

In our analysis, coordination costs are analogous to Coasian transaction costs. Just as Coase (1960) suggested that markets would completely displace firms as the costs of transacting approach zero, we expect rapid coordination on efficient equilibria when the costs of credible communication and collective action disappear. In reality, however, firms must choose amongst a limited set of costly paths to compatibility. Thus, forking, fragmentation and splintering may persist for some time as interested parties decide how to coordinate, and whether the effort is worth the candle. Farrell and Simcoe (2012b) describe four basic paths to compatibility: standard-setting organizations (SSOs), platform leaders, decentralized adoption and converters. Each has its own mix of direct and opportunity costs.[9]

Standard Setting Organizations (SSOs) such as the SAE and the IEEE provide a forum for reaching consensus on the standard itself, and also work to promote widespread *ex post* adoption and compliance. At the technology selection stage, one might think of SSOs as fora where firms seek to maximize c and b by identifying the most promising technologies and engaging in collaborative R&D to improve them. This approach can obviously involve substantial direct costs, as it typically involves many rounds of meetings among interested parties, as well as promotional costs after selection occurs. The essentially political process of selecting a technology can also produce large opportunity costs if consensus decision-making gets bogged down in rent-seeking or legitimate technical disputes (Farrell and Simcoe, 2012a). Ultimately, shared platform governance through SSOs is well-suited to relatively stable platforms, where end-users benefit from compatible competition amongst implementers of core infrastructure, and the large addressable market promotes innovation in complementary applications.

Platform leaders offer another important type of coordination mechanism. In the historical auto industry, one might view Ford and GM as platform leaders who coordinated all of their own component-level design decisions. Today, we are more likely to think of firms such as Google or Microsoft, who provide tools and interfaces that third-party software developers can

---

[9] We focus on the different institutions used to achieve a coordinated outcome. For a review of some of the tactics they employ, see Besen and Farrell (1994).

use to create new products and services. At the most general level, a platform leader could be a large customer or even the government.

The defining feature of a platform leader is that they have clout. Once a platform leader chooses a direction, it is generally in the interest of others to follow. In our simple model of fragmentation and splintering (but not forking), any first-mover that commits to their preferred standard could play the role of a platform leader, since the second mover favors coordination. In practice, platform leaders have a wide variety of tools for altering the incentives and expectations of users and complementers (Gawer and Henderson, 2007; Boudreau and Hagiu, 2009; Weyl and White, 2014). Thus, even when there are disagreements about compatibility, platform leaders may be able to engineer coordination by manipulating other player's payoffs.[10]

Relative to SSOs, platform leaders are often better at coordinating "architectural upgrades" to a core technology requiring the participation of numerous platform users with diverse interests. Moreover, because they internalize the opportunity costs of delay, platform leaders are typically more decisive than SSOs and less likely to get bogged down in distributional conflicts over the choice of technology. Thus, we often find platform leadership in markets that go through waves of upgrades, like video game consoles.

Platform leaders incur many of the same direct costs of developing and promoting a new technology as SSOs. However, an indirect cost of platform leadership is that they exert market power, restricting platform access (to some degree) in order to increase profits. And, relative to SSOs, platform leaders may be less innovative, particularly in settings where decentralized innovation yields a better menu of technologies (Boudreau, 2010), or when the platform leader is a customer or government agency lacking in technical expertise.

Converters provide a third imperfect path to compatibility. When they work well, converters allow users to make independent *ex ante* choices, but restore *ex post* compatibility in the event of forking, fragmentation or splintering. However, converters often underperform a dedicated standard (Baldwin and Clark, 2000). And like multi-homing, converters impose costs on end users, who must keep track of various "plugs and dongles" in an effort to engineer a degree of inter-operability.

Finally, decentralized adoption (or standard wars) can provide a route to compatibility. Here, the costs include the possibility of converging on an inferior solution (David, 1985),

---

[10] For example, see the discussion of tactics used by Microsoft in the browser wars below.

the possibility of delays as end-users delay their choice until the collective decision becomes clear (Farrell and Saloner, 1986), and duplicative or wasteful efforts by competing technology sponsors to tip the market in a preferred direction.

Our point in emphasizing the drawbacks of each path to compatibility is not to say that they don't work. Rather, the relative magnitude of various types of direct and opportunity costs will influence how long we should expect forking, fragmentation or splintering to persist, and the selection of a specific coordination mechanism under varying technological and economic conditions.

3 Case Studies

In order to illustrate the ideas developed above, this section presents several case studies of industries, technologies or standards that forked, fragmented or splintered.

3.1 Railroads

In the summer of 1886, more than 13,000 miles of railroad track in the Southern U.S. were converted from a gauge of 5 feet to a width of 4-feet 9-inches, making them compatible with the bulk of the Northern rail network. This episode is described in both Shapiro and Varian (1999) and Gross (2016).[11] A historical examination of railroad standards illustrates how compatibility matters outside of the information technology sector, and how splintering can occur even when the costs of incompatibility are large.

The first efforts to build commercial rail service in the United States occurred in the 1820s and 1830s. Most lines offered only local point-to-point service, and there was substantial technological experimentation, which included trying out various gauge-width specifications. From the 1830s through the 1860s there was major investment in building out the U.S. railroad network, and rail came to replace waterways as a dominant mode of transport. During this period, advances such as telegraphy allowed for increased network utilization, and greater integration. However, in the absence of any mechanism for creating or coordinating a national

---

[11] This account draws heavily on the historical section of Gross (2016) and the references cited therein.

network, the initial heterogeneity in gauge standards persisted. Siddall (1969) reports that there were at least 23 different gauge standards in use during the 1860s.

Network effects did influence the choice of early railroad builders, as new lines often chose a gauge that allowed for interoperability with existing adjacent lines. However, instead of producing a single national network these early decentralized choices led to the formation of "gauge regions" that allowed for seamless internal transport, with incompatibilities and switching costs concentrated at their geographic borders (Puffert, 2009). Although the companies operating in different gauge regions presumably had a preference for their own standard, railroad gauge standardization does not seem to be a case where fragmentation emerged from intense competition between a few sponsored alternatives. Rather, the lack of coordination emerged from a combination of initial experimentation, path-dependence and decentralized decision-making.

As regional networks grew and merged, the costs of incompatibility became clear. The largest costs were associated with trans-shipment: the process of moving goods from one gauge to another at the point where incompatible networks met. The direct costs of trans-shipment included hiring labor to perform the task, and maintaining specialized capital to facilitate the switch. There were also substantial opportunity costs, including delayed arrival (the process often took a day or more) and the cost of maintaining extra rolling stock and other capital.

Railroads tried using a variety of technologies to reduce the costs of incompatibility. For example, bogie exchange is the process of changing the wheels under a carriage in order to operate on otherwise incompatible track. Railroads also experimented with adjustable width rolling stock, and multi-gauge track (i.e. a third rail). However, none of these converter technologies were completely effective at removing delays or matching the overall performance of a uniform gauge.[12]

Over time, the costs of incompatibility scaled with utilization of the rail network, creating strong incentives for further convergence during the Civil War and reconstruction. By the 1880's, through both conversion and new construction, the U.S. rail network gradually converged to a system with two incompatible gauge standards, 5 feet and 4 feet 8.5 inches, with the former gauge highly concentrated in the South.

