

Andras Niedermayer
2003

MSc Economics Extended Essay

EC427 Economics of Industry

**Java vs. Microsoft:
Compatibility, Innovation and Anti-Trust Law**

Introduction

In 1995 Sun Microsystems and Netscape made an announcement which in their view would cause a revolution in the world of personal computers: the integration of Sun's Java Virtual Machine in Netscape's Navigator browser would give people the opportunity to run software stored anywhere on the Internet on their computers, independently of the operating system of the user or the server storing it. The idea was "write once, run anywhere." Traditional concepts like operating systems and installed applications dominated by Microsoft would disappear or become irrelevant: users would download the pieces of software they just need, and pay only for what they use.

Today, we are far away from the predictions made. Operating systems and installed applications still play a role and now even most browsers used are from Microsoft! What has happened in the last eight years?

Sun and Netscape (part of AOL in the meantime) claim that Microsoft's anticompetitive behavior destroyed their innovative idea. Microsoft has predated the market for browsers and has introduced a non-compatible version of the Java Virtual Machine. Microsoft claims that it was the low quality of the products distributed by Sun and Netscape which stopped their success. Netscape's browser was not appealing enough for consumers and Java is too slow. Originally it had also claimed that the idea of "write once, run anywhere" and of network programs was without future. However, after the launch of Microsoft .NET in 2001, a platform aimed at running the same software on any device and distributing software over the Internet, this claim would not be credible anymore. (To be fair, this need not be a contradiction by Microsoft: according to the Redmond software company, Java is a bad solution for the same problem that they want to solve well with .NET.)

This essay is about innovation and compatibility in the software industry in general and the case Java vs. Microsoft in specific. The question which side is right or wrong in its statements will not be answered as there are diverging opinions even among experienced computer scientists.¹

After the description of the market in which Java and Microsoft are competing and a summary of the anti-trust case against Microsoft, it will be shown, how different existing economic models relate to the developments.

Afterwards, a simulation based on a model created for this essay will be used to explain some of the development in markets, where compatibility and innovation play a significant role. The code performing the simulation is included in the appendix.

Description of the Market

It is difficult to define clearly the market in which Sun Microsystem's Java competes with Microsoft's products. Actually, it is even difficult to classify unmistakably what we mean when talking about Java. Sun itself mentions Java usually in expressions like "Java Technology" and "Java Platform", all different aspects of Java are meant. Java is not only a programming language, but also a Virtual Machine, a substitute for an operating system, a class library for programmers, a model for software components in general and distributed components in specific. The products in these categories can be both complements and

¹ Software developers locked into Java technology usually consider Java to be technically superior, while software developers locked in the Microsoft platform believe the opposite. Actually, this difference of opinions can be explained by a theory of Farrel and Saloner (1985) presented in the chapter "Application of Economic Theory".

substitutes to each other.² There are three editions of Java: the Standard Edition (SE) for desktop applications, the Enterprise Edition (EE) containing the Standard Edition and products for server applications and the Micro Edition (ME) for Personal Digital Assistants, mobile phones, etc. The statement made in the introduction that Java did not have the success Sun had hoped for is actually only true for the Standard Edition, the Enterprise Edition and the Micro Edition are dominant in there areas. The different Java products available and their respective competitors are listed in Table 1 in the Appendix. The different aspects and components can also be shown as in Figure 1.

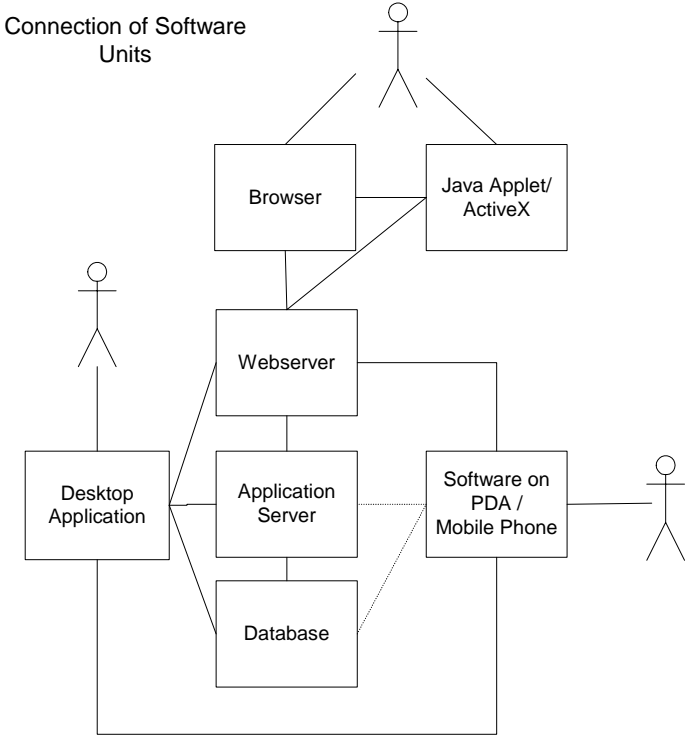


Figure 1

Compatibility problems often arise at the connections. E.g. when a company introduces an electronic flight booking service, customers should be able to book over the Internet using their browser or via their mobile phone. The data they enter is processed by the application server (which matches data available on free seats in planes with booking requests) and stored in a database. Employees of the company may have access to the system through a desktop application installed on their PCs giving them more possibilities to interfere than customers. Because the different units involved have different ways of handling data and requesting actions to be performed, it has been traditionally tedious for software developer to ensure that the parts interoperate correctly. There are estimates that ca. 60% of the software development time is spent with making different units work together. A unified software platform tries to reduce these interconnection cost, the advocates of the software platforms sometimes claim, that these costs fall away. Considering the fact that billions of dollars are spent worldwide on software development, this can make a huge difference.

² The Java Virtual Machine e.g. is in the short run a complement, on the long run possibly a substitute to the operating system.

