

Strategic Responses to Standardization: Embrace, Extend or Extinguish?

Jason Woodard
Singapore Management University
jwoodard@smu.edu.sg

Joel West
San José State University
joel.west@sjsu.edu

Jason Woodard and Joel West, “Strategic Responses to Standardization: Embrace, Extend or Extinguish?” in Gino Cattani, Simone Ferriani, Lars Frederiksen, Florian Täube, eds., *Advances in Strategic Management*, Vol. 28, 2011, pp. 263-285, ISBN: [1780521928](#).

Revision date: April 27, 2011

Abstract

Prior research on technology standardization has focused on two common patterns: processes in which product developers and other stakeholders cooperate to achieve a consensus outcome, and “standards wars” in which competing technologies vie for dominance in the market. This study examines Microsoft’s responses to 12 software technologies in the period between 1990 and 2005. Despite the company’s reputed tendency to pursue a strategy dubbed “embrace, extend, extinguish,” a content analysis of news articles from the same period reveals surprising diversity in Microsoft’s responses at the product level.

We classify these responses using a typology that treats “embrace” and “extend” as orthogonal decisions faced by product development organizations. This typology allows four kinds of outcomes to be distinguished, including two kinds of partial compatibility in addition to the familiar cases of full compatibility and incompatibility. To complement this cross-sectional perspective, we also examine the evolution of Microsoft’s strategy with respect to Sun’s Java technology. This longitudinal view highlights another underappreciated aspect of standardization, namely the extent to which a firm’s strategic posture toward a standard can change over time, even within the same product family.

Based on this evidence, we suggest that firms tend to publicly embrace a standard with the aim of gaining legitimacy with a community of adopters, while efforts to extend a standard tend to be motivated by the intent to leverage the underlying technology to achieve or strengthen architectural control. We argue that legitimacy and leverage are strategic complements, making the “embrace and extend” strategy attractive to firms like Microsoft, but that the resulting outcome is unstable. Firms that pursue this strategy ultimately face a choice between contributing their extensions back to the standard and losing proprietary leverage, or giving up the legitimacy associated with standards compliance in exchange for freedom from the constraints of compatibility.

Acknowledgements

We gratefully acknowledge the research assistance of Varun Agarwal, Varun Saraf, Shitij Nigam and Wai Phyo Kyaw, which was made possible by the financial support of the Singapore Management University School of Information Systems.

Oh, our eyes have seen the glory of the coming of the Net,
We are ramping up our market share, objectives will be met.
Soon our browser will be everywhere, you ain't seen nothin' yet,
We embrace and we extend!

— From “The Battle Hymn of the Reorg,” February 1996¹

1. Introduction

¹ A parody of “The Battle Hymn of the Republic” circulated internally by a Microsoft developer (Rebello, 1996). Full lyrics are available at <http://www.muq.org/~cynbe/humor/battle-hymn-of-the-reorg.html>.

Technology standards set the rules that enable technical coordination between firms. Prior research on standardization has focused on how standards promote coordination through market competition, voluntary negotiation and government intervention (David & Greenstein, 1990; Tushman & Rosenkopf, 1992). Firms are typically seen to face a binary choice of either adopting or rejecting a given standard. Although partial compatibility has been considered in the literature on adapters and converters (Katz & Shapiro, 1985; Farrell & Saloner, 1992), most analyses of standardization have assumed that adopting a standard implies at least a short-term commitment to full compatibility.

However, as Microsoft demonstrated during the browser wars of the late 1990s, a proprietary firm can publicly embrace a standard while adding proprietary extensions to its products that undermine compatibility with standard-compliant implementations. Microsoft's use of this practice, sometimes called "embrace, extend, extinguish," was criticized for creating the appearance of compliance while "facilitat[ing] Microsoft's control over standards, in particular network-centric and Internet-related standards" (Klein et al., 1998: para. 400). But proprietary extensions are not in themselves either harmful or uncommon. On the contrary, they have appeared frequently over the last 40 years in fast-moving technological fields such as programming languages and communication protocols, and many (if not most) have been perceived as innocuous. The relatively recent controversy over Microsoft's behavior thus warrants a more systematic examination of how firms respond to standardization efforts.

To better understand this phenomenon, we look more closely at Microsoft's responses to 12 externally created software technologies, viewing each response as a strategic product development

decision. These decisions are complex because they are both enabled and constrained by the firm's existing product portfolio, as well as its external relationships (e.g., strategic alliances, technology licensing agreements, and participation in standards organizations). Moreover, the realization of compatibility depends not only on the strategic choices made by product development managers, but also on detailed implementation decisions made by software architects and programmers. The patterns of partial compatibility we observe are thus contingent on a variety of factors whose interactions have not been thoroughly explored in the product development or strategic management literatures. Our study thus complements existing research on strategic innovation in complex product development projects (Hobday, 1998; Verona & Ravasi, 2003; Grunwald & Kieser, 2007).

We present data from two exploratory datasets covering the period 1990–2005: one based on a content analysis of news coverage related to these technologies, and the second based on a more in-depth analysis of Microsoft's strategy toward Sun's Java technology. This data supports the view that a proprietary firm's choice to embrace a standard is distinct from the choice to extend it. We offer tentative explanations for this diversity of responses, and suggest ways to study their antecedents and consequences in future work.

2. Prior Research on Standards Strategy

Researchers have long classified standards into two categories: those created through voluntary cooperative efforts sanctioned by formal standard-setting organizations (*de jure* standards) and those created by private actors to advance their own interests (*de facto* standards) (David &

Greenstein, 1990). In the former case, conflicting interests are resolved through negotiation or administrative processes, which may be linked to the larger concerns of the affected stakeholders (Besen, 1990; Rosenkopf et al., 2001). In the latter case, firms market competing offerings and decentralized user adoption decisions² determine the final outcome (Katz & Shapiro, 1994; Shapiro & Varian, 1999).