The final step in the process of achieving nationwide interoperability was a remarkable

---

[12] Levinson (2006) provides a related account of the costs of break-bulk shipping in ocean transport prior to the arrival of containerization.

conversion of roughly 13,000 miles of track during an extremely short period in May and June 1886. Just before this conversion, the majority of Southern freight carriers – including both rail and steamship – had organized themselves into a cartel called the Southern Rail and Steamship Association (SRSA). Although the main purpose of the cartel was rate-setting, they quickly realized the large potential efficiencies of converting to a gauge standard that would allow seamless interconnection with the Northern network. The conversion of the SRSA network to standard gauge was a carefully orchestrated engineering feat, described in detail by Hudson (1890) and more recently Puffert (2009).

From an economic perspective, the SRSA played two very important roles. First, it helped coordinate the switch, which was clearly more beneficial for members who were operating at the geographic boundary of the network than for those deep in the South, who would not regularly incur the costs of incompatibility. Evidence suggests that networks in the deep South were more reluctant to switch, and could only be brought along because the SRSA convinced them that all of their adjacent neighbors would be changing gauge. The SRSA's second role was to ensure (through coordinated pricing) that interoperability benefits flowed to its members, and were not dissipated through *ex post* competition.

Gross (2016) uses data from SRSA and other freight schedules to study the economic impacts of the 1886 switchover. He finds that there was a substantial reallocation of traffic from steamship to rail for routes that would have formerly required trans-shipment, or interchange via bogie exchange. The effect is concentrated on shorter routes, where the costs of delay were proportionally larger. However, he finds that there was little change in price or aggregate volume, presumably because of the price discipline imposed by the cartel. Using a model of supply and demand, he also computes counterfactual impacts of standardization in a competitive market, which suggest that under competitive conditions, the gauge change would have lead to a 10 percent average price decline, and a resulting 9 percent increase in traffic for the routes in his sample.

The railroad case study offers several important lessons about splintering. First, it demonstrates how a combination of decentralized adoption and technological uncertainty can lead to splintering. It also shows how splintering can persist, even in the presence of substantial opportunity costs, when the sunk costs of replacing installed capital are large. The case also illustrates two paths to compatibility. One is the use of converter technologies, like bogies and adjustable wheels, to reduce costs of incompatibility. The second is for a large "platform leader" such as the SRSA to step in and coordinate a switch.

The empirical work by Gross (2016) provides some quantitative evidence of the welfare gains from inter-operability in this setting. However, it also raises the interesting question of whether the large counter-factual benefits of interoperability *plus* competition could have been achieved in the absence of the SRSA, since that organization played an import role in coordinating the switch and ensuring that Southern railroads would benefit from it. Finally, it is worth noting that rail gauge standardization is not merely an intriguing historical episode: there are more than five gauge standards currently used in Asia, and some incompatible national networks have been negotiating towards technical interoperability for over 50 years (UNESCAP, 1996; UNTC, 2006).

3.2 Modems

During the 1990s, many U.S. consumers accessed the Internet by using a modem to connect with an Internet Service Provider (ISP) over the public telephone network. The invention of the browser and growth of the World Wide Web generated significant demand for faster connections, and by early 1997, modem suppliers and ISPs were both poised for an upgrade to equipment with a maximum transmission rate of 56 kilobits per second (56K). In a pair of complementary papers, Augereau et al. (2006) and Greenstein and Rysman (2007) describe the transition to 56K, which illustrates the incentives that lead to fragmentation as well as the use of an SSO to break the resulting deadlock.[13]

Prior to 1997, modems operated at a maximum speed of 33K. The market for 33K modem chipsets was dominated by Rockwell Semiconductor, who licensed its technology to various resellers that had a combined market share exceeding 80 percent. The largest of these resellers was US Robotics. The arrival of the World Wide Web and the emergence of a highly fragmented dial-up access market served by many local ISPs (Greenstein, 2000) helped foster demand for the 56K technology, and US Robotics began to work on its own solution, based on a standard called X2. Concerned that it might miss the transition, Rockwell entered into a consortium with Motorola and Lucent to develop their own standard called K56Flex (henceforth Flex).

The two incompatible standards – X2 and Flex – reached the market around the same time in early 1997. While there were some early reports of problems with Flex modems, the two technologies had similar quality and pricing within six months of introduction. However,

---

[13] Our account is largely based on the discussion in Greenstein and Rysman (2007).

because the standards were incompatible with one another, ISPs needed to purchase separate equipment in order to support Flex and/or X2. A mismatch between consumer and ISP hardware would limit speeds to 33K at best. This created indirect network effects in the diffusion process: consumer adoption of one standard increased ISPs' incentives to select similar technology, and vice versa.

Contemporaneous reports suggest that adoption of X2 and Flex modems was slow relative to expectations and the size of the market. By October 1997, just over 50 percent of ISPs had made the upgrade, but neither standard had emerged as the market leader. None of the major ISPs adopted 56K during this time.[14] The wait-and-see posture of both consumers and large ISPs suggests that fragmentation was leading to excess momentum for the 33K technology. Moreover, while ISPs did have an incentive to upgrade, it is not clear that they had strong incentives to coordinate on a single standard. In fact, Augereau et al. (2006) provide evidence that small ISPs used incompatibility as a source of differentiation. Specifically, their study shows that when competing ISPs adopted 56K, they tended to divide local markets, with roughly half of ISPs serving X2 and the other half Flex.

During the development and rollout of X2 and Flex, efforts were underway at both the Telecommunications Industry Association (TIA) and the International Telecommunications Union (ITU) to reach consensus on a single 56K standard.[15] These are several likely reasons why these efforts failed to yield a consensus before fragmentation occurred. First, participants in the formal standards process are typically interested parties, which in this case would include members of both the US Robotics and Rockwell-led consortia. Moreover, because SSOs lack formal enforcement power, it is not unusual for them to wait and see whether there are signs that a *de facto* standard will emerge in the market prior to endorsing any particular solution.[16] Nevertheless, Greenstein and Rysman (2007) report that both the X2 and Flex consortia expected to adopt an ITU standard. And the slow adoption of 56K technology by consumers and large ISPs placed some pressure on the SSOs to act quickly, in order to break the logjam that was holding back demand.

---

[14] The list of large non-adopters included AOL, AT&T, UUNET, MSN, GTE, BellSouth and EarthLink.

[15] The TIA is a U.S. industry association that develops standards under the auspices of ANSI, and can therefore serve as the U.S. representative to ITU, which is a Geneva-based UN treaty organization. ITU has set a variety of international telecommunications standards since the late 1800s.

[16] For example, the Internet Engineering Task Force requires has several tiers of formal endorsement, and will only advance a specification from "Proposed Standard" to "Draft Standard" if there have been multiple independent implementations.

