

## **Forking, Fragmentation and Splintering\***

Timothy Simcoe and Jeremy Watson  
Boston University, Questrom School of Business

*October 28, 2016*

### *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 economics 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 welfare effects of efforts to promote inter-operability.

Keywords: Forking, fragmentation, splintering, standards, platforms.  
JEL Codes: O31, L44, L17.

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 complements.

In simple economic models, the benefits of interoperability produce network effects that cause markets to “tip” towards a dominant standard. But in practice, there are many cases of persistent incompatibility, inter-platform competition and standards proliferation. For example, many U.S. cell phones do not work in Europe, where carriers use different transmission standards. Modern web browsers support dozens of different audio and video file formats. And 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 tend to view forking and fragmentation as evidence of coordination failure. That is, they believe that having fewer standards is generally better, and that markets often fail to produce that outcome. Many economists recognize the possibility of coordination failure, but also highlight the potential trade-off between variety and compatibility. 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 about right. The issue is complex, and large firms can find themselves on both sides of this debate. For example, Google has recently been criticized (and sued) for forking Java when it created the Android operating system, and at the same time, is under investigation for anti-fragmentation provisions in its Android licensing agreements.<sup>1</sup>

---

<sup>1</sup> For a contemporaneous account of the Java-Android forking dispute see, for example, <https://www.cnet.com/news/googles-android-parts-ways-with-java-industry-group/>. The European Commissions preliminary views on anti-fragmentation provisions in Android licenses agreements are at [http://europa.eu/rapid/press-release\\_IP-16-1492\\_en.htm](http://europa.eu/rapid/press-release_IP-16-1492_en.htm).

This paper examines the economics of forking, fragmentation and splintering in markets with network effects. Economic theory suggests that coordination failures can occur, and may be persistent. Theory also helps clarify the conditions that make inefficient splintering more likely. We illustrate the causes of forking, fragmentation and splintering, as well as the tools that firms and industries use to combat miscoordination, through several short cases. The cases examine fragmentation and standardization in railroad gauges, modems, mobile operating systems, instant messaging and Internet browsers. In the conclusion, we turn to the difficult question of how to gauge the welfare effects of efforts to promote inter-operability.

### ***1. Flavors of Forking***

Incompatibility can emerge in a variety of ways, and perhaps appropriately, economists have not converged on a standard terminology for differentiating them. In this essay, we distinguish between the closely related but conceptually distinct ideas of fragmentation, forking and splintering.

By our definition, a market is *splintered* whenever its key products or platforms exhibit persistent incompatibility. For example, Gabel (1994) describes incompatibilities in US long-distance telephone service that lasted from 1894 through 1910. During this period, AT&T refused to interconnect with independent local exchanges, and the regional networks that were formed to connect independent local exchanges often relied on different technologies. Below, we describe similar episodes of fragmentation in the adoption of gauge standards for the U.S. railroad network and Internet-based instant messaging software.

To be clear, splintering is not necessarily harmful. Users may prefer incompatible competition to a monopoly-owned standard. Splintering also increases variety, which is valuable when users have different tastes. However, splintering can be inefficient when network effects are strong, when users have similar tastes (so there is limited scope for product differentiation), and when the sunk costs of technology adoption make it difficult to achieve *ex post* compatibility through converters, multi-

homing or a coordinated switch. Although inefficient splintering creates incentives for standardization, the case studies below suggest that there are often parties who will oppose efforts to coordinate on any particular standard.

We use the terms *fragmentation* and *forking* to denote an increase in the number of incompatible choices. Under our definitions, forking is distinguished from fragmentation by a lack of consensus about the degree of inter-operability, as opposed to merely the choice of a particular standard. In game-theoretic terms, fragmentation represents a coordination failure in a setting where incentives resemble a “battle of the sexes” so that everyone prefers a common standard, even if they disagree on what it should be. Forking, on the other hand, occurs when payoffs resemble a game of “pesky little brother” (or “matching pennies”), where some players prefer to coordinate and others do not.<sup>2</sup>

In practice, the lines between forking, fragmentation and splintering can be blurry. However, all three are examples of what Farrell (2001) calls *horizontal incompatibility*: complements for one system cannot be used with a rival standard or platform. For example, horizontal incompatibility in the fragmented market for video game consoles implies that software developed for the 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

---

<sup>2</sup> There is no pure strategy equilibrium in Matching Pennies, and the mixed-strategy equilibrium might lead to either coordination or splintering. In a Battle of the Sexes, players always coordinate in pure strategy equilibria, but there is also a mixed-strategy equilibrium where they may not.

applications.<sup>3</sup> As this example suggests, the key vertical question is often whether it is necessary to regulate the access policies 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?