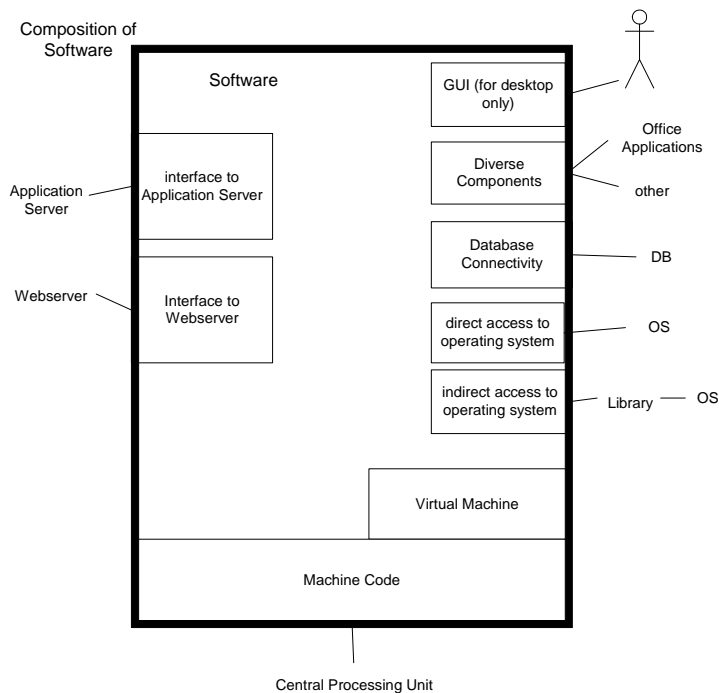


Figure 2

One can see from Table 1 that Sun's Java and Microsoft's .NET initiative are the only technologies that contain virtually every aspect mentioned. The two companies are not competing for single product markets but for platforms. Programmers and end users benefit from choosing a platform which covers all (or most aspects) of software development and usage and thus ensures compatibility and integration.

One has to note that not only Sun is providing the Java platform, but also IBM, BEA Weblogic, GNU, the Apache Software Foundation and other third party vendors. So Java is less a platform owned by Sun, but more a standard guaranteeing the compatibility of different products

The innovation of the Java and the .NET platforms are not the single products contained in them³, but the combination of all parts into one single platform and a company pushing the platform and supporting its development and its compatibility. Therefore, both Java and Microsoft's .NET compete in the same market for software platforms. Java is more targeted at a single-language multiplatform environment, where network connections are in the center; .Net is targeted at multi-language single-platform (MS Windows) environments where access to the operating system and to office documents is central.⁴

Because of the recent introduction of .NET most of the anti-trust investigations and economic analyses made were done before its introduction. For this reason, much of the considerations expressed here will focus on Microsoft's platform in general (operating system, office applications, etc.) before the launch of .NET.

³ E.g. the idea of a virtual machine was already introduced with Smalltalk in the beginning of the eighties. The interaction of software running on different computers existed already on UNIX in the seventies with remote pipes. Object oriented software exchanging data came up at the end of the eighties with CORBA and later with Microsoft's Distributed Component Object Model (DCOM). The concept of software being written in several languages has been used in the development of many software libraries

⁴ This targeting is not exclusive. There is multilanguage support for Java and it is possible to develop applications with access to specific features of an operating system and of office documents. There is also an open source attempt (Mono) to port a part of .Net (MS IL) to other operating systems. It is *possible* to use either technology for the environment it is specific for, but (so far) it has been considered inconvenient.

The Anti-Trust Case against Microsoft

What happened

In 1990 the Federal Trade Commission started an anti-trust investigation of Microsoft. However, without result; the investigation ended in 1993 when the FTC decided not to file a lawsuit under the Sherman Act.

In 1994 the Department of Justice filed a complaint against Microsoft claiming that Microsoft was engaged in anti-competitive behavior through its contracts with Original Equipment Manufacturers (OEMs). The case was abandoned after Microsoft and the Department of Justice issued a consent decree with the following content:

1. Microsoft had to end its “pre-processor” contracts
2. Microsoft was not allowed to make license agreements on the licensing of other products. Only contractual bundling but not technical bundling was forbidden.

In October 1997 the DOJ alleged that Microsoft violated the 1995 consent decree by bundling Internet Explorer with Windows. In December 1997 Judge Thomas Penfield Jackson granted an injunction barring the bundling of Internet Explorer and Windows; this injunction was, however, voided in May 1998 by the Court of Appeals. One month later the Court of Appeals ruled that the 1995 consent decree was not applicable for the bundling of Internet Explorer and Windows 98 because it was considered a case of technical bundling. At the same time the DOJ brought an antitrust lawsuit against Microsoft, joined by the Attorneys General of 20 States and the District of Columbia. Microsoft was alleged of having monopolized the market of personal computer operating systems and browsers through exclusionary contracts and predatory conduct. In November Judge Jackson declared his findings of facts. Settlement talks between the involved parties followed; when they broke off in April 2000 Judge Jackson issued his “conclusions of law”: Microsoft was declared guilty of monopolization of the software market. In June Judge Jackson decided for a break-up of Microsoft. In June 2001 the Appeals Court adjudicated to reverse most of the rulings of Judge Jackson (including the break-up). In short the Court ruled:

1. “Microsoft’s breakup and other remedies imposed by the District Court are vacated
2. Microsoft is found guilty of monopolization of the operating systems market for PCs
3. Microsoft is found not guilty of bundling of Internet Explorer with Windows
4. Microsoft is found not guilty of attempting to monopolize the Internet browser market
5. The district court judge Thomas Penfield Jackson is taken out of the case for improper behavior
6. The case is remanded to the District Court for remedies determination for the monopolization charge
7. The Appeals Court instructs the District Court to examine the bundling of IE and Windows (if plaintiffs bring it up) under ‘a rule of reason’ where the consumer benefits of bundling are balanced against the damage of anti-competitive actions” (Nicholas Economides’ website)⁵

On November 2002 Judge Colleen Kollar-Kotelly of the District Court of Columbia accepted with minor changes the agreement between Microsoft, the USDOJ and the Attorneys General of nine states.

On April 21, 2003, as part of the settlement and anti-trust case with the DOJ and State Attorney Generals, Microsoft gave third party vendors the possibility to license elements of the Windows client/server protocols, so that they could use them to ensure interoperability of their products with Windows clients⁶.

⁵ http://www.stern.nyu.edu/networks/Analysis_Appeals_Court_Decision.htm

⁶ Information taken from Microsoft’s website: www.microsoft.com

Economic Evidence

Gilbert & Katz (2001) note that both “Microsoft and the government considered Internet browsers combined with the Java language and webserver applications to be particularly potent sources of potential competition for Windows.” Java was a middleware layer between the operating system and applications. Bill Gates feared that Netscape threatened to “commoditize the underlying operating system.” (Gates, May 26, 1995)

The government claimed that Microsoft tried to stop this by eliminating Netscape and undermining Java.

The government considered as relevant market the market for operating systems for Intel compatible computers where Microsoft had a market share of over 90%. A direct entry was unlikely, as a failed attempt by IBM to introduce OS/2 had shown. IBM had invested \$1bn into their alternative to Windows, without any success. The failure was due to the “applications barrier to entry”: users prefer an operating systems with many applications available, software developers develop applications for operating systems which are widespread among users. Professor Richard Schmalensee, Microsoft’s chief economic expert at trial, claimed that platforms and not operating systems should be considered as the relevant market. He showed that Microsoft did not have a monopoly by calculating the price a monopolist would charge given the elasticities of demand present. This price was 16 times higher than the actual price. Microsoft could not overcharge their customers because they feared that a competitor could take over the market rapidly. According to Schmalensee, there was “competition *for* the market, rather than competition *within* the market.”