Thus, in February 1998, the ITU announced that there was consensus for a new 56K modem standard called V.90. (This represented a new "record" for elapsed time to develop an ITU standard, and was well ahead of the SSO's two-year forecast.) Although V.90 was an amalgam of X2 and Flex technology, the standard was not "plug and play" interoperable with either of the proprietary specifications. Customers could, however, use a firmware upgrade to make their existing X2 or Flex modem work with an ISP's V.90 equipment. In September 1998, the V.90 standard was approved, and sales were strong following the adoption of a coordinated standard.

The 56K modem case nicely illustrates how fragmentation can occur when it becomes time for a technical upgrade, and how SSOs can be pivotal in resolving an impasse in standards adoption. One of the more interesting features of this case is the role of ISPs. Large ISPs sat on the sidelines, rather than make a risky bet on a single standard that might lead to stranded investments, as in the model developed by Kretschmer (2008). Smaller ISPs viewed incompatibility as a potential source of differentiation in a highly competitive industry, and consequently exacerbated the fragmentation problem. Thus, even in the presence of indirect network effects, the early ISP adopters were not especially keen to coordinate.

The 56K modem case also highlights the interaction between market and non-market paths to compatibility. In their review of this episode, Greenstein and Rysman (2007) ask why US Robotics, who seemed to be ahead in the marketplace, was keen to adopt V.90. They propose that US Robotics never believed that the market would tip towards X2, and only expected to obtain some temporary advantages by establishing an early lead in adoption. In particular, one of the major benefits of X2's edge in the market was that Rockwell and others agreed to include a substantial amount of US Robotics' intellectual property in the V.90 standard. This meant that US Robotics would no longer be in the position of licensing and distributing Rockwell's technology, as they had been for 33K modems. With these IP concessions in place, the benefits of accelerated adoption presumably outweighed the costs of moving from X2 to V.90, and US Robotics quickly endorsed the ITU specification.

3.3 Unix

Unix is one of the most technically and commercially significant operating systems in the history of computing. The original Unix operating system was developed at Bell Laboratories in the early 1970s, and there have been hundreds of different implementations and offshoots since then. This short case study will focus on the "Unix Wars" that took place in the 1980s and 1990s.

When engineers at AT&T first developed Unix, the company was prohibited from entering the computing industry under the terms of a 1956 antitrust consent decree. Bell Labs therefore decided to license the source code "as-is" for a nominal fee, but without a guarantee of support or bug fixes. The inexpensive OS quickly diffused among minicomputer users, who were often located at large institutions such as universities that had the resources required to buy and operate these machines. Many early Unix users contributed to the ongoing development of the operating system. For example, then graduate student Bill Joy released the first Berkeley Software Distribution (BSD) as an add-on to Version 6 Unix in 1977. This fork would go on to become one of the major branches in the upcoming Unix wars.

Several key events leading to the first round of Unix wars occurred around 1982. The break-up of the Bell System produced a new consent decree that freed AT&T to enter the computer business. One year later, AT&T released Unix System V, one of the first commercially available versions of the OS. Meanwhile, Sun Microsystems was founded (by Bill Joy, among others), and enjoyed early success at commercializing Unix through bundling SunOS, which was derived from BSD, with hardware aimed at the nascent workstation industry. A key technical advantage of the early BSD implementations was that they had built-in support for TCP/IP networking. However, until 1988, implementations of BSD still required a license from AT&T because it was derived from their original source code.

As sales of workstations accelerated, Sun's business model of bundling hardware with a proprietary flavor of Unix – typically a derivative of either BSD or System V – was quickly adopted by many of the incumbent minicomputer manufacturers. The result was somewhere between a stable fork (if one focuses on the two main flavors) and complete splintering. Salus (2015) describes the market for Unix implementations in the early 1980s:

> "Apollo, DEC, Eakins, Gould, Integrated Solutions, Masscomp, NSC, and Wollongong were marketing Berkeley UNIX. System III or System V derivatives were being marketed by AT&T, Altos, Apollo, Compaq, Convergent, HP, Honeywell, IBM, ITT, Intel, Interactive, Masscomp, Microport, Microsoft, Motorola, NCR, NUXI, Opus, SCO, Silicon Graphics, Sperry, Sun, Tandy, UniSoft, and Wollongong. Finally, a host of vendors, including Amdahl, Apple, Cray, DEC, Data General, HP, IBM, and Motorola, offered proprietary versions of UNIX, some based on 4.1 or

4.2BSD."

With splintering leading to interoperability and portability concerns, AT&T began requiring vendors to conform to a variety of standards in order to use the System V brand. Another significant effort to promote Unix standardization was started within the IEEE, under the POSIX (Portable Operating System Interface) trademark. Although these efforts at platform leadership did increase interoperability, the first round of the Unix Wars essentially ended in a stalemate between the BSD camp and the System V camp.[17]

The second round of the Unix Wars began in 1987 when AT&T announced a large investment in Sun Microsystems. Sun simultaneously announced that its future Unix OS development (Sun Solaris) would be based on AT&T's System V Release 4, as opposed to BSD. Although this collaboration was hailed by customers and the press as helping to resolve the prior incompatibility issues, many of Sun's competitors – who were also often AT&T licensees – feared that they would be placed at a significant competitive disadvantage. In 1988 these competing vendors formed the Open Software Foundation (OSF), a consortium whose key members included Digital Equipment, Hewlett Packard, and IBM.[18]

OSF members jointly developed the OSF/1 operating system, which did not incorporate any of AT&T's intellectual property. In response, AT&T, Sun and a group of SVR4 licensees formed Unix International (UI) as a counter-consortium. Despite the significant resources spent on its development, OSF Unix was not a commercial success. Digital Equipment was the only company to produce a complete implementation, and Cargill (2011) summarizes this round of the Unix battles by writing that, "OSF/1 was an idea whose time had come and gone, and the proprietary offering (UNIX SVR4) won."

By the early 1990s, the market for workstations appeared mature compared to the fast growing desktop market, which was increasingly dominated by Microsoft. GNU/Linux had also emerged as a fully open source alternative to the various proprietary flavors of SVR4 then on the market. With these commercial developments as a backdrop, the members of both UI and OSF formed the Common Open Software Environment (COSE) initiative in March 1993, with UI and OSF merging into what eventually became The Open Group. The second round of the UNIX wars came to a close when AT&T sold its Unix rights to Novell. The Open Group

---

[17] Salus (2015) relates how the two camps had marketing campaigns at the 1988 USENIX (user group) conference with the competing tag-lines "System V: Consider it Standard" and "4.2 > V."

[18] According to Axelrod et al. (1995), Sun's CEO Scott McNealy joked that OSF actually stood for "Oppose Sun Forever."

continues to hold the trademarks to Unix, and offers testing and certification programs based on the Single Unix Specification (SUS), whose core specification development takes place under the auspices of the IEEE POSIX program.