## ***2. Equilibrium Incompatibility***

As a starting point for analyzing the horizontal interoperability question, it is worth noting that splintering may be an equilibrium outcome. For example, consider a simple game of complete information with three players ( $i = 1,2,3$ ) who must choose among three possible standards ( $j=a,b,c$ ). To capture the idea that there are benefits from coordination, even if each player prefers a different standard, let player  $i$ 's payoff from choice  $j$  equal \$2 times the total number of players who choose standard  $j$ , plus an additional \$3 if  $j$  is player  $i$ 's preferred option.

It is easy to verify that one Nash equilibrium of this game is for each player to select their preferred standard: the splintered equilibrium yields a payoff of \$5 to each player, and any unilateral deviation pays at most \$4. However, it is also a Nash equilibrium for all three players to choose the same standard and receive a payoff of \$6 (or \$9 for the lucky player whose preferred outcome is chosen). The key point of this example is that even if every player wants to avoid fragmentation, it takes *coordinated* action to escape a splintered equilibrium.<sup>4</sup>

Do we observe persistent splintering in more realistic settings, where players can bargain and communicate? Postrel (1990) suggests that fragmentation prevented quadrophonic sound from overtaking stereo recording systems in the early 1970s. In that episode, CBS, JVC and RCA each sponsored a different quadrophonic-sound standard, leading to weak availability of complements (i.e. recorded music) and

---

<sup>3</sup> This example is due to Farrell (2007), who offers a longer discussion. See also Bresnahan (2002).

<sup>4</sup> In the language of game theory, splintering may be an equilibrium, even though it is not coalition proof.

slow end-user adoption. Thomson (1954) argues that the early U.S. automotive industry suffered from excessive variety. Before the rapid growth of Ford and General Motors, there were a host of automotive component manufacturers who 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 manufacturers had to fit a variety of springs and so forth. The costs of managing a wide variety of incompatible components grew large, and Thompson describes how this situation, along with the growing competitive threat posed by Ford and GM, led to the creation of the Society of Automotive Engineers (SAE). Much of the SAE's early technical standardization work focused on reducing component variety.

The quadrophonic sound and SAE examples highlight two different types of fragmentation. The first type of coordination failure results from many adopters making decentralized choices among a wide variety of options, such that accounting for others' choices is difficult. This seems to have been the case in the early auto industry, where component suppliers chose specifications to suit their own needs, giving little thought to the downstream benefits of standardization. A second type of coordination failure occurs when key players have divergent preferences over the choice of the standard, and each push for their own preferred solution. This describes the situation faced by quadrophonic sound, and also the transition to 56K modems that we describe below. Farrell and Saloner (1986) show how this second type of fragmentation can lead to "excess momentum" for an incumbent technology if users delay adoption because they do not want to pick the losing standard.

We can also find persistent splintering in settings where variety and specialization are valuable. 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 that encourage coordination on a small number of languages. 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 there are clear compatibility benefits from file-sharing.<sup>5</sup>

Subtle trade-offs between variety and compatibility can lead to disagreements and forking. For example, many programming languages are variants of the “C” language invented by Dennis Ritchie in the early 1970s, and we describe below how the Unix operating system has forked on several occasions. Forking often resembles the type of fragmentation that occurs when competing technology sponsors hold out for alternative standards, except that with forking, one of the options already has an installed base.

New standards or platforms that emerge from a fork may be vertically open or closed.<sup>6</sup> Open forks increase variety, may promote competition and are seldom controversial. Open source licenses that contain copy-left provisions (which mandate free access to source code for derivative works) could even be viewed as bundling a “right to fork” with a commitment to vertical openness. When there is little switching away from the incumbent technology to an open fork, then the welfare implications of the fork are probably small (even if one believes that a coordinated switch would be desirable). If there is switching, the new variant presumably offers some advantage over the status quo.

Forks can also be vertically closed. For example, new entrants often launch incompatible platforms and standards implementers can add proprietary features to a specification. Closed forms are often more controversial. Like other vertical

---

<sup>5</sup> When standards proliferate, converter technologies are often used to provide a limited degree of inter-operability. For example, there are “container” standards for audio and video files that help web browsers work with a wide-variety of different formats. And computer programmers often use integrated development environments that facilitate collaboration even among coders writing in different languages (O’Mahoney et al, 2005).