The DOJ and Microsoft agreed that catastrophic entry could have jeopardized Microsoft’s market position. In the DOJ’s view Microsoft used “exclusionary and predatory tactics to block catastrophic entry”, whereas Microsoft claimed having competed in a fair way.

According to Professor Franklin Fisher (the government’s expert witness) and Professor Daniel Rubinfeld (the DOJ’s chief economist) the low prices did not contradict the thesis of Microsoft having a monopoly: Microsoft had incentives to lower prices in order to increase the installed base of its product and because of problems with software piracy. Microsoft’s high market value was evidence that market participants believed in Microsoft being able to raise prices sharply in the future.

According to the DOJ Microsoft prevented the fall of the applications barrier to entry by exclusionary and predatory behavior.

The exclusionary behavior consisted of:

- Contracts with Internet Service Providers: possibilities to simplify their networks was included in the operating system if they made their customers use Internet Explorer
- Contracts with OEMs to include IE in the configuration delivered to customers
- Contracts with internet content providers: links to their sites were included in Internet Explorer’s channel bar in exchange for making their pages “optimized”⁷ for IE

Exclusionary contracts can be either socially desirable or undesirable. An analysis of the details led the court to the conclusion that only the first two kinds of contracts were harmful.

Microsoft’s predatory conduct consisted of giving away Internet Explorer for free to exclude Netscape and establish an installed base.

Microsoft’s defense argued with the “one-monopoly-rent” argument: a monopolist does not have an interest in predation against complementary products. A counter argument to this was that browsers were only in the short-run complements of the operating system, on the long run they could have become substitutes.

The government’s witness, Paul Maritz – a senior Microsoft executive – said that Microsoft wanted “to cut off Netscape’s air supply, ... by giving away free browsers, Microsoft was going to keep Netscape from getting off the ground.”

⁷ An expression often used with the meaning of “inconvenient to users with other browsers.”

On the issue of bundling the government claimed that it was close to contractual bundling and therefore illegal. Microsoft's defense pointed out "that other operating system vendors, such as BeOS, Linux, and Sun, bundled browsers with their operating systems". (Gilbert and Katz (2001)) The court also decided that Microsoft was not guilty of illegal bundling..

The most important question in anti-trust lawsuits is whether Microsoft harmed the consumers. Consumers clearly benefited from lower browser prices on the short-run, but on the long run they would lose because of higher prices and less innovation.

A difficulty arising in the assessment of potential effects on consumers is the fact that one cannot predict whether the combination of Netscape/Java would have been successful in replacing the dominance of the operating system. Gilbert & Katz note that it is enough that there is evidence that Microsoft *believed* in the possible success of Netscape/Java.

Application of Economic Theory

Many of the developments in the competition between software platforms can be explained by economic theory quite well.

Companies developing or buying software based on a platform make a specific investment; it is costly to switch to another platform at a later point in time.

Williamson's (1975) theory of long-run relationships states that a specific investment by the buyer or seller decreases bargaining power ex post. This gives incentives for mergers or long term contracts.

Williamson (1975) and Coase (1937) state that the transaction costs resulting from incomplete long-term contracts can be overcome either by arbitration (internal or external) or by authority (one side is given the bargaining power ex post).

One can observe all of these solutions in the software market. Microsoft has bought several software companies. E.g. Visio (software for the creation of diagrams) was developed by an independent software vendor, until a tighter integration with MS Office was required and Visio Corp. was bought by Microsoft.

When Sun wanted to complete its platform with an office product in order to compete with Microsoft, it bought Star Division and started distributing their product Star Office for free.

Interesting cases of external arbitration is the situation of Sun binding its hands by handing over decision rights over the development of the Java platform to the Java Community Process⁸ which is supervised by PricewaterhouseCoopers. Originally Sun made contracts with the licensees of the Java platform, but it proved to be costly and ineffective to enforce the contracts: Sun accused Microsoft of deliberately making its MS Java Virtual Machine incompatible with the Java standard; Microsoft on the other hand accused Sun of changing its compatibility testsuits in order to make Microsoft's Virtual Machine seem incompatible with the standard. The case was brought to court and it took several years to settle the dispute. Microsoft and Sun agreed that Microsoft was not allowed to use the trademark Java and that it had to cease support for the MS Java VM until 2004 and eliminate it from all of its products by 2008. This dispute incurred high costs to both companies and to the software developer community.

Microsoft faced with its .NET platform a similar problem as Sun with its Java platform: third party software houses feared that Microsoft would alter the specifications of its platform in order to be ahead of other companies after they had already committed themselves to the platform. To overcome these concerns, Microsoft submitted the specifications of a part of its platform, the MS Intermediate Language, to the standards committee ECMA. Participants of

⁸ www.jcp.org

the open source project Mono⁹ and the company Ximian supporting the project have used these specifications to port some parts of the .Net architecture to Linux. Interestingly, if a full-scale port of .Net became available for other operating systems than Windows, Microsoft could lose its operating system monopoly. For this reasons, some observers are skeptical whether Microsoft has the incentives to allow a successful full scale port of its .NET platform without interference and whether the restriction posed by the standards committee will suffice to ensure compatibility. One could argue that Microsoft would lose to a certain extent its operating system monopoly, but in exchange gain a better market position (maybe even monopoly) in the platform market.

Farrel and Saloner (1985) consider in their model the role of standardization, compatibility and innovation in a market with network externalities. Standards have the benefit of ensuring the compatibility of products of different vendors and thus enabling consumers to benefit from larger network externalities. But they might have the cost of causing excess inertia: companies may stick to a standard even if a better innovative product is available. The authors show that with heterogeneous firms and incomplete information, there may be “bandwagon effects”: initially only firms strongly benefiting from the new technology join. When the bandwagon starts rolling, others join as well, even if they are not in favor of the new technology per se.

One can see this pattern in the software platform market. When Java was launched in 1995 many companies were cautious about committing themselves to the technology. The first companies to integrate Java in their products were those benefiting the most from an OS independent programming language with enhanced network capabilities. One of the first adopters, Netscape, had two reasons to favor Java: its browser profited from a network oriented language and the operating system independence helped its attempt to introduce a middleware layer between applications and (mainly Microsoft’s) operating system, and thus “steal” the market for operating systems. Other companies got the bandwagon as it started rolling. Even Microsoft, presumably being harmed by this new platform, supported Java initially. It developed the MS Java VM¹⁰ and intended to implement its COM components in Java in the beginning.