The Unix wars contain several lessons about the economics of forking and splintering. First, the early work on BSD shows how forking need not always be harmful. In particular, the experimentation of Bill Joy and others in the academic community arguably fostered the development and improvement of an operating system that AT&T had all but abandoned. At the same time, those forks created an environment in which camps could easily form around the competing BSD and System V specifications. Secondly, this case illustrates the potentially complex interplay among various paths to compatibility, including decentralized adoption of proprietary standards, "sponsored" consortia such as UI and OSF, and more neutral SSOs such as the IEEE.

The history of Unix also shows how hardware vendors can play a similar role to the ISPs in the 56K modem standards war. In particular, even though there were arguably positive network effects among end-users and software developers who all favored a greater level of inter-operability, the key *adopters* were minicomputer and workstation producers who often preferred a proprietary flavor of Unix that could provide a greater level of product differentiation. One of the most interesting questions posed by this case is whether Unix fragmentation in the minicomputer and workstation market contributed to the rise and eventual dominance of Windows in the market for personal computer operating systems.

Another lesson from the Unix wars is the importance of intellectual property. AT&T's licensing activities played a role in both the early BSD vs. System V fights, and the later formation of OSF. OSF's commercial failure illustrates how divided governance of a standard may fail in the face of strong competition from a proprietary alternative, a pattern we will see repeated below in the case of Symbian. Finally, the creation of The Open Group illustrates how slower market growth, along with the introduction of outside threats (in this case from Linux and Windows), can help resolve a stalemate over standards that once appeared to be a stable fork.

3.4 Instant Messaging

Although messaging applications date back to the era of mainframe computing, Internet-based instant messaging was introduced in the mid-1990s. This case study of instant messaging draws heavily on the account provided in Faulhaber (2002, 2004).

The first messaging applications such as ICQ, PowWow, and AOL Instant Messenger (AIM) had graphical user interface (GUI) clients and allowed for real-time conversations that distinguished them from email. Another defining characteristic of first-generation messaging protocols was a lack of horizontal openness. Users of one service could not communicate with the users of another, competing instant messaging application. Although multi-homing was possible, users needed to maintain accounts on each separate IM network, and concurrently run multiple client applications in order to communicate across multiple networks.

AOL Instant Messaging (AIM) was the largest of the first-wave of messaging platforms. AIM was introduced in 1989, but surged in popularity around 1996 when AOL added a "buddy list" feature that allowed users to see whether their frequent chat partners were currently online. Although AOL initially limited the messaging network to its own subscribers, in 1997 AIM was offered as a standalone application for non-AOL customers. As AIM's user base grew, a number of competing services made efforts to interconnect. While the technical problems were not large – AOL had already published its OSCAR messaging protocol on the Internet – all of the initial attempts to connect without AOL's permission were blocked.[19] This resulted in periods of intermittent compatibility, consistent with the cat-and-mouse dynamics of a contested fork.

During the late 1990s, a number of AOL's competitors also deployed proprietary messaging protocols. Microsoft Messenger utilized the MS Notification Protocol (MSNP), and Yahoo Messenger relied on a protocol called YSMG. Given the lack of interoperability among these standards, several efforts to create open instant messaging standards were started within the Internet Engineering Task Force (IETF).[20] However, the initial push for standards-based inter-operability in instant messaging largely failed. In particular, although some third-party software allowed for connections to multiple IM networks from a single client application, most

---

[19] Perhaps ironically, OSCAR stands for Open System for Communicating in Realtime.

[20] Examples include the Session Initiation Protocol (SIP), Session Initiation Protocol for Instant Messaging and Presence Leveraging Extentions (SIMPLE), Application Exchange (APEX), Instant Messaging and Presence Protocol (IMPP), and the Extensible Messaging and Presence Protcol (XMPP).

of the network providers proved willing and able to refuse interconnection with their rivals.

This situation started to change in the early 2000's. In 2001, as a condition for approving the merger between AOL and Time-Warner, the U.S. Federal Communications Commission required AOL to commit that it would provide rivals with access to the AIM Names and Presence Directory (NPD) before offering "advanced" IM services, such as voice and video communications (Faulhaber 2002).[21] This regulatory step was followed by a series of deals that facilitated cross-network communications. In 2003, Reuters signed agreements that allowed users of its proprietary Reuters Messenger service to communicate with users of AIM, ICQ and Microsoft Messenger. In 2005, Microsoft's SIP/SIMPLE based enterprise IM product, Live Communications Server 2005, was opened to communicate with users of AIM, MSN Messenger, and Yahoo! Messenger. And in 2007, Google's XMPP-based Google Talk service allowed for communication with AIM users.

By the late 2000s, new technologies and platforms were providing consumers with alternatives to the previous generation of stand-alone desktop-based instant messaging applications. These alternatives included SMS text messaging services operated by wireless carriers, proprietary text-messaging protocols such as Apple's iMessage, and standalone messaging applications such as WhatsApp. Popular social networks, such as Facebook and Twitter, also added instant messaging features to their platforms.

The history of instant messaging clearly parallels that of the early telephone network, as described in Gabel (1994). Both communication technologies produced strong direct network effects, and the service providers who established an early lead refused to interconnect with their smaller competitors, presumably out of a desire to differentiate based on the size of their installed base. However, whereas the regional phone networks eventually merged to monopoly, the instant messaging market evolved differently. The FCC intervened to promote horizontal inter-operability, and facing competition from a variety of substitute communication platforms, the largest messaging platforms ultimately agreed to inter-connect.

While the open protocols developed by the IETF during the late-1990s were not initially embraced by proprietary IM networks, they eventually played an important role in bilateral interconnection. This reinforces a theme that also appeared in the case of OSF Unix – just because an open protocol exists, it will not necessarily succeed in the marketplace. However,

---

[21] As explained in Faulhaber (2002) the NPD is the critical component in terms of "network effects" because it provides real-time information on user availability.

open technology can often provide a foundation for subsequent iterations of the platform.

3.5 Internet Browsers

Tim Berners-Lee developed the first web browser in 1990, while working at the European Organization for Nuclear Research (CERN). However, the first commercially significant browser was Netscape Navigator. From its early release in 1994, Navigator's feature richness, combined with its free use for non-commercial purposes helped Netscape establish an early lead in browser adoption. The company's business model at the time called for giving away the browser, while charging for both its web server software and support for business users.

By mid-1995, Microsoft clearly perceived the Internet as a major opportunity, and Netscape Navigator as a significant threat. Bill Gates' now-famous Internet Tidal Wave memo spells out several elements of Microsoft's catch-up strategy, including "a decent client that exploits Windows95 shortcuts," working to "figure out additional features that will allow us to get ahead with Windows customers" and "get[ting] OEMS to start shipping our browser preinstalled." Gates clearly recognized the importance of complementary content, and the memo also discussed the evolution of document standards, suggesting that, "We need to establish OLE [an MS proprietary document protocol] as the way rich documents are shared on the Internet. I am sure the OpenDoc consortium will try to block this."[22]

Many of the tactics described by Gates played an important role in the subsequent "browser wars." Microsoft released its Internet Explorer (IE) browser in August 1995, as an add-on to Windows 95.[23] In 1996, Microsoft began bundling IE3 with Windows 95 free of charge, marking the start of serious competition in the browser space. Starting with just over 3 percent of usage share in 1996, Internet Explorer captured over 30 percent of the market by 1998, and became the market leader by 1999. Bresnahan and Yin (2007) show how Microsoft's strategy of pushing hardware OEMs to pre-install IE played a crucial role in helping IE catch up to, and eventually surpass Netscape.