Analysis of standards strategy at the level of the implementing firm is thus usually abstracted down to a binary choice: adopt or don't adopt an existing standard — or, where no standard yet exists, cooperate with others to create one or go it alone (e.g., Katz & Shapiro, 1985; Besen & Farrell, 1994). Cooperatively produced standards without private ownership rights have been considered “open,” in contrast to “proprietary” standards owned by a firm that can extract rents (Gabel, 1987). All things equal, firms that have a strong market position will tend to prefer a go-it-alone proprietary standard, while weaker firms will prefer more open standards (Shapiro & Varian, 1999: 250; West, 2007).

From Binary Adoption to Partial Compatibility

Although the open vs. proprietary typology remains widely used, more recent practice and theorizing has blurred the lines between these two extremes. Examples include rival standard-setting organizations competing for the interests of their respective members, and standards created through formal standardization processes that accrue patent royalties to some of the

² While most of the standards adoption literature examines the decisions by users of a technology to adopt a standard, here we focus on the adoption decisions of firms that produce products that implement a standard.

key participants (Cargill & Bolin, 2007; West, 2007).

Even if the form of cooperation may vary, a general assumption is made in cooperative standard setting that a single standard is produced and all adopters work to create mutually interoperable implementations of it. Little research has been done on partial compatibility, which may arise either intentionally (as an engineering choice by the adopter) or unintentionally (perhaps due to implementation difficulties). Farrell & Saloner (1992) showed how partial compatibility can be profitably provided by making available an adapter or converter. In one of the few empirical studies on this topic, Gandall (1995) examined hedonic prices of spreadsheet applications, in which firms chose to implement full or partial compatibility with the Lotus 1-2-3 file format. More recently, Egyedi & Dahanayake (2003) provided a taxonomy of various causes of unintentional incompatibilities, from sources such as ambiguities in specification and programmer errors.

A common source of incompatibility arises from the decision of a firm to implement more or less than the functionality defined by an externally developed standard.³ In either case, the result is at best one-way compatibility or interoperability. In the case of supersets, the creator of a standard-compliant product implements the standard features and then adds its own proprietary extensions. Such extensions have been widespread for a

³ Here we use “external” to include technologies promulgated as a standard by any organization other than the focal firm, whether through a formal standardization process, the effort of an *ad hoc* consortium, or even the actions of a single competitor.

number of information technologies with active standards communities, including Fortran compilers and libraries in the 1970s and 1980s, dial-up modems during the late 1980s and early 1990s, and web browsers (with respect to their implementation of HTML) in the late 1990s.⁴

In the case of subsets, a firm adopts some of the technology underlying a standard, but does not attempt or claim to be fully compliant. The Open Systems Interconnection (OSI) networking protocols suffered from this problem, as did the Distributed Computing Environment (DCE) initiative. Many firms implemented parts of these standards, and some parts became *de facto* standards themselves (such as OSI's X.500 directory services, which became LDAP, and DCE's Remote Procedure Call mechanism, which was adapted by Microsoft in its DCOM technology).

Despite the prevalence of partial compatibility, prior research has yet to characterize this phenomenon systematically, let alone provide a theory of its antecedents and consequences. This is an important gap, because a decision to be partially incompatible — either by design (e.g., adding or deleting features) or simply by not investing the resources necessary to achieve full interoperability — is one of the key strategic choices made by product development teams in complex system-oriented industries.

⁴ The consequences of incompatible extensions (and incompatibility more broadly) differ by both the product category and the user. For example, incompatibility in driver software between two implementations of Unix might be irreparable for a business user while easily remedied by a programmer. We do not further investigate these distinctions here.

Embrace or Extend: A Typology of Strategic Responses to Standardization

We hypothesize that partial compatibility can be explained in part by a richer view of standards strategy that goes beyond a binary view of adoption. Inspired by Microsoft's "embrace and extend" strategy, we suggest that these two decisions are in fact orthogonal (i.e., embrace *or* extend). The resulting 2x2 typology is shown in Figure 1. Note that the traditional "adopt vs. reject" decision corresponds roughly to the "embrace" axis. If a firm chooses not to extend the standard, this axis covers the well-known cases of full compatibility (Quadrant II) and incompatibility (Quadrant I). On the other hand, if a firm does extend the standard — that is, uses the underlying technology to provide additional functionality, whether or not it adopts the original standard — then we obtain the cases of supersetting (Quadrant III) and subsetting (Quadrant IV). Embracing and extending a standard yields a strict superset: the firm faithfully implements the full standard *and* provides additional non-standardized functionality. Extending a standard that a firm does not embrace (a strategy we call "adapt and replace") typically yields an overlapping subset: the firm borrows some aspects of the standard while eschewing others in favor of its own technology.⁵

⁵ This is distinct from implementing a strict subset of the standard, which is often (though not always) explicitly supported by the standard itself. In the past, strict subsetting often led to compatibility problems between products that implemented different subsets of the standard. However, increases in computing power due to Moore's Law have radically decreased the cost of including rarely-used or optional functionality with minimal performance overhead, and/or making products "smart" enough to negotiate with each other dynamically to achieve

The remainder of the paper examines two specific choices available to firms in their standards strategies: whether or not to embrace (i.e., publicly adopt or reject) an externally developed standard, and whether or not to extend such a standard (i.e., build on the underlying technology and promote these extensions as a new standard). We focus on these research questions:

- Are these separate or interrelated choices;
- If they are related, how are they related; and
- What factors influence these choices?

In this study, we examine these questions from the perspective of a leading technology vendor that has previously promulgated successful proprietary standards. Based on prior research and our own observations, such firms have a wider range of strategic alternatives related to product standards than their less successful rivals, and thus are more likely to exhibit variation in their behavior. In addition, we consider the methodological question of how such choices could be measured reliably in ways that would be feasible for a wide range of firm-standard pairs, and would allow classification of a firm's decision at the earliest point in the product development process. While our exploratory

analysis yields only preliminary answers to these questions, the initial results suggest a promising avenue for further research.