Something similar seems to happen with .Net today. First movers have taken a risk by committing early to the platform. These are typically software houses specialized in products for Microsoft Windows whose software interoperates with MS Office. It is yet too early to tell, but others will probably join the bandwagon and in the end even companies not in favour of the .Net platform will have to switch in order not to forego benefits from network externalities.

This model may also give an explanation why software developers using a specific platform have an interest to convince others that “their” platform is superior (the situation mentioned in the Chapter Introduction): if others join the platform one prefers to use, one can enjoy benefits from network externalities. If a bandwagon starts rolling to another technology, one has the choice of two evils: either getting stranded in a technology no one else uses or using a technology one does not favor (because one solves specific problems for which the other technology is better suited or because one has sunk costs in learning to use the other technology.)

Katz & Shapiro (1985) write about the interaction of network externalities and compatibility choices made by firms. In their model consumers first have expectations about network sizes of firms. Then firms choose the produced quantities based on consumers expectations (their products may or may not be compatible). The quantities determine the prices, which in turn influence the purchase decisions. The authors look for Fulfilled Expectations Cournot

⁹ go-mono.com

¹⁰ Originally being thought of being fully compatible with Sun’s standard.

Equilibria (FECE): the expected network sizes are equal to actual network sizes. The justification for considering only FECEs is that consumers cannot be fooled over and over.¹¹ The authors find equilibria with complete incompatibility between all products and with complete and partial compatibility.

Farrel & Saloner (1986) look at the effect of an installed base on the innovation of products. Companies deciding for a product incur sunk costs are less willing to spend money on further sunk costs and switching costs. If a new superior technology is developed, firms with an installed base are reluctant to accept it; new entrants may prefer to choose the most wide-spread product. This gives an incumbent monopolist the possibility of predatory pricing: he reduces prices in order to keep a potential entrant with a superior technology out of the market. After some time the installed base of the product is large enough to sustain the monopoly.

The choice of commitment to a product also depends on expectations about future products. A company anticipating the introduction of a new product may hold back its investment in the durable good until the new product is launched. This gives firms the possibility of anticompetitive predatory preannouncement.

Installed base does have a large effect on the software platform market. It has been estimated that the language with most lines of source code productively used in commercial applications is COBOL – a language developed in the seventies. Both Sun and Microsoft preannounced their products before their wide-scale launch in 1995 and 2001, respectively.

Farrel & Saloner conjecture that a sudden temporary increase in the arrival of new users increases the probability of a technological change.

This seems to be true for the development around 1995. With the launch of the Internet many new users joined the market for PCs, this gave Sun and Netscape the chance to try to take over the market for platforms dominated by Microsoft. It remains open whether they failed to do so because of not supplying products with sufficient quality or because of Microsoft's aggressive intervention.

Arthur (1989) explores “the dynamics of allocation under increasing returns in a context where increasing returns arise naturally: agents choosing between technologies competing for adoption.” Two technologies exist; new agents enter the market one by one and choose which technology they want to adopt. Agents prefer choosing a technology already tested by others, because it is better developed through learning by doing. The outcome of the system can be determined by small historical events: if one technology is more widespread in the beginning by chance, it will be even more widespread later on with a high probability. Eventually, the system might lock in an inferior technology.

Shy (1996) introduces a discrete time overlapping generations (OLG) model where in each period t , $t=1,2,\dots$, “the consumer population of the economy consists of η_t young consumers and η_{t-1} old consumers” (Shy (2001)). In each period a new technology T_t is invented. Young consumers can adopt the new technology, in which case the currently used technology V_t becomes T_t , or they can use the technology used in the previous period V_{t-1} . Their utility is $u(T_t, \eta_t)$ if adopting the new and $u(V_{t-1}, \eta_{t-1} + \eta_t)$ when using the old technology. A technological revolution takes place if $u(T_t, \eta_t) \geq u(V_{t-1}, \eta_{t-1} + \eta_t)$. Given the speed of innovation (development of T_t) and the utility function of the consumers, the duration of a technologies and the frequency of technological revolutions can be computed.

¹¹ One might argue that uncertainty can lead to expectations not being fulfilled or that there are unique historical events (such as the decision between VHS and Beta Max or between MS Windows and OS/2 and others) which cannot be repeated or reversed. but this approach clearly has the advantage of delivering strong statements through analytical reasoning.

Simulation of Dynamic Competition with Compatibility and Innovation

There seems to be a complicated interaction between compatibility and innovation. Trying to ensure compatibility with past products and with products of competitors might hinder innovation; a revolutionary innovation can make compatibility impossible. When Microsoft was accused of deliberately making its products incompatible, Bill Gates replied with the initiative “Freedom to Innovate.” He was thus implying a trade-off between compatibility and innovation; according to his view, Microsoft simply made the choice for innovation. However, there can also be cases where compatibility and innovation go hand in hand: a company might increase (or reduce) the compatibility and the quality of its product at the same time. A case of an increase in both was e.g. when Microsoft launched its Internet Explorer, increased the quality of its browser at a rapid pace and complied well (better than Netscape at that time) with the standards set by the World Wide Web Consortium (W3C, www.w3c.org). Another case was in the beginning of the eighties when Oracle Corp. brought its database system on the market, which was highly innovative and also followed the theory of relational databases developed in the beginning of the seventies. Today there are many innovations by several software companies in connection with the open standard XML defined by the W3C. Both Java and .Net have newly introduced features for the handling of XML. Microsoft’s communication protocol XML-RPC is also based on this standard.

On the other hand, a company may choose a measure which decreases both compatibility and innovativeness (I use the term equivalently with quality). The first versions of MS Windows were compatible with both MS DOS and DR DOS. Soon after Windows’ introduction, Microsoft released a version with a feature disabling the interoperability with DR DOS, this was one of the reasons the FTC started its anti-trust investigation in 1990.

There are also certain patterns to be explained: companies tend to comply well with standards when they have a low market share and to reduce their compatibility as soon as their market share becomes large enough. E.g. in the eighties Apple was market leader in the PC market, it did not allow for compatibility with third party vendor devices: a Macintosh user had to buy an Apple mouse and an Apple printer and mostly also application software from the same company. MS DOS, in contrast, interoperated with peripheral devices from different companies and software from different vendors was available (e.g. Lotus-1-2-3, WordPerfect, etc.). After Microsoft gained large market shares, it reduced Windows’ openness to other companies, a part of the anti-trust case was about Microsoft not disclosing all of its Application Programming Interface (APIs) to other companies. When Netscape dominated the browser market, it was making its browser non-compatible with the W3C specification, Microsoft (as mentioned above) introduced with its Internet Explorer a browser much closer to the standards. After Internet Explorer gained significant market share, the compatibility decreased again. Also products of companies started diverging after they had acquired enough market power (e.g. Oracle with its Database).