During Internet Explorer's rise, the web content standards supported by IE and Netscape diverged, with each team adding proprietary features to attract developers. It was common for websites to be specially targeted at either Netscape or Internet Explorer, displaying logos

---

[22] The full memo is available at `http://www.lettersofnote.com/2011/07/internet-tidal-wave.html`.

[23] Microsoft licensed much of its original browser code from Spyglass Mosaic.

such as "Best Viewed With Internet Explorer" or "Best Viewed With Netscape Navigator." Many web sites also utilized scripts that would detect a visitor's browser version, and then load the appropriate version of their content. These practices increased costs all around – websites were slower for the end-user to download due to increased markup, web-server load was higher, and developers needed to expend greater effort developing duplicate versions of websites for different browser standards.

Throughout this first stage of the browser wars, the World Wide Web Consortium (W3C), an SSO founded by Tim Berners-Lee in 1994, worked to prevent forking of key document standards by publishing specifications for Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) and other web protocols. In 1998, a group of web developers also founded the Web Standards Project (WaSP) to campaign for browser compatibility. WaSP published a series of influential "acid tests" for browser compliance with key standards, such as HTML, CSS, and ECMAscript (an SSO-maintained version of Sun's JavaScript language).[24]

By the early 2000's, Internet Explorer had a 90 percent share of the web browser market, and Microsoft's strategy was increasingly characterized as an effort to "embrace, extend and extinguish" a set of standards that might threaten the dominance of its Windows platform. This three-step strategy begins when a platform leader embraces a standard by providing vertical inter-operability. The "extend" part involves forking the standard by adding proprietary extensions that competitors cannot or will not implement. Finally, when the proprietary extensions become a *de facto* standard – presumably, in this case, because of Microsoft's large installed base – the open specification can be extinguished in the marketplace. Gilbert and Katz (2001) describe some tactics Microsoft used to "pollute" Java, such as refusing to incorporate certain Java components, incorporating proprietary extensions into its Java developer tools, and pressuring developers and hardware OEMs to use Windows-specific Java technology. The judge in *U.S. vs. Microsoft* ruled that these actions were inconsistent with Microsoft's arguments that it was merely optimizing Java for the Windows platform.

With both the Unix and browser wars taking place during the 1990s, there was growing recognition of the potential for forking and fragmentation, and also several innovations in *platform governance* that played a role in the subsequent evolution of web standards. One of these governance innovations is the copy-left provision in Open Source Licensing agreements (Stallman et al., 1985; Lerner and Tirole, 2001). Broadly speaking, a copy-left provision requires

---

[24] ECMA was originally an acronym for the European Computer Manufacturers Association.

anyone that releases a modified version of an open source program to make their own source code freely available to anyone that desires a license.[25] Under a copy-left licensing regime, it is possible to embrace and extend standards, but very difficult to extinguish competition, because rivals will always (eventually) have access to the underlying code and any rights required to implement the proprietary extension.[26]

In 1998, Netscape decided to release all of its browser source code under an open source license with copy-left provisions. The license was called the Mozilla Public License, and is used by the community that develops the eponymous browser. Starting around 2004, Mozilla Firefox 1.0 and several other browsers based on the Netscape source code began to recapture market share from IE (which peaked at around 90 percent in 2002).

Around the same time, engineers from Mozilla, Opera, and Apple formed the Web Hypertext Application Technology Working Group (WHATWG) to speed up the development of web standards. The members of WHATWG expressed frustration with the pace and direction of W3C's standardization efforts, which had stalled after the release of HTML 4.01 in 1998. The WHATWG members initially approached the W3C to discuss creating an HTML extensions working group, but were denied permission because they did not intend to base this work on the XML specification that W3C was pushing at the time. Within a few years, WHATWG had rewritten the base HTML specification and added many commercially significant extensions, such as increased support for audio and video functionality.

In 2007, the W3C responded to WHATWG's growing influence by forming a new working group to develop non-XML based HTML extensions. The W3C committee incorporated a substantial amount of the WHATWG's HTML5 specification into its standards, leading to a conflict between the two groups over the "right to fork." WHATWG was arguably created with the goal of forking the HTML specification, and has a copyright policy that allows others to use its standards as the basis for further specification development. The W3C, on the other

---

[25] An alternative approach to preventing forking and fragmentation is to place the specification in the public domain under relatively restrictive copyright licensing terms that prevent unauthorized "extending" of the standard. This is the approach taken by several SSOs, including the W3C. However, this approach does not prevent proprietary extensions in implementations of a standard. Moreover, ongoing litigation between Oracle and Google raises the question of whether copyright protection extends to the APIs defined by many software standards, or can only cover the implementation code.

[26] Many standards consortia have substantively similar provisions, requiring implementers to freely license essential patents for the specification and future extensions of it. These policies fall under contract law as opposed to copyright. There are also "Commercial Open Source" licenses that allow re-use and modification of source code, but do not contain copy-left provisions.

hand, had a more restrictive policy that did not allow this type of specification forking.[27] Open source advocates objected to the W3C's restrictive policy, especially when the HTML working group seemed to take liberal advantage of the right-to-fork offered by WHATWG. This led to considerable debate within the W3C over more permissive licensing, and experiments within the W3C's HTML working group at changing the SSO's policy.

The browser wars illustrate several economic facets of forking. The initial battles between Microsoft and Netscape/Sun are a nice illustration of a contested fork, where one side seeks to preserve compatibility and the other to degrade it. While Microsoft's advantages in distribution helped them win the initial battle for browser share, it is worth noting that the technology advanced rapidly during this time, and the threat of HTML fragmentation was ultimately averted with help from groups like W3C and WaSP. Netscape's eventual decision to release its browser source code under an open-source license, after losing the initial battle to IE, shows how the governance innovation of copy-left licensing provisions can prevent "hijacking" of a specification by conditioning implementation rights on a promise to place follow-on innovations back into the public domain.

This case also illustrates the now-famous embrace-extend-extinguish idea, where a platform leader may try to fork an open standard because that specification threatens the firm's dominance in an adjacent market. In defense, Microsoft argued that they were simply innovating and optimizing the standard for their own platform. Ultimately, distinguishing efficient from anti-competitive forking necessarily involves a close look at at the tactics of a platform leader, and their likely impact on consumer, complementers and competition.

Finally, the more recent fights between WHATWG and W3C illustrates a key tension at the heart of copy-left open source licensing: it enshrines the "right to fork" (thus preserving the option to innovate on top of a standard) by insisting on vertical openness, so that future implementers retain the ability to access the fork (and therefore the ability to merge useful innovations back into the standard). Perhaps as a result of this tension, many collaborative software development projects have chosen less restrictive "commercial open source" licenses that trade the prophylactic benefits of copy-left for more permissive rules around proprietary implementations and extensions.