<sup>6</sup> This distinction is correlated with, but conceptually distinct from property rights or “sponsorship” which Katz and Shapiro (1994) emphasize provides incentives to solve coordination problems.

restrictions, closed forks are not necessarily harmful, but they could be.<sup>7</sup> One scenario that has received considerable attention is when a dominant platform forks an open standard merely to degrade interoperability, thereby reducing the supply of complements to a nascent rival. The most famous example of this strategy involves Microsoft's proprietary flavors of Java and HTML, which allegedly made it harder for software developers to produce code that could easily be ported from Windows to rival platforms. On the other hand, closed forks allow implementers to restrict access to proprietary features, and the resulting profit could spur innovation. Indeed, a firm's rivals might be inclined to label any particularly successful implementation a "fork" if it is based on proprietary technology that they would like to include in an easily accessible standard.

### ***3. Fighting Forking***

Although splintering can occur in equilibrium, it is not the only possible outcome. When network effects are strong, players have a strong incentive to avoid fragmentation, and several tools for doing so.<sup>8</sup>

Standard Setting Organizations (SSOs) such as the SAE and the IEEE play an important coordinating role in many industries. These groups provide a forum for reaching consensus on the standard itself, and also work to promote widespread adoption and compliance.

Platform leaders are another important type of coordination mechanism. The defining feature of a platform leader is that they have clout. Once a platform leader decides which way to go, it is generally in the interest of others to follow. 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

---

<sup>7</sup> See Farrell and Weiser (2003) for an extended discussion of platforms' vertical access policies.

<sup>8</sup> Farrell and Simcoe (2013) suggest that there are four main tools for coordination: SSOs, platform leaders, decentralized adoption and converters or multi-homing. We focus on the first two options.



that third-party software developers can use to create new products and services. At the most general level, a platform leader could be a large customer or even the government.

In the simple numerical example above, any first-mover could play the role of a platform leader. In particular, if one of the three players publicly commits to their preferred standard, the others should follow, since doing so yields a payoff of \$6, compared to \$5 in the best alternative equilibrium. A platform leader may also be able to manipulate the choice of standards by altering other players' payoffs. For example, if player 1 can offer a "bribe" of \$1.01 for choosing its preferred standard, it then becomes a dominant strategy for player 2 to go along.

In reality, monetary side-payments are rare, but SSOs and large platform leaders have a variety of other instruments they can use to cajole followers into coordinating on a single standard. In fact, the economic literature on "standards wars" provides an extensive list of tactics to encourage coordination.

One obvious way to promote a standard is to price it aggressively. Many SSOs make their standards freely available. For platform leaders, aggressive pricing is often part of a broader strategy that involves subsidizing one group of users to encourage adoption, while extracting rents from another group that is less price-sensitive (Rysman 2009). However, Weyl and Whyte (2014) emphasize that while platforms' "freemium" or "usage revenue later" pricing strategies may help solve consumer coordination problems, fragmentation can still be a problem if too many firms enter markets where strong network effects lead to natural monopoly profits.

SSOs and platform leaders also use a wide variety of non-price instruments to promote standards adoption and compliance. For example, many software platforms supply application programming interfaces (APIs) and software development kits (SDKs) to their developers. Test suites, technical roadmaps and reference implementations are also widely tools for reducing the costs of developing

complements while ensuring broad adherence to underlying standards. Most large platform sponsors host “developer forums” to communicate important changes to their developer base, and many SSOs organize “plug fests” or inter-operability events where independent vendors come together to see if their products work together well.

Another tactic to promote coordination is to try and influence expectations. Because expectations can be self-reinforcing in the presence of network effects, a reputation for setting the standard is arguably one of the key advantages held by an established platform leader or SSO. Platform providers are notorious for inflating statistics about their current installed base of users, in part because a large network signals to customers that the market is likely to “tip” towards the leading platform.

SSOs and platform leaders also use intellectual property to encourage coordination. For example, many SSOs require contributors to make Reasonable and Non-Discriminatory (RAND) licensing commitments, so that future implementers have some assurance that they can access patented technologies needed to comply with a specification (Farrell, Hayes, Shapiro and Sullivan 2007). Certain types of open source software licensing go one step further, by requiring implementers to place innovations that build on a standard back into the public domain, so they can be accessed by future users.

SSOs and platform leaders may also prevent fragmentation, forking and splintering by conditioning access to intellectual property on compliance with the standard. For example, most RAND commitments extend only to compliant implementations of the relevant standard. Some SSOs have copyright-based anti-forking policies that prohibit using the text of their standards as the basis for a proprietary extension. And for established standards such as Wi-Fi or USB, where consumer brand recognition is important to implementers, access to trade-marks and participation in promotional activities can provide another means for enforcing compliance. Red

Hat, for example, withdraws access to its trademarks and support services for customers that create an incompatible fork.