I numerically simulate the implications of a model derived from Shy’s (2001) model of product innovation, Arthur’s model of path dependency and Katz & Shapiro’s (1985) model of compatibility.

Consumers in this model benefit from both network externalities arising from compatible products and from the innovativeness (=quality) of the product. The utility a consumer derives from using technology T_p in period t is:

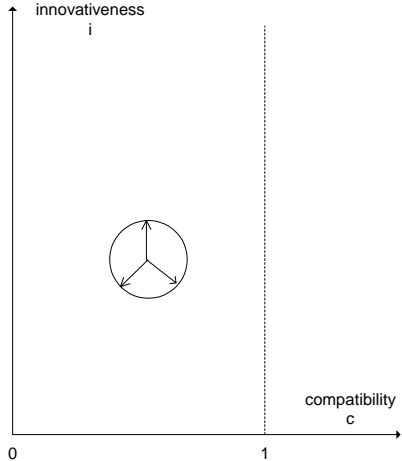
$$u_t^h(T_p) = \eta_t^p + c_t^p \sum_{q \neq p} c_t^q \eta_t^q + i_t^p + \varepsilon_t^{p,h}$$

Where η_t^p denotes the number of users of T_p in period t , c_t^p denotes the compatibility level of product p at period t , i_t^p the innovativeness of product p at t and $\varepsilon_t^{p,h}$ a random noise with $\varepsilon_t^{p,h} \sim N(0, \sigma^2)$.

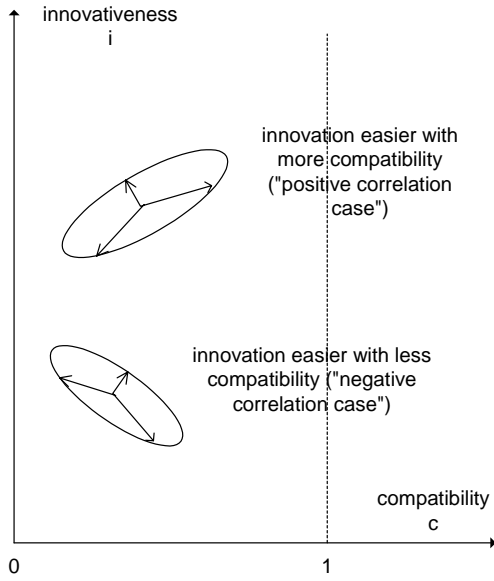
The compatibility level means compatibility with a common standard (it is assumed that there is only one standard), 0 denoting full incompatibility, 1 denoting full compatibility. The extent to which a consumer can benefit from network externalities from another technology depends on the product of his technology's compatibility level and the other technology's compatibility level. A company reducing the compatibility thus reduces the benefit both of its own and its competitors' potential customers.

Consumers' entry and exit from the market follows an overlapping generation concept: a customer enters the market in period t and chooses a technology. In period t+1 he has already committed to his choice and cannot change it, his choice constitutes the installed base. At the end of the period he leaves the market (or he can make a new choice in the following period). Also at t+1 new consumers enter the market and make their choice according to the installed base and expectations about future network sizes.

Firms compete only through the choice of compatibility and innovation. They cannot set prices. (The idea of this simplifying approach was taken from Farrel & Saloner). It should not impose an excessive restriction, as we are only considering competition through the other two parameters. The change compatibility and innovativeness have a joint technological restriction: $g(\Delta c_t^p, \Delta i_t^p) \leq 0$. In other words, a company has to split its effort into innovating and changing compatibility (changes in both parameters can be both negative and positive.) The figure below illustrates the concept.



I will mostly assume that a company can move in one period inside a circle of radius r and the constraint is binding: $g(\Delta c, \Delta i) = \sqrt{\Delta c^2 + \Delta i^2} - r = 0$. One can also extend the model by replacing the circle with an ellipse: it can be easier to innovate while reducing compatibility or while increasing it, as shown in the figure. The former means that a company wanting to innovate as fast as possible has to depart from compatibility.



The timing of the actions is the following:

0. Initially given is $\eta_{t-1} = (\eta_{t-1}^1, \eta_{t-1}^2, \dots, \eta_{t-1}^P)$ and $H_t = \sum_p \eta_{t-1}^p$. The number of new consumers is H_t . The characteristics of the products are $c_{t-1} = (c_{t-1}^1, c_{t-1}^2, \dots, c_{t-1}^P)$ and $i_{t-1} = (i_{t-1}^1, i_{t-1}^2, \dots, i_{t-1}^P)$.

1. Firms form expectations of network sizes η_t and choices of other firms c_t , it and choose compatibility and innovation with constraints. firm $p \in \{1, \dots, P\}$:

$$\max_{c_t^p, i_t^p} \pi_t^p(\eta_t^e, c_t^e, i_t^e) + \delta \pi_{t+1}^p(\eta_{t+1}^e, c_{t+1}^e, i_{t+1}^e) + \delta^2 \pi_{t+2}^p(\eta_{t+2}^e, c_{t+2}^e, i_{t+2}^e) + \dots$$

s.t.

$$\Delta c_t^p = c_t^p - c_{t-1}^p$$

$$\Delta i_t^p = i_t^p - i_{t-1}^p$$

$$g(\Delta c_t^p, \Delta i_t^p) \leq 0$$

2. New consumers choose product: $h \in \{1, \dots, H_t\}$:
consumer

$$\max_{p \in \{1, \dots, P\}} [u_t^h(T_p) + u_{t+1}^h(T_p)]$$

In a fulfilled expectations equilibrium the expected values of η_t , c_t and i_t should be equal to the actual ones. This should also apply to all future value $\{\eta_\tau, c_\tau, i_\tau\}_{\tau=t}^\infty$.

To make the problem numerically solvable, I assumed as a first approximation adaptive expectations: firm p expects $c_t^{q,e} = c_{t-1}^q, i_t^{q,e} = i_{t-1}^q \forall q \neq p$, estimate with these values the consumers' choices and make their decisions accordingly. Consumers use their knowledge of c_t and i_t and expect for the network sizes (=the choices of other consumers) $\eta_t^e = \eta_{t-1}$. Plugging in these expected values, one can solve their maximization problems. The solution $\{\tilde{\eta}_\tau, \tilde{c}_\tau, \tilde{i}_\tau\}_{\tau=t}^\infty$ thus obtained could be used as a second approximation for expectations: $\eta_t^e = \tilde{\eta}_t, c_t^e = \tilde{c}_t$ and $i_t^e = \tilde{i}_t$. The results calculated from the second approximation could be used to iterate one step further $\{\tilde{\tilde{\eta}}_\tau, \tilde{\tilde{c}}_\tau, \tilde{\tilde{i}}_\tau\}_{\tau=t}^\infty$, etc. If the values converge (it is quite possible that they do not) a rational expectations equilibrium has been found.