---

[27] Several other SSOs, including the IETF, have copyright policies with prohibitions on derivative works. This may be a historical legacy of SSOs' business model of selling copies of the standard, but also reflects a divergence in the norms and objectives of more formal SSOs (who tend to view forking as a threat), and the open source community (who focus more on open access to source code).

3.6 Symbian

Symbian was a widely used mobile phone operating system that shipped on nearly 450 million handsets between 2000 and 2010. The OS was developed by a London based company with the same name, founded in 1998 as a spinoff from Psion, which had previously sold an operating system for palmtop personal digital assistant (PDA) devices. Symbian's key initial investors were Nokia, Ericsson and Motorola.[28]

The original Symbian business model called for licensing the OS to handset manufacturers that would then combine it with a variety of other components to produce a mobile phone. The key additional components include an ARM-based microprocessor and a customized user interface (UI). UI customization was a unique feature of Symbian that distinguished the mobile OS from other platforms and allowed device makers to customize the "look and feel" of their phones. Third-party software developers could develop applications in either Java or C++ programming languages. However, programming for Symbian was difficult because its APIs differed from popular PC based programming environments, and because each user-interface had its own custom APIs, so a program written for one Symbian phone would not necessarily run on another.

The Symbian platform grew exponentially between 2002 and 2007, mainly because it was incorporated into a series of very successful Nokia handsets. In 2006, under pressure from its owner-customers, Symbian cut its prices from \$5 per handset (with a \$2.25 surcharge on the first 2 million units) to a sliding scale starting at \$5 with volume discounts that could lower the price to \$2.50. By 2008, Nokia had a roughly 80 percent share of all Symbian handsets, and was the only firm to benefit from the volume discount on license fees. Given the fixed cost of developing and supporting a customized UI, Nokia also achieved substantial economies of scale relative to Ericsson, Motorola and other Symbian licensees.

Between 2007 and 2010, Symbian's share of smartphone OS shipments declined precipitously. Nokia acquired the platform in 2008, but by 2011 announced that it would switch to Microsoft's Windows Phone as the operating system for future smartphone development.

Several factors contributed to the rapid fall of Symbian. First, although the Symbian achieved considerable international success, its North American sales were always slow. This

---

[28] Our short history of Symbian draws upon the more detailed case studies by West and Wood (2013) and West (2014).

allowed Research in Motion (BlackBerry) to capture a substantial share of the early U.S. market, and left an opening for Apple's iOS and Google's Android to grow rapidly starting around 2007. Second, Nokia's role as the dominant sponsor of the Symbian ecosystem created a situation where rival device makers were happy to switch once alternative platforms became available. Finally, Symbian failed to develop an "App Store" for direct distribution of third-party applications to consumers, which was a key element in the rapid diffusion of iPhone and Android. While Symbian did develop app distribution capabilities, Nokia insisted on a wholesale model, where Symbian could not bypass the device makers and network operators, so that customers ultimately bought from a storefront branded by Nokia or one of the carriers.

As Symbian's share of the smartphone market plummeted in 2008 and 2009, many of these issues became apparent, and in a bid to preserve the platform, Nokia created the Symbian foundation, transferred the source code and intellectual property rights to that organization, and released the entire project under an open source license. However, as key OEMs such as Samsung and Sony-Ericsson continued to leave the Symbian platform for Android, Nokia took development back in-house before finally abandoning the Symbian platform in 2012.

The Symbian case illustrates how fragmentation can occur when the sponsors of a collectively governed platform have divergent interests. In particular, the conflicting goals of the Symbian platform and the device makers who owned it can be discerned in several key decisions. For example, allowing OEMs to develop their own user interfaces and APIs enabled more differentiation among handset vendors, but frustrated independent software developers, who never adopted Symbian at the same rate as iPhone or Android. Thus, while fragmentation provided end-users with a greater variety of Symbian devices, they paid for this variety through higher prices, less inter-operability, and a much smaller and slower supply of complementary software. When Symbian began to succeed in the marketplace, its owner-customers renegotiated the licensing terms for access to the OS. And when it became clear that direct-to-customer app distribution was an important feature of iPhone and Android, Symbian's owners insisted on manufacturer or carrier-branded distribution.

One of the key lessons of the Symbian story is that although distributed leadership can help jump-start a platform by signaling broad support from a key constituency (in this case, the handset producers) it may also interfere with a platform's ability to balance the interests of different user-groups and respond quickly to market developments. According to Lee Williams, who briefly ran the Symbian Foundation, "The broad range of operator/OEM support, and the extensive range of technology and device types couldn't help but make the platform difficult to market and ultimately difficult for others to accept as a good solution for the marketplace."[29] West and Wood (2013) also concludes that, "the divided leadership of the ecosystem limited the ability of Symbian and its ecosystem to respond to the new dominant design created by the iPhone." Ultimately, when Nokia emerged as the first among equals in Symbian deployment, rival handset suppliers had few qualms about switching to Android.

4 Implications for Policy and Management

The preceding case studies provide examples from each category in our classification scheme: stable forks (BSD and System V Unix), contested forks (Instant messaging, or HTML and Java during the browser wars), fragmentation (X2 and Flex in 56K modems) and splintering (early railroad gauges).[30] They also illustrate how forking, fragmentation and splintering can influence the evolution of commercially significant technology, regardless of whether that technology is a centrally managed platform like Symbian, a consensus standard like HTML, or a communications protocol such as AIM.

In general, the cases suggest that forking, fragmentation and splintering emerge when the costs and benefits of coordination are asymmetrically distributed. Our simple models highlight the role of asymmetry among technology sponsors. in particular, the payoffs imply that coordination can only occur if one player gives up the benefits of implementing their preferred technology. This assumption fits well with railroad gauges and 56K modems. However, the case studies also highlight the importance of vertical asymmetries that are not an explicit feature of our model. Even if customers derive large benefits from standardization, a supplier's private benefits may be scant if there are few options for product differentiation within an open platform. This can lead to the endogenous fragmentation that we observed among small ISPs in the 56K modem case, handset OEMs in the Symbian case, and workstation manufacturers in the Unix Wars, even though software developers and end-users might have preferred more inter-operability in each of those examples.

Finally, the case studies illustrate how technologies follow various paths to inter-operability. There were examples of coordination through SSOs, such as the ITU in the 56K modem case,

---

[29] Lomas, N. "A look back at Symbian on the eve of its demise." TechCrunch, June 13, 2013. Online at `https://techcrunch.com/2013/06/13/rip-symbian/`

[30] Of course, the cases do not tell us about the overall incidence of fragmentation and forking, since they are not drawn from a representative sample of standards and platforms.

IEEE/OSF for Unix and W3C for web standards. We also observed several platform leaders such as Symbian and the SRSA cartel. Converters and multi-homing provided temporary solutions in the railroad gauge and Instant Messaging cases respectively. Within each of these environments, the cases show how interested parties use a variety of tactics to encourage coordination. These tactics include aggressive pricing and copy-left open source licensing in the browser wars, the use of alliances and sponsored consortia in the evolution of Unix, and a combination of government intervention and bilateral contracting in instant messaging.