Selecting an Empirical Context

Our research design uses a multi-case analysis, based on archival data that chronicles firm choices over a number of years. To maximize comparability between cases, we sought a single proprietary firm that was involved with (or against) a number of industry standards during the observation period. By doing so, we complement the existing managerial literature on technology strategy, which often uses one or two examples to illustrate a firm's entire standards strategy (cf. Shapiro & Varian, 1999; Gawer & Cusumano, 2002).

In the information and communications technology (ICT) sector, proprietary standards were quite common in the computer industry, particularly during the period 1955–1985.⁶ Among the most visible and successful proprietary computer industry firms were IBM, Digital Equipment, Apple Computer, Microsoft and Intel (cf. Bresnahan & Greenstein, 1999). Of these, DEC, Apple and Intel offered a smaller potential dataset because they had a more limited scope of products and technologies than the other firms, and thus faced fewer opportunities to join (or contest) standardization efforts.

interoperability between different subsets. For example, modern high-definition television standards define many different video formats (720p, 1080i, etc.), not all of which may be displayed directly by a particular HDTV set. Nonetheless, most sets are capable of automatically upgrading or downgrading signals to their “native” resolutions, possibly with in some loss of picture quality but otherwise without adverse impact. Although this is another distinction that may be worth investigating further, in this paper we classify strict subsetting under Quadrant **II** (full compatibility).

⁶ Proprietary standards appear to have been less common in telecommunications firms, perhaps due to interoperability constraints. Even so, proprietary extensions can be found in such nominally open telecommunications standards. For example, one model of test equipment designed for testing SS7 switching networks will (for an additional charge) support proprietary extensions defined by Alcatel, Ericsson, Lucent, Motorola, Nokia or Siemens (Agilent Technologies, 2004).

Both of the remaining firms — IBM and Microsoft — had long histories of creating proprietary standards in competition with industry standards, as well as cooperating with standardization efforts. However, anecdotal evidence suggests that IBM began to shift away from proprietary standards with the 1993 appointment of CEO Louis Gerstner — going so far as to become one of the computer industry’s biggest supporters of open standards and open source software (Ferguson & Morris, 1993; Gerstner, 2002). A study of IBM’s proprietary standards efforts would thus emphasize the period from 1965–1990 rather than more recent events. Meanwhile, many in the computer industry observed that as IBM was moving away from proprietary standards in the early 1990s, Microsoft was assuming its role as a proprietary platform leader in the computer industry.

In choosing between IBM and Microsoft, we thus selected Microsoft for two reasons. First, its proprietary efforts are more recent, and thus reflect more current standardization practice, including strategies developed after the elaboration of economic theories of network effects and switching costs.⁷ Second, most of Microsoft’s technical efforts after 1990 were directly or indirectly related to the emergence of open Internet standards, thus offering a greater degree of comparability among the potential cases.

⁷ In June 1985, Microsoft CEO Bill Gates demonstrated a clear understanding of network effects with his recommendation that Apple license its Macintosh operating system to other firms (Carlton 1997: 40–41).

3. Dataset One: Content Analysis

Sample

We focus on the period between 1990 and 2005, during which Microsoft faced a series of significant technological changes and strategic challenges, including the rise of the Internet. This is also the period in which its conduct came under antitrust scrutiny, and thus (as with earlier IBM and AT&T cases) its internal decisions became more visible due to court-ordered disclosure of internal documents.

We used prior literature and industry histories (especially Auletta, 2001; Bank, 2001) to identify cases involving Microsoft’s responses to a standard (or standards) developed outside the firm, either by an open standardization process (e.g., TCP/IP, HTML) or a rival firm (Java). In some of these cases, Microsoft had (or later developed) its own standard; in other cases, Microsoft did not have (or develop) its own standard and thus the strategy strictly concerned the external standard(s). In addition to differences in Microsoft’s strategy, the cases involving competing standards varied in the outcomes of competition for market adoption.

We identified a total of 12 discrete cases involving 21 external standards and 10 internal standards (Table 1). Research assistants then gathered articles and other information available on the Internet for each case. The data were collected by two first-year undergraduate students majoring in information systems. The students were selected for the project based on their interest in the software industry, but had little or no direct knowledge of the events under study. They were provided with the name of each case, along with a verbal description given by one of the authors. They were instructed to search the Internet for news articles, press releases, and other material related to each case (e.g., technical

documentation, downloadable software), and to compile a web-based document summarizing the material they collected for each case. After gathering the data, each case was summarized by a different student who was not aware of the research objectives.

Of the 12 original cases, one was dropped from further analysis before the data collection phase because the research assistants found the scope of the case to be unclear, and three others were dropped after data collection. One of these was dropped due to a high degree of overlap with another case. Two more were dropped during the coding process because of lack of material. This left eight cases that were used in the content analysis.

Coding

We used content analysis to convert the archival data into explicit measures of firm strategy. Content analysis provides a systematic way of transforming textual data into numerical measures. When appropriate precautions are taken, it provides replicability and other forms of reliability that reduce the potential subjectivity of other forms of qualitative data analysis, and also allows coding a larger volume of data than might be feasible with other methods (Holsti, 1969; Riffe et al., 2005; Fico et al., 2008).

To provide comparability across cases and coders — and isolation from researchers who understood the theoretical objectives — the authors iteratively developed a coding protocol that provided instructions for coding the archival content. As recommended by Riffe et al. (2005), the protocol included a brief discussion of the project objectives, a definition of the constructs, and instructions on how to classify the content. Excerpts of the coding protocol are provided in Table 2. The protocol was provided to two research

assistants who were not involved in the data collection process. They were each asked to classify a subset of the cases, which were (in all but one instance) cases they had previously summarized.