#### ***4. Case Studies***

Despite the wide variety of tools available to an SSO or platform leader that wishes to promote inter-operability, the process of coordinating on a standard is not always smooth or successful. This section presents several case studies of industries, technologies or standards that fragmented, forked or splintered.

##### *4.1 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 the more common 4-foot 9-inch width, making them compatible with the bulk of the Northern rail network. This episode is described in both Varian and Shapiro (1999) and Gross (2016).<sup>9</sup> A historical examination of railroad standards illustrates how compatibility matters outside of the information technology sector, and how fragmented outcomes can emerge and persist even when the costs of incompatibility are large.

The first efforts to build commercial rail service in the United States occurred in the 1820's and 1830s. Most lines offered 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 US railroad network, and rail came to replace canals 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 network, the initial heterogeneity in gauge standards persisted. Sidall (1969) reports that there were at least 23 different gauge standards in use during the 1860s.

---

<sup>9</sup> This account draws heavily on the historical section of Gross (2016) and the references cited therein.

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. Railroads also experimented with adjustable width rolling stock, and multi-gauge track (i.e. a third rail). However, none of these converter technologies was completely effective at removing delays or matching the overall performance of gauge-specific rolling stock.<sup>10</sup>

Over time, the costs of incompatibility scaled with utilization of the rail network, creating strong incentives for further convergence during the Civil War and

---

<sup>10</sup> Levinson (2006) provides a related account of the costs of break-bulk shipping in ocean transport prior to the arrival of containerization.

reconstruction. By the 1880's, through both conversion and new construction, the US 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 a nationwide interoperability was a remarkable 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 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. 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 produced a 27 percent price decline and a 20 percent increase in shipments for the routes in his sample.

The railroad case study offers several important lessons about fragmentation and forking. 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).

#### *4.2 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, Greenstein and Rysman (2006), and Greenstein and Rysman (2012) 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.<sup>11</sup>

Prior to 1997, modems operated at a maximum speed of 33K. The market for 33K modem chipsets was dominated by Rockwell Semiconductor. Rockwell licensed its technology to various resellers whose combined market share exceeded 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 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, 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

---

<sup>11</sup> Our account is largely based on the discussion in Greenstein and Rysman (2012)

market leader. None of the major ISPs (AOL, AT&T, UUNET, MSN, GTE, BellSouth or EarthLink) adopted 56K during this time. 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, Rysman and Greenstein (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.<sup>12</sup> Because SSOs lack formal enforcement power, it is not unusual for them to wait and see whether a *de facto* standard emerges in the market prior to endorsing any particular solution in a standards war.<sup>13</sup> Moreover, participants in the formal standards process are typically interested parties, which in this case would include members of the US Robotics and Rockwell-led consortia, and perhaps a few of the larger ISPs. However, Greenstein and Rysman (2012) report that both the X2 and Flex consortia expected to adopt an ITU standard. Additionally the slow adoption of 56K technology by consumers and large ISPs placed some pressure on the SSOs to act quickly, in order to break to the logjam that was holding back demand.

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

---

<sup>12</sup> The TIA is US industry association that develops standards under the auspices of ANSI, and can therefore serve as the US representative to ITU, which is a Geneva-based UN treaty organization. ITU has set a variety of international telecommunications standards since the late 1800s.

<sup>13</sup> 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.



“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. Although large ISPs sat on the sidelines, rather than make a risky bet on a single standard that might lead to stranded investments, the smaller ISPs actually exacerbated the fragmentation problem. These small firms viewed incompatibility as a potential source of differentiation in a highly competitive industry, so that 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, Rysman and Greenstein (2012) 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 a lead in the marketplace. 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.

#### *4.3 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.

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. Fragmentation followed. 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.”

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. Network file storage protocols also played an important role in the initial Unix battles. Sun released its Network File System (NFS) and AT&T promoted its own Remote File System (RFS). While RFS has been considered a superior model, NFS eventually won out due to Sun's sharing of open-source code along with specifications.

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. However, the first round of the Unix Wars essentially ended in a stalemate between the BSD camp and the System V camp.<sup>14</sup>

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

---

<sup>14</sup> 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.”

Open Software Foundation (OSF), a consortium whose key members included Digital Equipment, Hewlett Packard, and IBM.<sup>15</sup>

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, that Sun pioneered, 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 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 offer several broad lessons on the economics of forking, fragmentation 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

---

<sup>15</sup> According to Axelrod et al. (1995), Sun's CEO Scott McNealy joked that OSF actually stood for "Oppose Sun Forever."