For the sake of simplicity, I only made the first approximation (adaptive expectations). I try to avoid problems arising from adaptive expectations by evading situations where market

participants are systematically “fooled” by someone with a further horizon: all agents in the model are equally myopic.

I have simulated several situations. Several patterns observed in reality appear in the outcomes of the simulations.

Lock-in

In a setting with two firms who produce incompatible products we have path-dependency as described by Arthur’s model (1989): small historical events in the beginning are self-reinforcing (new customers prefer the more wide-spread platform, therefore it becomes even more wide-spread, etc.). The system will lock in a technology chosen by random.

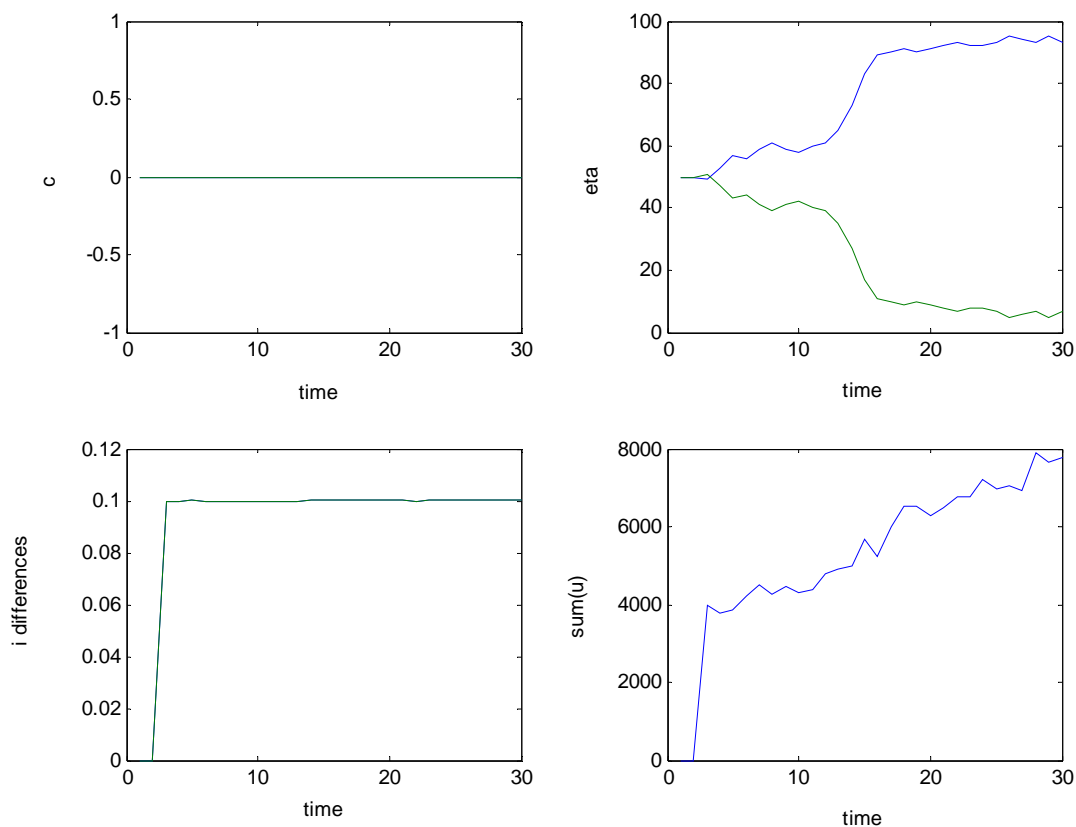


Figure 3

The top left, bottom left and top right graphs of Figure 3 show the evolution of compatibility (c_t^p), innovation (change of i_t^p) and the network size (η_t^p) over time. Compatibility remains 0 throughout in this setting. The bottom right graph is the sum of utilities of the consumers; it is a proxy for the welfare function.

Market leader drops out after reducing compatibility

This is a situation where the initial market leader (dashed line) reduces the compatibility of his product in order to capture as much of the market as possible. Consumers switch to the other two products which are mutually compatible and the initial leader loses market shares drastically. One of the two remaining companies (solid line) manages to secure a high market share and can push out his competitors by making his product incompatible to others.

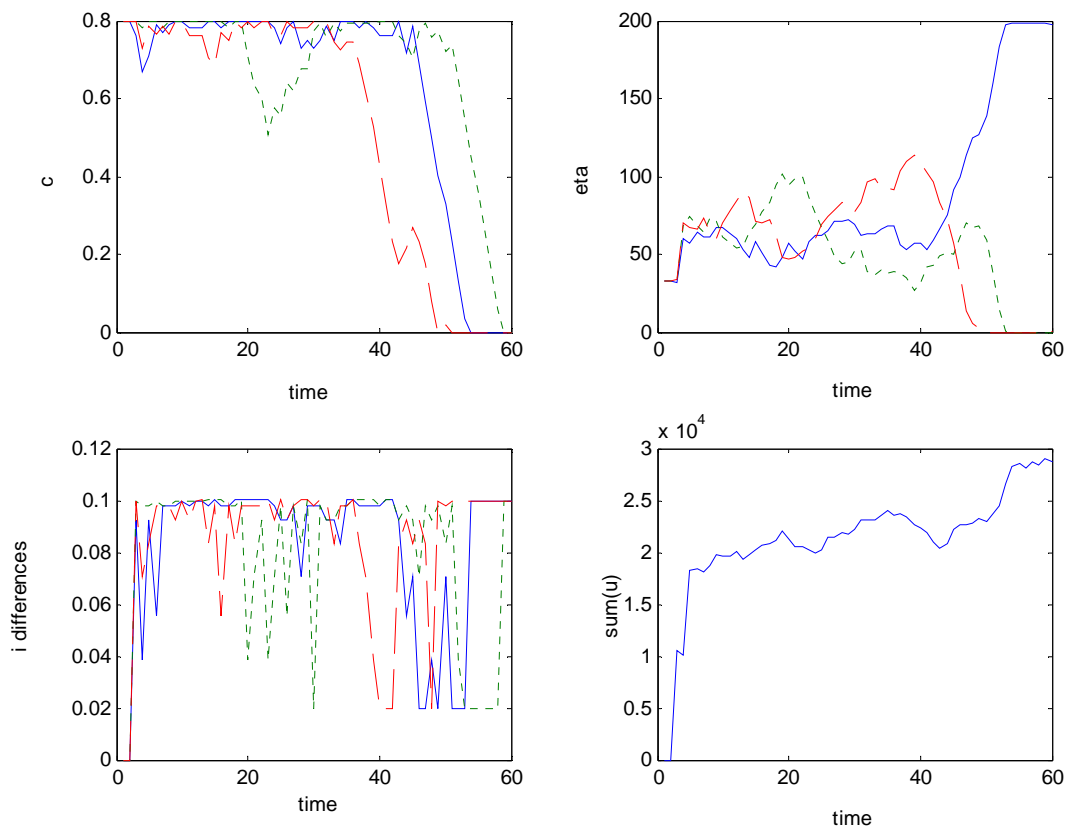


Figure 4

The right timing determines the success: reducing compatibility too early may scare away customers; later on, however, it helps to gain market dominance. Figure 5 shows the market development in a compatibility/innovativeness coordinate system. The technologically given maximal compatibility level in this system is 0.8.