Our interest was in measuring for each document (e.g., news article) evidence of Microsoft's standards strategy on two dimensions. The first was the "embrace" dimension: whether the firm would publicly adopt (or reject) the external standard. The second was the "extend" dimension: whether the firm was using the technology underlying the standard to provide functionality beyond the scope of the standard itself, and promoting these extensions as a new standard. For each dimension, we asked the coder to also measure the strength of support (or opposition). Each article was classified on each dimension using a five-point scale (± 2 , with 0 meaning no evidence could be found in either direction). We asked the coders to mark data as missing if there was contradictory or conflicting evidence on any dimension. Finally, for each article, we asked them to report whether the article indicated an intention ("something that is not yet done but is promised") or a result ("an outcome or consequence").

Results

The results of our content analysis for the eight cases are summarized in Table 3 and plotted on Figure 2. We did find clear and consistent evidence of three of the four actions hypothesized from prior theory.

- *Embrace (adopt)*: Microsoft adopted the external standards SMTP/POP/IMAP.
- *Not embrace (reject)*: Microsoft fought OpenDoc (with OLE) and the combination of CORBA and Enterprise JavaBeans (with COM+).

- *Extend*: Microsoft extended HTML/CSS/JavaScript by creating DHTML, and Java with C#, while its SMB was extended by others into SMB2 and CIFS.⁸

What was missing was evidence of Microsoft not extending (preserving) a standard. In fact, in two cases (OLE and COM+) there was no data at all on extend vs. preserve.

Rather than read a strong message into this result — as Sherlock Holmes did with the dog that didn't bark — we believe it is in the nature of news reporting to assume that public support for a standard also entails preserving that standard (in other words, is a dog-bites-man story rather than a man-bites-dog story). Consistent with that assumption, data was scarce here: evidence relating to extend vs. preserve was reported in only 47% of the case documents.

In three cases, Microsoft did not take a consistent position on embracing a standard:

- *Active Directory*: Kerberos and LDAP were both adopted and rejected, but then Microsoft extended them to create Active Directory.
- *DHTML*: HTML + CSS + JavaScript were both adopted and rejected, but Microsoft then extended them to create DHTML.
- *C#*: Microsoft initially adopted Java, then rejected it and adapted the underlying technology to create a

new programming language called C#. This complex (and prominent) standards strategy is examined in detail in the next section.

4. Dataset Two: Microsoft and Java Case Study

To extend the content analysis, we examined in more detail the three cases involving Microsoft and Java. This approach illustrates the interdependence of multiple standards strategies as part of a larger rivalry, in this case that between Sun and Microsoft. It also demonstrates the dynamism of a firm's choices regarding embrace and extend, and illustrates how firms can move between the quadrants of our typology as a technological field develops. This section draws heavily on news articles and previous accounts in the academic literature, especially Garud et al. (2002), as well as the experience of one of the authors who worked at IBM as a technical evangelist for Java from 1997–99. Our aim is not to present new facts, but to interpret established facts in light of our framework. In the following section, we will build on this interpretation to suggest questions and hypotheses for further research.

Phase 1: A Java Reference Implementation for Windows

After initially declining to adopt Java and focusing instead on its own comparable technology called Blackbird (Quadrant I),⁹ Microsoft made a surprise announcement at its Internet Strategy Workshop in December 1995 that it intended to license Java from Sun. According to Sun's press release confirming the announcement, Microsoft

⁸ The ActiveX case showed evidence of both "embrace" and "extend," which is consistent with Microsoft's response to Java in general (as discussed in the next section). However, Microsoft pointedly declined to adopt JavaBeans. The fact that this was not reflected in the coding highlights the need to be clear about the exact scope of the case, particularly the external standard to which the focal firm is responding.

⁹ Quadrants in the typology of Figure 1 are indicated in bold (e.g., **III** refers to "extend and embrace").

would provide a “reference implementation” of the Java virtual machine and applet API for Windows (II). Sun hailed the announcement as “further proof that Java has become the de facto open standard for programming on the Internet,” and anticipated a harmonious relationship with Microsoft as a Java licensee: “We are happy to be working with Microsoft on a license for the Java technology and look forward to working with them on optimizing the Java technology for Windows” (Sun Microsystems, 1995).

Phase 2: Microsoft Embraces and Extends Java

Potential conflicts were apparent even before Sun’s press release hit the news wires. At the very same Microsoft strategy workshop in 1995, Bill Gates famously announced the company’s intent to “embrace and extend” Internet standards. After describing Microsoft’s approach to spreadsheets (Excel with respect to Lotus 123) and local area networking (Windows NT’s built-in functionality with respect to Novell Netware), Gates (1995) applied the same logic to the Internet: “So [for] the Internet, the competition will be kind of, once again, embrace and extend, and we will embrace all the popular Internet protocols. Anything that a significant number of publishers are using and taking advantage of we will support. We will do some extensions to those things” (III).

These extensions were focused on giving software developers access to the APIs of the Windows platform through the Java language, and providing ways to connect with Microsoft’s own proprietary Internet technologies, such as ActiveX controls (which competed with Java applets but only ran in Microsoft Internet Explorer). When Microsoft announced the signing of a Java license in March 1996, a spokesman said, “We’re building interoperability

between Java applets and ActiveX controls. ... These aren’t separate worlds. They’re totally integrated” (CNET, 1996a). The initial release of Microsoft’s Visual J++ development environment, shipped in October 1996, indeed offered a superset of Sun’s Java functionality: the ability to write cross-platform Java applications, as well as to take advantage of “native” Windows features. However, the J++ product manager acknowledged the potential incompatibility that might result: “In some cases, it will break the cross-platform nature of Java. In some cases, it won’t” (CNET, 1996b).

Phase 3: 100% Pure Java vs. Java for Windows

Alarmed at what it perceived as Microsoft’s attempt to fragment the Java developer community into Windows-centric and cross-platform subsets, Sun launched the “100% Pure Java” marketing campaign in February 1997. Alan Baratz, the head of Sun’s JavaSoft division, criticized Microsoft’s superset strategy and its impact on interoperability: “We encourage all Java licensees to innovate, but to create innovations that are cross-platform. ... Occasionally, there is innovation that is not cross-platform. That’s their business, but that’s playing with fire” (CNET, 1997a). Sun announced a program to certify Java applications as “100% Pure,” and to reward developers of such applications with the use of a special logo as a kind of “Good Housekeeping Seal” of approval.