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* of Unix were minicomputer and workstation producers who often preferred a proprietary flavor of Unix that could provide a greater level of product differentiation.

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. 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. Finally, one of the most interesting questions posed by the Unix wars 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.

#### *4.4 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.<sup>16</sup>

A number of AOL’s competitors also deployed proprietary messaging protocols in the late 1990s. 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).<sup>17</sup> 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 of the

---

<sup>16</sup> Perhaps ironically, OSCAR stands for Open System for Communicating in Realtime.

<sup>17</sup> Examples include the Session Initiation Protocol (SIP), Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE), Application Exchange (APEX), Instant Messaging and Presence Protocol (IMPP), and the Extensible Messaging and Presence Protocol (XMPP).

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).<sup>18</sup> 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

---

<sup>18</sup> 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.

the regional phone networks eventually merged to monopoly, the instant messaging market evolved differently. The FCC intervened to promote horizontal interoperability, 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, open technology can often provide a foundation for subsequent iterations of the platform.

#### *4.5 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.”<sup>19</sup>

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.<sup>20</sup> 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 (2005) 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 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 such as “Best Viewed With Internet Explorer” or “Best Viewed With Netscape Navigator.” Another common practice for web developers during this time was to utilize scripts that detected the visitor's browser version, and load the appropriate version of the website tailored to a specific browser. Such 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 the first stage of the browser wars, the World Wide Web Consortium (W3C), an SSO founded by Tim Berners-Lee in 1994, worked to prevent fragmentation of key document standards by publishing specifications for Hypertext

---

<sup>19</sup> The complete memo from gates is available at <http://www.lettersofnote.com/2011/07/internet-tidal-wave.html>

<sup>20</sup> Microsoft licensed much of its original browser code from Spyglass Mosaic.

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).<sup>21</sup>

By the early 2000’s, Internet Explorer had a 90 percent share of the web browser market, and Microsoft’s strategy towards the open web was increasingly characterized as an effort to “embrace, extend and extinguish” a set of standards that might threaten the dominance of its Windows platform (Gilbert and Katz 2001).<sup>22</sup> The antitrust concerns raised in DOJ versus Microsoft stem from the fact that its forks of HTML (and later Java) were vertically closed, and that by limiting Netscape’s access to the Windows platform, Microsoft might be able to limit long-run competition from browser-based rivals.

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 1985; Lerner and Tirole 2000). Broadly speaking, a copy-left provision requires 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.<sup>23</sup> Under a copy-left licensing regime, it is possible to

---

<sup>21</sup> ECMA was originally an acronym for the European Computer Manufacturers Association.

<sup>22</sup> This three-step strategy begins when a platform leader embraces a standard by providing vertical interoperability. 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. Microsoft was accused of intentionally breaking Java portability in this manner (in violation of contractual commitments with Sun), and paid \$2 billion to settle the resulting legal dispute.

<sup>23</sup> 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,

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.<sup>24</sup>

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 penetration 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

---

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.

<sup>24</sup> 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.

has a copyright policy that allows others to use its standards as the basis for further specification development. The W3C, on the other hand, has a more restrictive policy that does not allow this type of specification forking.<sup>25</sup> 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 fragmentation and forking. Once again, the initial battles between Netscape and Microsoft show how incentives to differentiate by adding new features to a standard can degrade interoperability and impose negative externalities on other platform users. 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 fragmentation was ultimately averted with help from groups like W3C and WaSP.

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. Netscape's 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. Finally, the 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

---

<sup>25</sup> 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).

future implementers retain the ability to access the fork (and therefore the ability to merge useful innovations back into the standard).<sup>26</sup>

#### *4.6 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.<sup>27</sup>

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

---

<sup>26</sup> 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.

<sup>27</sup> Our short history of Symbian draws upon the more detailed case studies by West and Wood (2013) and West (2014).

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 allowed Research in Motion (BlackBerry) to capture a substantial share of the early US 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 splintering 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."<sup>28</sup> West (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.

---

<sup>28</sup> 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/>

## 5. Welfare and Policy Implications

The preceding case studies show that 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 Unix, or a communications protocol such as AIM.<sup>29</sup> We also saw how splintering can be persistent, as with Unix or railway gauges, or short-lived, as in the case of 56K modems.