What advice can we give to those in charge of antitrust or innovation policy, who must often evaluate the tactics that firms use in these settings, as well as the resulting outcomes? Given the complex trade-offs among variety, inter-operability, competition and innovation highlighted above, the only foolproof answer is "it depends." However, like Farrell (2007), we suspect that it would be wise (on average) to have a preference for compatibility. This doesn't mean that policy should always work to prevent forking, or accept the idea that every restriction on fragmentation is necessarily pro-competitive. Rather, the idea is that policy makers should recognize that coordination is a messy process. Mistakes happen, and inefficient incompatibilities may persist.

A policy preference for inter-operability can nudge markets towards compatible competition, as the FCC did in the case of AOL and instant messaging. It can also consider the potential coordination benefits of anti-forking tools that might otherwise appear problematic. For example, the U.S. National Cooperative R&D Act provides certain antitrust exemptions to accredited Standards Developing Organizations.

Within a framework that has a preference for compatibility, policy-makers are likely to encounter two types of difficult cases. The first are modern analogues to the SRSA rail cartel, which encouraged efficiency-enhancing technical coordination, but perhaps only because it was able to deny consumers a large portion of the resulting benefits. Modern cases are less likely to involve explicit price-fixing, and more likely to focus on a platform leader's access and pricing policies.[31] In these cases, it is important to keep in mind that not all vertical restrictions are harmful, and that the resulting profit can spur innovation. However, the interests of end-users and platform leaders need not be perfectly aligned, and it is important to consider whether coordination is achieved in a way that preserves competition in complementary markets (Farrell and Weiser, 2003).

---

[31] Though, see the eBooks litigation in *United States of America v. Apple, Inc., et al.,* 12 Civ. 2826 (DLC).

The second type of difficult case involves forking of an established standard by a platform leader. This type of fork may degrade inter-operability while producing few benefits for complementers and consumers, as in the Microsoft antitrust case, where Microsoft allegedly forked Java in order to reduce the supply of complements to a nascent rival. But not all forks are bad, and a firm's rivals may have incentives to label any particularly successful implementation a "harmful fork" if it is based on proprietary technology that they would like to include in an easily accessible standard.

To evaluate these cases, policy makers should keep in mind some indicia of a "good fork." The social value of forking increases with uncertainty about future demand or performance, because experimentation can provide useful information. Furthermore, the costs of incompatibility decline when network effects are weak, when it is easy to patch together compatibility *ex post* (e.g. through converters), or when a coordinated *ex post* switch is fairly easy (e.g. because there are a small number of prospective adopters, or there exists a focal customer whom all will follow). In these cases decentralized competition between incompatible systems becomes more attractive, because excess variety is easily remedied. Finally, good forks add functionality without actively trying to degrade horizontal compatibility, and are more often clearly documented and explained.

When evaluating a fork, copyleft open-source licensing (and related governance mechanisms adopted by SSOs) can provide a useful reference point. Many open source projects attempt to strike a balance by reserving for standards implementers both a "right to fork" and a "right to standardize." For example, the GNU General Public License (GPL) guarantees users the right to copy, modify, and re-distribute any code covered by the GPL. This right-to-fork creates competitive pressure that can reduce problems of free-riding or stalemates in the standardization process. However, the GPL also requires implementers to grant the same copy and redistribution rights to third-parties. This right-to-standardize prevents extensions of an open standard from becoming proprietary *de facto* standards, thus reducing the threat that long-term inter-operability is extinguished.

We conclude by proposing a pair of managerial implications based on our exploration of forking, fragmentation and splintering. Both of these ideas flow from the observation that a messy coordination process that produces forking, fragmentation or splintering must (by definition) leave some economic surplus unrealized. This creates an opportunity for entrepreneurial managers, who may be able to capture some of the benefits created by engineering a switch to a more efficient outcome.

First, managers should be on the lookout for windows of opportunity that can emerge when there is an unmet need for coordination, or an imminent upgrade to a standard or platform. For example, in the 56K modem case study US Robotics took advantage of the upgrade cycle to achieve intellectual property parity with Rockwell Semiconductor. Similarly, the advent of smartphones provided an opportunity for Android to overtake Symbian in mobile operating systems.

Secondly, in order to take advantage of windows of opportunity, managers should have a good sense of the comparative advantages of different types of platform governance. SSOs provide a useful forum for reaching a compromise, as we saw in the modem and Unix cases. Platform leaders are often better positioned to engineer major upgrades, as we observed with the SRSA in railroads. For small firms, seeking to become a platform leader can be a risky and resource intensive strategy. Thus, it will often be wise to establish a strong presence within relevant SSOs, where technical expertise and coalition-building create opportunities to influence technology selection. Large firms may also find SSOs useful, particularly when they aim to promote innovation in complementary markets. However, as the case of Nokia and Symbian vividly illustrates, when core technologies experience rapid technological change, a shift towards active platform leadership may be called for.

References

Adner, R. and R. Kapoor (2010). Value creation in innovation ecosystems: How the structure of technological interdependence affects firm performance in new technology generations. Strategic Management Journal 31(3), 306–333.

Arthur, W. (1989). Competing technologies, increasing returns, and lock-in by historical events. Economic Journal 99(394), 116–131.

Augereau, A., S. Greenstein, and M. Rysman (2006). Coordination versus differentiation in a standards war: 56k modems. The RAND Journal of Economics 37(4), 887–909.

Axelrod, R., W. Mitchell, R. E. Thomas, D. S. Bennett, and E. Bruderer (1995). Coalition formation in standard-setting alliances. Management Science 41(9), 1493–1508.

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

Bernheim, B., B. Peleg, and M. D. Whinston (1987). Coalition-proof nash equilibria i. concepts. Journal of Economic Theory 42(1), 1–12.

Besen, S. M. and J. Farrell (1994). Choosing how to compete - strategies and tactics in standardization. Journal of Economic Perspectives 8(2), 117–131.

Boudreau, K. J. (2010). Open platform strategies and innovation: Granting access vs. devolving control. Management Science 56(10), 1849–1872.

Boudreau, K. J. and A. Hagiu (2009). Platforms rules: Multi-sided platforms as regulators. In A. Gawer (Ed.), Platforms, Markets and Innovation. Cheltenham, UK: Edward Elgar Publishing.

Bresnahan, T. (2002). The economics of the Microsoft case. Stanford Law and Economics Olin Working Paper No. 232.

Bresnahan, T. F. and P.-L. Yin (2007). Standard setting in markets: The browser war. In S. Greenstein and V. Stango (Eds.), Standards and Public Policy, Chapter 1, pp. 18–59. Cambridge, UK: Cambridge University Press.

Cabral, L. and D. Salant (2014). Evolving technologies and standards regulation. International Journal of Industrial Organization 36, 48–56.