Over the next year, the paths of Sun and Microsoft continued to diverge. In April 1997, Sun announced an enhanced graphical user interface framework called the Java Foundation Classes (JFC) (Wingfield & Ricciuti, 1997). On the same day, Microsoft announced enhancements to its competing Application Foundation Classes (AFC) framework (CNET, 1997b). Internal documents revealed during the United States

v. Microsoft antitrust trial suggest that that this was part of a coordinated strategy to “Let [the] Java class library space fragment, so that ‘write once, run everywhere’ does not happen (Algaze, 1996). In July 1997, Microsoft explicitly indicated that it would not distribute JFC with its next release of Internet Explorer, choosing to promote AFC instead (IV) (Wingfield, 1997).

When IE 4.0 shipped in September 1997 without JFC and several other Sun-sponsored technologies, Sun responded with a lawsuit (Pelline et al., 1997). Rather than accept Sun’s demand to support what Microsoft considered competing technologies,¹⁰ Microsoft eventually paid Sun \$20 million to settle the lawsuit and terminate its license agreement. Microsoft proceeded to develop C#, a Java-like programming language, as part of its .NET programming framework, abandoning even minimal efforts at cooperation with its Silicon Valley rival (I).

Although Microsoft ended up in nearly the same place it started with respect to Java, the changes in its strategy over two year period illustrate how a firm may be forced to evolve its responses due to unexpected eventualities. In this case, Microsoft appears to have not anticipated the degree to which Sun would fight its efforts to extend Java, as it had done with other rival standards before. In the face of this opposition, Microsoft abandoned its embrace and extend strategy and shifted back to direct rivalry.

¹⁰ The Sun technologies included Remote Method Invocation (RMI), a distributed component technology that was comparable to Microsoft’s DCOM protocol, and the Java Native Interface (JNI), a way to invoke native code that was comparable to Microsoft’s J/Direct mechanism for invoking the Windows APIs from Java.

5. Discussion

Taken together, the two parts of this study highlight the gap between the relatively simple theories of standardization prevalent in the research literature and the complex dynamics of real-world standards strategies pursued by leading high-technology firms. We contribute to bridging that gap by developing a typology that treats “embrace” and “extend” as orthogonal decisions. This typology allows distinguishing four possible outcomes: two kinds of partial compatibility in addition to the familiar cases of full compatibility and incompatibility. More fundamentally, viewing standardization as a multi-dimensional construct reasserts the question of what exactly *is* a standard, as well as the question of why firms differ in their strategic postures along each dimension. In this section, we offer perspectives on both of these questions, along with a brief discussion of the limitations of the study and opportunities for future work.

Standards as Social and Technological Artifacts

Two views of standards are common in the literature. One view sees standards primarily as social artifacts: *conventions* (mutual, though often tacit agreements) to do things a certain way. Another view sees standards as technological artifacts: *designs* that specify how things are to be done. The former emphasizes the ends — coordinated action — while the latter emphasizes the means. These views lead to different notions of what it means to adopt a standard. The most important aspect of adopting a convention is the agreement among the parties; the details of implementing that agreement are often immaterial.¹¹ By

¹¹ A third view can be sometimes found among researchers or practitioners who emphasize *how*

contrast, adopting a design implies arranging technological inputs in a particular pattern, whether or not others use the same design.

As our study of Microsoft demonstrates, these actions are in fact orthogonal: it is possible to espouse coordination while extending a standard in incompatible ways, just as it is possible to implement a standard (fully or more often selectively) without any agreement or claim of compatibility with other adopters. These are by no means pathological situations. On the contrary, we find that cases of deliberate partial compatibility are no less common than the “classic” cases of full compatibility through coordination on a common design, and full incompatibility as a result of a standards war.

Legitimacy and Leverage as Driving Forces

So why do firms choose to respond to standardization efforts (or candidate standards) in different ways? Our examination of Microsoft’s responses to nearly a dozen such efforts over a 15-year period suggests that at least two distinct forces are at work: the desire for *legitimacy* among a community of adopters, and the desire to *leverage* the underlying technology while establishing a new standard under a firm’s own control. Legitimacy is primarily about strategic perception, while leverage is about technical reality. Both can be equally powerful — or in some cases irrelevant, as when a firm has sufficient resources or lead

that agreement is reached. In this view (e.g., Krechmer, 2006), the central issue is process fairness with respect to who gets to participate and how equal are their voices: a standard produced by a formal standards body is more legitimate than the identical outcome realized by a single firm or an invitation-only standards consortium.

time advantage to create its own *de facto* standard.¹²

While there are many reasons a firm might want to be perceived as a “good citizen” by a standards community (or a set of customers who themselves confer legitimacy on such a community), this desire may be especially strong for a firm that is suffering from a credibility gap in a particular technological field. Reasons for such a gap include being late to market, having bet on a “losing” technology, and a lack of trust among customers or complementors. All of these applied to Microsoft in late 1995 when Bill Gates announced the company’s push into Internet technologies. Gates recognized that the company could not go it alone in this segment (e.g., by positioning the Microsoft Network as an alternative to the Internet, as America Online had done until then). Microsoft needed to embrace before it could extend.