The cases also provide suggestive evidence about factors that increase the likelihood of forking, fragmentation and splintering. In some cases, such as railroad gauges and the early Unix variants, splintering emerges from decentralized experimentation and technology adoption by users who are not thinking about system-wide scale economies from inter-operability. This type of accidental forking may lead to inefficient equilibria, featuring excessive variety, as we saw in the case of railroad gauges and the early auto industry. However, a somewhat speculative (and optimistic) reading of the evidence suggests that this problem can generally be resolved if the costs and benefits of inter-operability are evenly distributed. For instance, the early work of the SAE helped reduce variety in auto component dimensions, while the SRSA helped coordinate Southern railways' expectations regarding a gauge switch.

But when the costs and benefits of adopting a standard are distributed asymmetrically, the coordination process can become rather messy. In some cases, this type of asymmetry reflects an incidence problem, where the group that adopts the standard does not receive the bulk of the benefits. For example, 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 differentiation that we observed among

---

<sup>29</sup> 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.



ISPs in the 56K modem case, or among handset OEMs in the Symbian case, or among workstation manufacturers in the Unix Wars, even though software developers and end-users might have preferred more inter-operability in each of those examples. Another type of asymmetry can emerge when rival sponsors prefer to adopt different technologies as the standard. This happened with Rockwell versus US Robotics in 56K modems, and the BSD versus System V camps in the Unix wars. In practice, firms may prefer different standards for a variety of reasons, including intellectual property in alternative solutions; differences in the lead times or redesign costs from standardizing on a particular technology; the degree of backwards compatibility with an installed base, or the ease of integration with proprietary complements.

Beyond showing when and why splintering may happen, the case studies also illustrate tools that interested parties can use to fight forking and fragmentation. At the most basic level, there were examples of coordination through SSOs, such as the ITU in the 56K modem case, and through platform leaders such as Symbian or the SRSA cartel. The cases also illustrate a variety of tactics (not always successful) to combat splintering, such as 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 does all of this tell us about the welfare effects of forking, fragmentation and splintering? The answer, of course, is “it depends.” It is certainly possible that everything comes out right when balancing the interests of parties that favor interoperability with those that favor variety. However, we know of no such theorem stating that should be the case. Rather, we believe it makes sense to evaluate the welfare impacts of forking and fragmentation by articulating the basic trade-offs, and considering factors in individual cases that increase or reduce the importance of key factors.

At bottom, there is a fundamental trade-off between the benefits of variety and the costs of incompatibility. Fragmentation is more likely to be harmful when consumers have little taste for variety, and derive large benefits from coordination. For example, it would clearly be a bad idea to fork the standard for driving on the right (or left) side of the road. Similar considerations apply in selecting the width of a railroad gauge, or the shape of a cellular phone charger. However, when users have a strong taste for variety, it may be efficient to forgo some compatibility even when network effects are strong. For example, there are a wide variety of different document and media-file formats, reflecting trade-offs among size, fidelity, simplicity and speed that make different formats work better for different applications, even if those differences also reduce overall portability.

The trade-off between variety and compatibility is complicated by the fact that splintering can influence competition and innovation. Splintering leads to horizontally incompatible inter-platform competition, which economic theory suggests can be especially intense if rivals expect the market to tip, leading to a period of *ex post* market power. Some economists suggest that “serial dominance” is a natural outcome in markets with strong network effects, and that threat of displacement from innovation produces competitive discipline (e.g. Evans and Schmalensee 2002). However, incompatible competition can raise complementors’ costs if they need to incur a separate entry cost for each platform.<sup>30</sup> And if the market does tip, there might be reasons to be concerned about the incentives and actions of a platform leader that lacks competition (Farrell and Weiser 2003).

Standards mavens generally prefer horizontally compatible intra-platform competition. The general idea is captured by the old technology industry adage that firms should, “cooperate on standards and compete on implementation.” When network effects are strong and the opportunities for differentiation on standardized features are few, this may be quite attractive. The potential problem in this case is

---

<sup>30</sup> Corts and Lederman (2009) study multi-homing in the video game industry, and emphasize that the costs of porting complements across platforms may also change over time.

that no one firm is responsible for the standard or platform (as we saw with Symbian, where ownership was divided among the competing handset manufacturers). In general, a platform leader may have more tools and a stronger interest in managing its complementors, especially if it incurs a direct cost of supporting multiple complements. Thus, it is relatively common to impose some level of standardization on vertical complements, through APIs and test suites. Platform providers can also internalize adverse selection issues, and police quality to provide incentives for producing high quality complements. While the participants in an SSO-managed platform may jointly have similar incentives, they do not have the same individual costs and benefits, and this can lead to free-riding problems.