Cargill, C. F. (2011). Why standardization efforts fail. Journal of Electronic Publishing 14(1).

Coase, R. (1960). The problem of social cost. Journal of Law and Economics 3, 1–44.

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

David, P. A. (1985). Clio and the economics of QWERTY. American Economic Review 75(2), 332–337.

Evans, D. and R. Schmalensee (2002). Some economic aspects of antitrust analysis in dynamically competitive industries. In A. B. Jaffe, J. Lerner, and S. Stern (Eds.), Innovation Policy and the Economy, Volume 2. National Bureau of Economic Research.

Farrell, J. (2007). Should competition policy favor compatibility? In S. Greenstein and V. Stango (Eds.), Standards and Public Policy, Chapter 12, pp. 372–388. Cambridge, UK: Cambridge University Press.

Farrell, J. and M. Rabin (1996). Cheap talk. Journal of Economic Perspectives 10(3), 103–118.

Farrell, J. and G. Saloner (1986). Installed base and compatibility: Innovation, product preannouncements, and predation. American Economic Review 76(5), 940–955.

Farrell, J. and G. Saloner (1988). Coordination through committees and markets. The RAND Journal of Economics 19(2), 235–252.

Farrell, J. and T. Simcoe (2012a). Choosing the rules for consensus standardization. The RAND Journal of Economics 43(2), 235–252.

Farrell, J. and T. Simcoe (2012b). Four paths to compatibility. In M. Peitz and J. Waldfogel (Eds.), Oxford Handbook of the Digital Economy, Chapter 2, pp. 34–58. Oxford, UK: Oxford University Press.

Farrell, J. and P. Weiser (2003). Modularity, vertical integration, and open access policies: Towards a convergence of antitrust and regulation in the internet age. Harvard Journal of Law and Technology 17(1), 85–134.

Faulhaber, G. (2002). Network effects and merger analysis: instant messaging and the AOL–Time Warner case. Telecommunications Policy 26(5), 311–333.

Faulhaber, G. (2004). Access and network effects in the "new economy": AOL-Time Warner (2000). In J. Kwoka and L. White (Eds.), The Antitrust Revolution: Economics, Competition, and Policy, pp. 453–475. Oxford, UK: Oxford University Press.

Fudenberg, D. and J. Tirole (1987). Understanding rent dissipation: On the uses of game theory in industrial organization. The American Economic Review 77, 176–183.

Gabel, D. (1994). Competition in a network industry: The telephone industry, 1894–1910. The Journal of Economic History 54(3), 543–572.

Gawer, A. and M. Cusumano (2014). Industry platforms and ecosystem innovation. Journal of Product Innovation Management 31(3), 417–433.

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

Gilbert, R. J. and M. L. Katz (2001). An economist's guide to US v. Microsoft. Journal of Economic Perspectives 15(2), 25–44.

Greenstein, S. (2000). Building and delivering the virtual world: commercializing services for internet access. Journal of Industrial Economics 48(4), 391–411.

Greenstein, S. and M. Rysman (2007). Coordination costs and standard setting: lessons from 56k modems. In S. Greenstein and V. Stango (Eds.), Standards and Public Policy, Chapter 4, pp. 123–159. Cambridge, UK: Cambridge University Press.

Gross, D. P. (2016). The ties that bind: Railroad gauge standards and internal trade in the 19th century US. Working Paper.

Hudson, H. (1890). The southern railway and steamship association. Quarterly Journal of Economics 5(1), 70–94.

Katz, M. and C. Shapiro (1985). Network externalities, competition, and compatibility. American Economic Review 75(3), 424–440.

Klepper, S. and K. Graddy (1990). The evolution of new industries and the determinants of market structure. The RAND Journal of Economics 21(1), 27–44.

Kretschmer, T. (2008). Splintering and inertia in network industries. The Journal of Industrial Economics 56(4), 685–706.

Leibowitz, S. and S. Margolis (1990). The fable of the keys. Journal of Law and Economics 30(1), 1–26.

Lerner, J. and J. Tirole (2001). The open source movement: Key research questions. European Economic Review 45(4), 819–826.

Levinson, M. (2006). The Box: How the Shipping Container Made the World Smaller and the World Economy Bigger. Princeton, NJ: Princeton University Press.

Matutes, C. and P. Regibeau (1988). "mix and match": Product compatibility without network externalities. The RAND Journal of Economics 19(2), 221–234.

O'Mahony, S. and R. Karp (2017). From proprietary to collective governance: How platform participant strategies adapt. Working Paper.

Parker, G., M. Van Alstyne, and X. Jiang (2017). Platform ecosystems: How developers invert the firm. Management Information Systems Quarterly 41(1), 255–266.

Postrel, S. R. (1990). Competing networks and proprietary standards: The case of quadraphonic sound. The Journal of Industrial Economics 39(2), 169–185.

Puffert, D. J. (2009). Tracks across Continents, Paths through History: The Economic Dynamics of Standardization in Railway Gauge. Chicago, IL: University of Chicago Press.

Salus, P. H. (2015). Dueling UNIXes and the UNIX wars. ; login:: the magazine of USENIX & SAGE 40(2), 66–68.

Shapiro, C. and H. R. Varian (1999). Information Rules: A Strategic Guide to the Network Economy. Brighton, MA: Harvard Business Review Press.

Siddall, W. R. (1969). Railroad gauges and spatial interaction. Geographical Review 59(1), 29–57.

Simcoe, T. (2012). Standard setting committees: Consensus governance for shared technology platforms. American Economic Review 102(1), 305–336.

Stallman, R. et al. (1985). The GNU manifesto.

Suarez, F. and J. Utterback (1995). Dominant designs and the survival of firms. Strategic Management Journal 16(6), 415–430.

Thompson, G. V. (1954). Intercompany technical standardization in the early American automobile industry. The Journal of Economic History 14(1), 1–20.

Tsai, J. and J. Wright (2015). Standard setting, intellectual property rights, and the role of antitrust in regulating incomplete contracts. Antitrust Law Journal 80(1), 157–183.

UNESCAP (1996). Trans-Asian Railway Route Requirements: Development of the Trans-Asian Railway in the Indo-China and ASEAN Subregion. United Nations Economic and Social Commission for Asia and the Pacific.

UNTC (2006). Intergovernmental Agreement on the Trans-Asian Railway Network. United Nations Treaty Collection.

Weiss, M. B. and M. Sirbu (1990). Technological choice in voluntary standards committees: An empirical analysis. Economics of Innovation and New Technology 1(1-2), 111–134.

West, J. (2014). Challenges of funding open innovation platforms. In H. Chesbrough, W. Vanhaverbeke, and J. West (Eds.), New Frontiers in Open Innovation, Chapter 4, pp. 70–93. Oxford, UK: Oxford University Press.

West, J. and D. Wood (2013). Evolving an open ecosystem: The rise and fall of the Symbian platform. Advances in Strategic Management 30, 27–67.

Weyl, E. G. and A. White (2014). Let the right ”one” win: Policy lessons from the new economics of platforms. Competition Policy International 10(2), 28–51.