Conversely, the main reason to invest technical effort in extending an existing standard — rather than either creating a competing technology from scratch or simply adopting the standard without modification — is because the standard embodies technological knowledge to which the firm does not otherwise have easy access. For example, Microsoft built its Windows file sharing protocols on Server Message Block (SMB) technology developed at IBM, probably for the simple reason that it had been published at the 1984 IBM PC Conference (French, 2007), thus saving Microsoft the trouble of

¹² A recent example of this was the “FairPlay” digital rights management (DRM) algorithm introduced by Apple to copy-protect downloaded songs from its iTunes Music Store in 2003, and officially abandoned in 2009 when music companies no longer demanded such protections as a condition of their agreements with Apple.

implementing it from scratch. Since interoperability with file servers based on SMB implementations other than Microsoft's was not an issue until roughly a decade later, there was little impetus either to launch a standardization effort or publicly commit the company to interoperability with IBM's SMB protocols. The story with DCOM is similar, but in this case Microsoft lacked the experience with enterprise-level distributed component technology that was embodied in the Distributed Computing Environment (DCE) effort. Implementing DCOM based on the DCE/RPC specification was a logical way for Microsoft to promote Windows NT as a viable solution for enterprise customers. Given the low penetration of DCE and high fragmentation of the UNIX market, compatibility was again a secondary issue.

The concepts of legitimacy and leverage thus provide a possible way to explain firms' decisions along each of the axes in our typology. What about the particular attraction of the "embrace and extend" combination? We believe (based on our limited data) that these strategies are complementary, i.e., each is more valuable in the presence of the other. Certainly Microsoft seems to have been tempted to "have its cake and eat it too," as in the Java case where the company valued both the association with a popular Internet technology and the ability to modify the technology to benefit the Windows franchise.

However, our data suggest that this combination of strategies is inherently unstable. Many standards communities tolerate and even encourage some degree of incompatible innovation, especially when extensions are contributed back to the standard (or used only in narrow ways that do not threaten compatibility with the broader base of adopters). This can create a

healthy cycle in which full compatibility is the norm except at the frontier of innovation. On the other hand, when a firm is perceived to be trying to fragment the standard, it quickly loses legitimacy, shifting from "extend and embrace" to "adapt and replace." With no credit to be gained by implementing the external standard in its entirety, there is little reason to continue delivering a superset of the standard functionality, which drives the competing implementations to diverge into overlapping subsets. In the Java case, Microsoft eventually lost the right to use Sun's technology at all, driving the two firms back into direct competition.

Limitations and Future Research

While this exploratory study has considered previously unexamined dimensions of firm product strategies, it has both specific limitations and also the more customary limitations that suggest avenues for future research.

The current paper summarizes an analysis of very preliminary datasets. The limited dataset used for content analysis could be augmented from richer data sources (such as discussions in standards setting organization or standards-related email lists) or expanding the analysis to additional cases.

Additional data may or may not address a key limitation of our current analysis, which is a lack of evidence for the hypothesized "preserve" alternative. From the current data, we cannot conclusively determine whether this is a theoretical finding (firms don't preserve) or merely reflects a methodological challenge (certain types of archival data don't report efforts to preserve standards).

In addition to the limitations of the content analysis, a case study (such as the in-depth Java case) faces certain inherent limitations that require larger scale studies

both to confirm the external validity of the observed outcomes and to offer a more representative explanation for the associated antecedents. In particular, two specific findings would require larger scale confirmation. First, Microsoft's evolving strategy regarding Java suggests that an attempt to both embrace and extend a standard is a unstable combination, perhaps because of the inherent conflict between seeking legitimacy from external stakeholders while maximizing control against direct rivals. Second, more detailed data (such as internal interviews or archival data) is needed to explain why, when and how a firm decides to change its strategic posture towards an external standard.

Finally, we have examined only two sources of partial compatibility — deliberate efforts to create one-way compatibility through intentional supersetting or subsetting of a standard. A broader range of incompatibilities have been observed in practice, so further research is needed to consider the impacts of such strategies, and also whether the causes of incompatibility (e.g., intentional superset vs. lazy compliance) make a difference in the effects.

References

- Agilent Technologies. 2004. Signaling advisor – 37907A: Configuration guide, http://www.home.agilent.com/upload/cm_upload/All/5988-4929EN_ConfigGuide5-1.pdf.
- Algaze, B. 1996. RE: Microsoft Java strategy. Internal memo, Microsoft Corp., October 25, <http://www.justice.gov/atr/cases/exhibits/518.pdf>.
- Auletta, K. 2001. *World War 3.0*. New York: Random House.
- Bank, D. 2001. *Breaking Windows*. New York: Free Press.
- Besen, S. M. 1990. The European Telecommunications Standards Institute: A preliminary analysis. *Telecommunications Policy*, 14(4): 521–530.
- Besen, S. M., & Farrell, J. 1994. Choosing how to compete: Strategies and tactics in standardization. *Journal of Economic Perspectives*, 8(2): 117–131.
- Bresnahan, T. F., & Greenstein, S. 1996. Technical progress and co-invention in computing and in the uses of computers. *Brookings Papers on Economic Activity: Microeconomics*: 1–77.
- Cargill, C., & Bolin, S. 2007. Standardization: A failing paradigm. In S. Greenstein & V. Stango (Eds.), *Standards and public policy*: 296–328. Cambridge: Cambridge University Press.
- Carlton, J. 1997. *Apple: The inside story of intrigue, egomania, and business blunders*. New York: Times Business.
- CNET. 1996a. Microsoft does Java whole hog. March 12, http://news.cnet.com/Microsoft-does-Java-whole-hog/2100-1023_3-207247.html.
- CNET. 1996b. MS debuts Visual J++. October 1, http://news.cnet.com/MS-debuts-Visual-J/2100-1001_3-233769.html.
- CNET. 1997a. MS rivals unified on Java. February 21, http://news.cnet.com/MS-rivals-unified-on-Java/2100-1001_3-272935.html.