Although the case studies above will not resolve the debates over compatibility and competition, they do highlight problems with extreme or simplistic versions of either position. For example, the history of instant messaging (and before it the telephone), the Unix wars and the 56K modem case all illustrate that markets need not tip, as predicted by theories of serial dominance. Inefficient incompatibilities may persist, and conversely, the period of *ex post* monopoly can last for decades. On the other hand, the railroad gauge and Symbian cases beg the question of whether inter-operability can be achieved in the absence of a powerful platform leader that can help internalize some of the costs and benefits across different types of system adopter.<sup>31</sup>

Moreover, just as the relationship between fragmentation and competition is complex, there is no clear *a priori* relationship between forking and innovation. When fragmentation prevents adoption of a standard, it can clearly reduce incentives for innovation, sometimes to the point that a standard simply collapses. This was arguably the case with quadrophonic sound, and can also be used to understand why Nokia's efforts to rescue Symbian by making it open-source failed

---

<sup>31</sup> The long slow transition to a new version of the Internet Protocol (from IPv4 to IPv6) is another example of a standard that stalled without sponsorship.

to launch. On the other hand, the forks that led to BSD Unix and the formation of WHATWG to advance HTML standardization arguably accelerated the pace of innovation in their respective platforms. As we discussed in the case of the browser wars, copyleft open-source licensing (and related governance mechanisms adopted by SSOs) attempt to strike a balance by reserving for standards implementers both a “right to fork” and a “right to standardize.” The right-to-fork creates competitive pressure that can reduce problems of free-riding or stalemates in the standardization process. The 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. Even if splintering persists, thereby reducing the size of the addressable installed base, it might promote innovation by alleviating free-riding problems that would arise if there were a single shared platform.<sup>32</sup>

Given the complex trade-offs among variety, inter-operability, competition and innovation, what advice can we give to policy-makers? Farrell (2007) considered whether competition policy should favor compatibility, and concluded that (on average) it should. 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 US National Cooperative R&D Act provides certain antitrust protections to accredited Standards Developing organizations. The difficult cases for any such policy are likely to be modern

---

<sup>32</sup> For a formal model of this idea, see Cabral and Salant (2014).

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 benefits that coordination produced.

In analyzing the complex cases, policy makers might keep in mind some indicia of a “good fork.” First, forking is more useful when there is lots of uncertainty about demand or performance, because in that case, experimentation can provide useful information. Second, the costs of fragmentation 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 relatively more attractive, because excessive fragmentation is easily remedied. Finally, good forks add functionality without actively trying to degrade horizontal compatibility. They are more likely to be clearly documented and explained. And, building on the open-source licensing insight, good forks are more likely to provide access rights to future implementers, so that new functionality can be standardized if it proves valuable.

## Bibliography

- Augereau, A., Greenstein, S., & Rysman, M. (2006). Coordination versus differentiation in a standards war: 56K modems. *The RAND Journal of Economics*, 37, 887-909.
- Axelrod, R., Mitchell, W., Thomas, R. E., Bennett, D. S., & Bruderer, E. (1995). Coalition formation in standard-setting alliances. *Management science*, 41, 1493-1508.
- Baron, D. (2013, May 22). *David Baron's Weblog*. Retrieved July 26, 2016, from <http://dbaron.org/log/20130522-w3c-licensing>
- Blagdon, J. (2012, July 22). *W3C and WHATWG finalize split on HTML5 spec, forking 'unlikely'*. Retrieved July 26, 2016, from <http://www.theverge.com/2012/7/22/3175248/html5-fork-w3c-whatwg>
- Bresnahan, T. F., & Yin, P.-L. (2007). *Standard setting in markets: the browser war*. New York: Cambridge University Press.