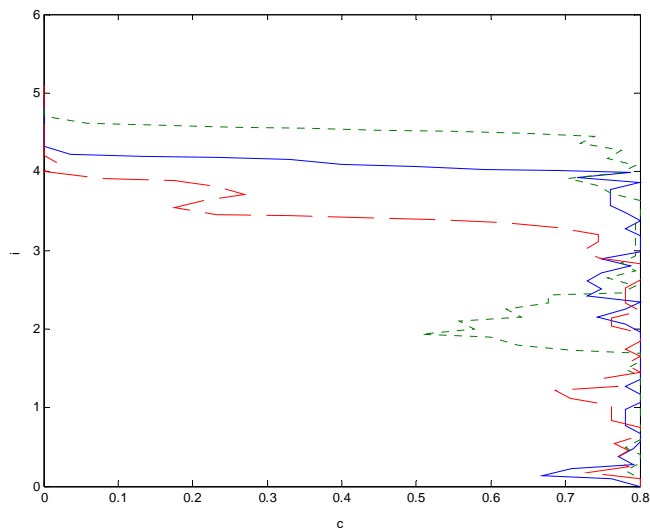


Figure 5

Deliberate reduction of quality

The model shows an environment where it can make sense for a company to temporarily reduce the quality of its product. This is the “positive correlation case” where it is easier to change innovativeness and compatibility in the same direction.

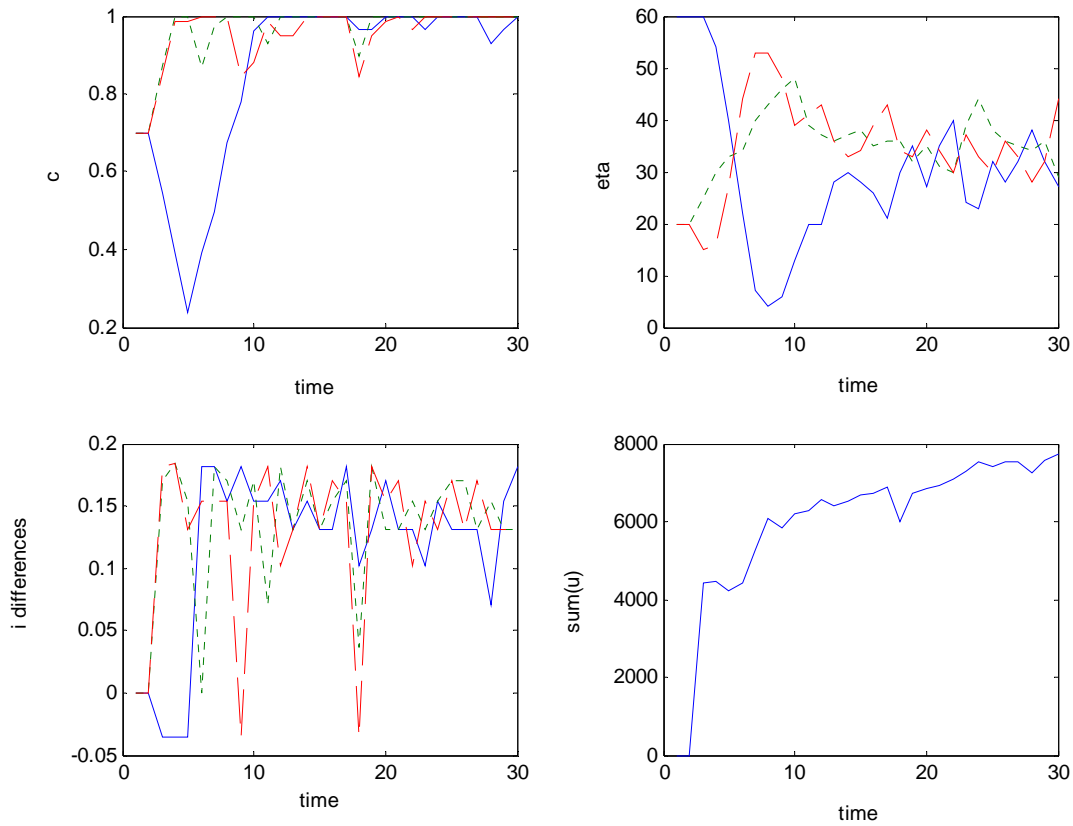


Figure 6

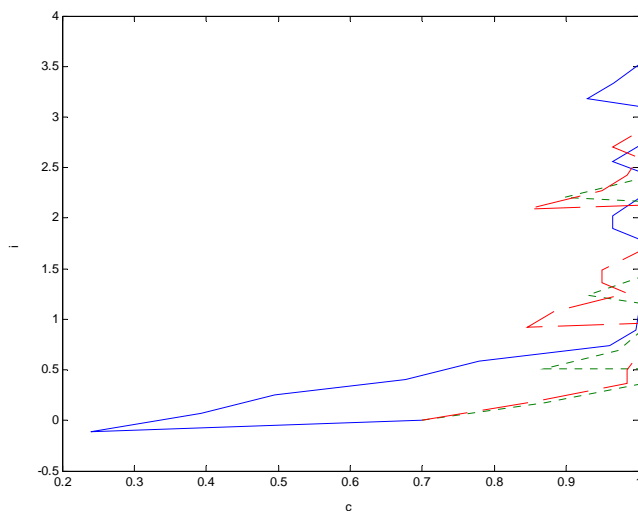


Figure 7

The initial market leader (solid line) tries to capture the market by reducing compatibility drastically at the price of giving up quality. In this example he is not fast enough, the competitors manage to overtake him before the market would lock in; he has to change his strategy in order to survive.

Innovative firm ahead of its time

This example shows a firm that has an innovative product. However, it is not compatible with the wide-spread standard. The firm loses market shares because customers appreciate interoperability with other products. In this “negative correlation” setting the incompatible company tries to increase its compatibility as fast as possible, also by reducing the quality of its product.