- CNET. 1997b. Microsoft wants end to Java wars. April 2, http://news.cnet.com/Microsoft-wants-end-to-Java-wars/2100-1001_3-278516.html.
- David, P., & Greenstein, S. 1990. The economics of compatibility standards: An introduction to recent research. *Economics of Innovation and New Technology*, 1(1): 3–41.
- Egyedi, T. M., & Dahanayake, A. 2003. Difficulties implementing standards. *Proceedings of the 3rd IEEE Conference on Standardization and Innovation in Information Technology* (SIIT 2003): 75–84.
- Farrell, J., & Saloner, G. 1992. Converters, compatibility, and the control of interfaces. *Journal of Industrial Economics*, 40(1): 9–35.
- Ferguson, C. H., & Morris, C. R. 1993. *Computer wars*. New York: Times Books.
- Fico, F. G., Lacy, S., & Riffe, D. 2008. A content analysis guide for media economics scholars. *Journal of Media Economics*, 21(2): 114–130.
- French, S. M. 2007. A new network file system is born: Comparison of SMB2, CIFS and NFS. <http://www.docstoc.com/docs/202716/ols2007-paper-smb2-french>.
- Gabel, H. L. 1987. Open standards in computers: The case of X/OPEN. In H. L. Gabel (Ed.), *Product Standardization and Competitive Strategy*. Amsterdam: North-Holland.
- Gandal, N. 1995. Competing compatibility standards and network externalities in the PC software market. *Review of Economics and Statistics*, 77(4): 599–608.
- Garud, R., Jain, S., & Kumaraswamy, A. 2002. Institutional entrepreneurship in the sponsorship of common technological standards: The case of Sun Microsystems and Java. *Academy of Management Journal*, 45(1): 196–214.
- Gates, B. 1995. Internet strategy workshop keynote. December 7, <http://www.intercom.co.cr/internet/research/1995/1207.htm>.
- Gawer, A., & Cusumano, M. A. 2002. *Platform leadership*. Boston: Harvard Business School Press.
- Gerstner, L. V., Jr. 2002. *Who says elephants can't dance?* New York: HarperBusiness.
- Grunwald, R., & Kieser, A. 2007. Learning to reduce interorganizational learning: An analysis of architectural product innovation in strategic alliances. *Journal of Product Innovation Management*, 24(4): 369–391.
- Hobday, M. 1998. Product complexity, innovation and industrial organisation. *Research Policy*, 26: 689–710.
- Holsti, O. R. 1969. *Content analysis for the social sciences and humanities*. Reading, MA: Addison-Wesley.
- Katz, M. L., & Shapiro, C. 1985. Network externalities, competition, and compatibility. *American Economic Review*, 75(3): 424–440.
- Katz, M. L., & Shapiro, C. 1994. Systems competition and network effects. *Journal of Economic Perspectives*, 8(2): 93–115.
- Klein, J. I. et al. 1998. U.S. v. Microsoft: Proposed findings of fact. *US Department of Justice*. <http://www.usdoj.gov/atr/cases/f2600/2613.htm>.

- Krechmer, K. 2006. Open standards requirements. *International Journal of IT Standards and Standardization Research*, 4(1): 43–61.
- Pelline, J., Lash, A. & Kornblum, J. 1997. Sun suit says IE failed Java test. October 7, http://news.cnet.com/Sun-suit-says-IE-failed-Java-test/2100-1001_3-203989.html.
- Rebello, K. 1996. Inside Microsoft. *Business Week*, July 15: 56.
- Riffe, D., Lacy, S., & Fico, F. G. 2005. *Analyzing media messages: Using quantitative content analysis in research* (2nd ed.). Mahwah, NJ: Lawrence Erlbaum.
- Rosenkopf, L., Metiu, A., & George, V. P. 2001. From the bottom up? Technical committee activity and alliance formation. *Administrative Science Quarterly*, 46(4): 748–772.
- Shapiro, C., & Varian, H. R. 1999. *Information rules*. Boston: Harvard Business School Press.
- Sun Microsystems. 1995. Microsoft signs letter of intent to license Java technology. Press release, December 7.
- Tushman, M. L., & Rosenkopf, L. 1992. Organizational determinants of technological change: Toward a sociology of technological evolution. In L. L. Cummings & B. M. Staw (Eds.), *Research in organizational behavior*: 311–347. Greenwich, CT: JAI Press.
- Verona, G., & Ravasi, D. 2003. Unbundling dynamic capabilities: An exploratory study of continuous product innovation. *Industrial and Corporate Change*, 12(3): 577–606.
- Weiss, M., & Cargill, C. 1992. Consortia in the standards development process. *Journal of the American Society for Information Science*, 43(8): 559–565.
- West, J. 2007. The economic realities of open standards: Black, white and many shades of gray. In S. Greenstein & V. Stango (Eds.), *Standards and public policy*: 87–122. Cambridge: Cambridge University Press.
- Wingfield, N. 1997. Sun posts Java foundation class. July 8, http://news.cnet.com/Sun-posts-Java-foundation-class/2100-1001_3-201250.html.
- Wingfield, N. & Ricciuti, M. 1997. Java to get standardized toolkit. April 2, http://news.cnet.com/Java-to-get-standardized-toolkit/2100-1001_3-278493.html.

Figures and Tables

Figure 1: Typology of strategic responses to standardization

		<i>Embrace</i>	
		No	Yes
Extend	Yes	IV <i>Adapt & replace (overlapping subset)</i>	III <i>Extend & embrace (strict superset)</i>
	No	I <i>Fight the standard (incompatible)</i>	II <i>Adopt the standard (fully compatible)</i>

Table 1: List of cases in study

#	External Standard	Microsoft Standard	Outcome
1.	JavaBeans	ActiveX	MS adopted own standard; both survived
2.	CIFS, SMB2	SMB	MS adapted external technology, which became <i>de facto</i> standard
3.	Kerberos, LDAP	Active Directory	MS extended external standards; both survived
4.	OpenDoc	OLE	MS adopted own standard; external one failed
5.	SMTP, POP, IMAP		MS adopted external standard
6.	CORBA, EJB	COM+	MS adopted own standard; both survived
7.	Java	C#	MS shifted from external to own standard; both survived
8.	HTML, CSS, JavaScript	DHTML	MS extended external standards; some extensions became standardized, others were dropped
9.†	TCP/IP		MS adopted external standard
10.†	HTTP, SSL		MS adopted external standard
11.†	DCE/RPC	DCOM	MS adopted own standard based on external technology; both survived but MS dominated
12.†	Java, JNI	J/Direct, Visual J++	MS extended external standards but substituted some of its own technologies; external ones survived

† Dropped from content analysis (see text).