- Bresnahan, T. (2002). The economics of the Microsoft case.
- Cabral, L., & Salant, D. (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.
- Corts, K. S., & Lederman, M. (2009). Software exclusivity and the scope of indirect network effects in the US home video game market. *international Journal of industrial Organization* , 27, 121-136.
- Efrati, A. (2014, 9 26). *Google's Confidential Android Contracts Show Rising Requirements*. Retrieved 9 27, 2015, from The Information: <https://www.theinformation.com/Google-s-Confidential-Android-Contracts-Show-Rising-Requirements>
- Evans, D. S., & Schmalensee, R. (2002). *Some economic aspects of antitrust analysis in dynamically competitive industries*. in Innovation Policy and the Economy, vol. 2., Lerner, Jaffe and Stern, eds.
- Farrell, J. (2007). Should competition policy favor compatibility? in *Standards and Public Policy*, Grenstein and Stango, eds., 372-389.
- Farrell, J., & Klemperer, P. (2007). Coordination and lock-in: Competition with switching costs and network effects. *Handbook of industrial organization* , 3, 1967-2072.
- Farrell, J., & Saloner, G. (1986). Installed base and compatibility: Innovation, product preannouncements, and predation. *The American economic review* , 940-955.
- Farrell, J., & Weiser, P. (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.
- Farrell, J., Hayes, J., Shapiro, C., & Sullivan, T. (2007). Standard setting, patents, and hold-up. *Antitrust Law Journal* , 74, 603-670.
- Faulhaber, G. (2004). Access and Network Effects in the "New Economy": AOL-Time Warner (2000). *The Antitrust Revolution: Economics, Competition, and Policy* , 453-475.
- Faulhaber, G. (2002). Network effects and merger analysis: instant messaging and the AOL--Time Warner case. *Telecommunications Policy* , 26, 311-333.
- Forking - WHATWG Wiki*. (n.d.). Retrieved July 26, 2016, from <https://wiki.whatwg.org/wiki/Forking>
- Gabel, D. (1994). Competition in a network industry: The telephone industry, 1894--1910. *The Journal of Economic History* , 54, 543-572.
- Garcia, F., & Vergari, C. (2015). Revealing Incentives for Compatibility Provision in Vertically Differentiated Network Industries. *Journal of Economics & Management Strategy* .
- Gilbert, R. J., & Katz, M. L. (2001). An economist's guide to US v. Microsoft. *The Journal of Economic Perspectives* , 15, 25-44.

- Greenstein, S., & Rysman, M. (2007). Coordination Costs and Standard Setting: Lessons from 56K Modems. *Standards and public policy* , 123-159.
- Gross, D. P. (2016). The Ties that Bind: Railroad Gauge Standards and Internal Trade in the 19th Century US. *Working Paper* .
- Kretschmer, T. (2008). Splintering and inertia in network industries. *The Journal of Industrial Economics* , 56, 685-706.
- Lerner, A. V. (2016, July 5). Memorandum on Governance Structures for Multi-sided Platforms.
- Lerner, J., & Tirole, J. (2001). The open source movement: Key research questions. *European economic review* , 45, 819-826.
- Levinson, M. (2006). *The Box: How the Shipping Container Made the World Smaller and The World Economy Bigger*. Princeton University Press, Princeton, NJ.
- Markoff, J. (1992, 6). Coalition Seeks to Adopt a Standard Unix Version. *The New York Times* .
- Miller, B. P., Fredriksen, L., & So, B. (1990). An empirical study of the reliability of UNIX utilities. *Communications of the ACM* , 33, 32-44
- O' Mahoney, S., Cela Diaz S.F. & Mamas, E. (2005). IBM and Eclipse. Harvard Business School Case N9-907-007.
- Postrel, S. R. (1990). Competing networks and proprietary standards: the case of quadraphonic sound. *The Journal of Industrial Economics* , 169-185.
- Puffert, D. J. (2009). *Tracks across continents, paths through history: the economic dynamics of standardization in railway gauge*. University of Chicago Press.
- Rysman, M. (2009). The economics of two-sided markets. *The Journal of Economic Perspectives* , 23, 125-143.
- Saloner, G. (1990). Economic issues in computer interface standardization. *Economics of Innovation and New Technology* , 1, 135-156.
- Salus, P. H. (2015). *Dueling UNIXes and the UNIX Wars. ; login:: the magazine of USENIX \& SAGE* , 40, 66-68.
- Shapiro, C., & Varian, H. R. (1999). *Information rules: a strategic guide to the network economy*. Harvard Business Press.
- Siddall, W. R. (1969). Railroad gauges and spatial interaction. *Geographical Review* , 29-57.
- Stallman, R. (1985). The GNU manifesto.
- United Nations Economic and Social Commission for Asia and the Pacific (UNESCAP). (1996). *Trans-Asian Railway Route Requirements: Development of the Trans-Asian Railway in the Indo-China and ASEAN Subregion*.

United Nations Treaty Collection (UNTC). (2006). *European Agreement on Important International Combined Transport Lines and Related Installations (AGTC)*.

Villa, L. (2013, June 5). *Open Law and Strategy*. Retrieved July 16, 2016, from <http://lu.is/blog/2013/06/05/forking-and-standards-why-the-right-to-fork-can-be-pro-social/>

West, J. (2014). Challenges of Funding Open Innovation Platforms. *New Frontiers in Open Innovation* , 22-49.

West, J., & Wood, D. (2013). Evolving an open ecosystem: The rise and fall of the Symbian platform. *Advances in strategic management* , 30, 27-67.