Table 2: Excerpts from coding protocol

Introduction

This study of news stories is intended to chronicle the choices made by IT firms related to technologies that are competing for acceptance as standards. These technologies may be defined by formal standards organizations (*de jure* standards), industry consortia, or other firms (*de facto* standards). The origin of these technologies is not a concern of the study, nor is the question of whether they succeed or fail as standards; we are simply interested in how firms respond to them.

For convenience sake, we are currently limiting our focus to a single firm (Microsoft) and its reaction to a succession of IT standards during the period 1990–2005. We would like to develop a chronological picture of the firm’s efforts to support, oppose or modify a given standard or family of standards, as reflected in a series of events as reported by news articles.

...

Article Coding

Concepts

We are interested in two possible dimensions of a firm’s response to the emergence of a new technology standard:

- **Adopt or reject.** Adoption means that the firm (Microsoft) publicly accepts and implements the standard in a way that provides interoperability with other compliant implementations. Rejection means publicly declining to adopt the standard, and choosing instead to develop products or technologies that cannot fully interoperate with compliant implementations.
- **Extend or preserve.** Extending a standard means that the firm uses the technology underlying the standard to provide functionality beyond the scope of the standard itself, and promotes these extensions as a new standard. The extended functionality need not be compatible with the existing standard (i.e., using it may result in a loss of interoperability with compliant implementations). Preserving a standard means the firm chooses not to provide additional functionality or does not promote its extensions as a new standard.

...

Coding Rules

Within each article, we are coding data on each dimension on a scale of +2 to –2. By convention, we assign positive values to “adopt” and “extend,” and negative values to “reject” and “preserve.”

Code	Position	Description
+2	Definitely adopt / extend	Strong and direct evidence in support of this response
+1	Possibly adopt / extend	Indirect, suggestive or other weak evidence in support of this response
0	No evidence	No evidence either way
-1	Possibly reject / preserve	Indirect, suggestive or other weak evidence against this response
-2	Definitely reject / preserve	Strong and direct evidence against this response
?	Mixed or contradictory evidence	A combination of positive and negative evidence

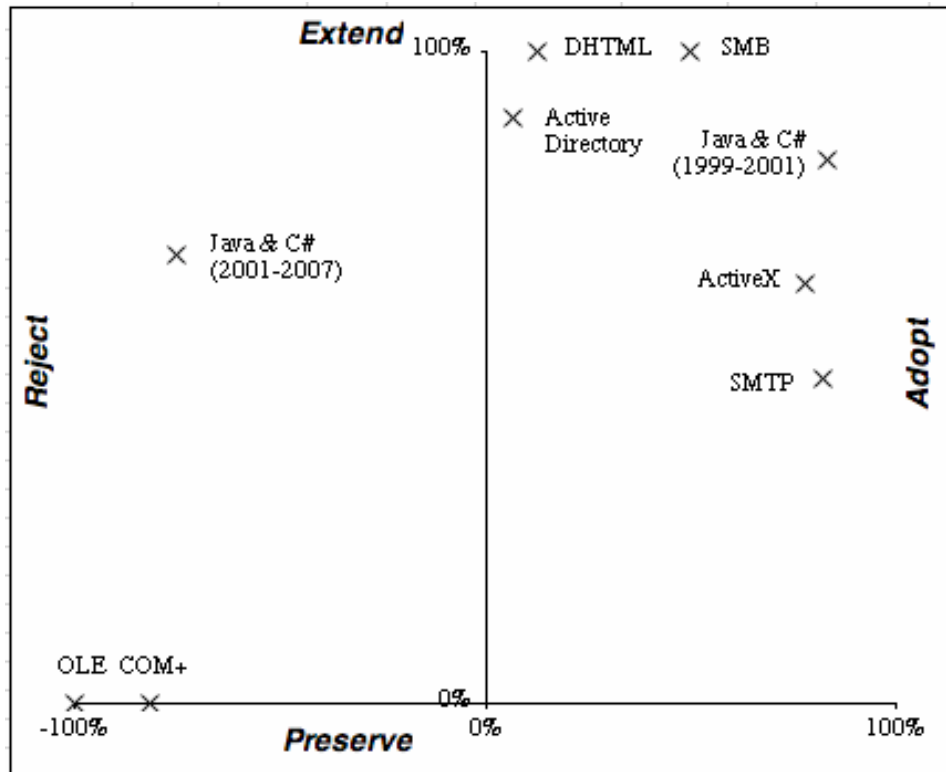
Table 3: Results from content analysis of 8 cases

	Standards	No. of Articles	Adopt Avg.†	Extend Avg.†	Remarks
1.	JavaBeans versus ActiveX	16	1.56	1.29	
2.	SMB into CIFS, SMB2	3	1.00	2.00	
3.	Kerberos + LDAP → Active Directory	16	0.14	1.80	Evidence of both adopt and reject
4.	OLE versus OpenDoc	1	-2.00	n/r	
5.	SMTP, POP, IMAP	11	1.64	1.00	
6.	COM+ vs. CORBA, EJB	8	-1.63	n/r	
7.	Java into C#	12	1.67	1.67	Shift from strong adopt to strong reject
8.	HTML + CSS + JavaScript → DHTML	9	-1.50	1.38	Evidence of both adopt and reject

† Where +2 is strongly adopt or extend, and -2 is strongly reject or preserve; omits articles for which no data is available.

n/r: No measures were reported among the surveyed articles.

Figure 2: Normalized plot of content scores from 8 cases



Source: Data of Table 3, normalized from ± 2 to $\pm 100\%$.

Figure 3: Application of the framework to the Java / C# case