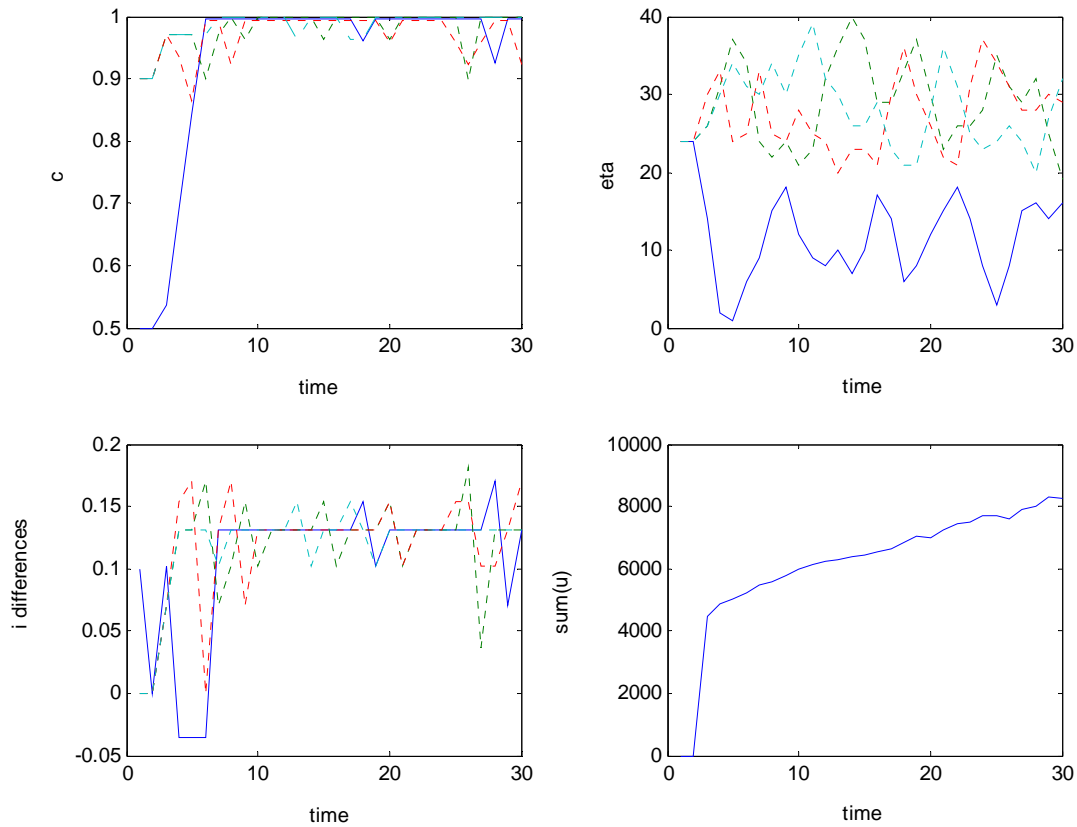


Figure 8

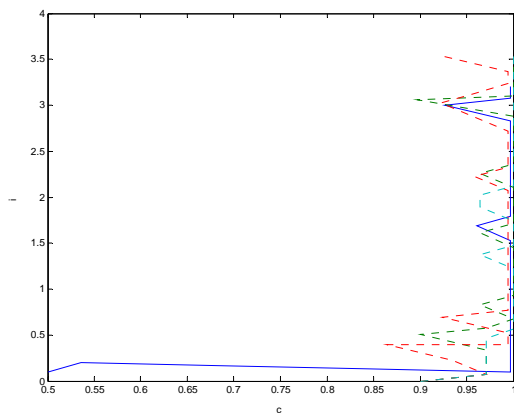


Figure 9

Welfare implications of enforced compatibility

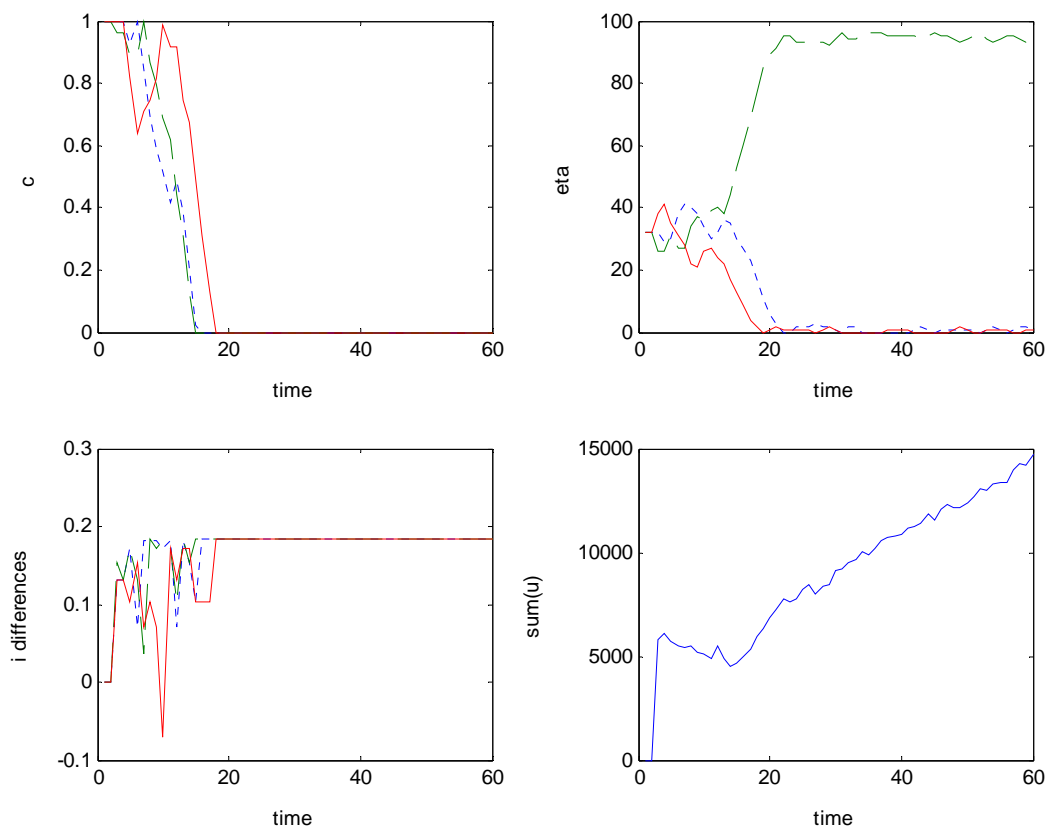
It is interesting to see what happens if the government or a standardization committee enforces compatibility. If the choice of compatibility does not affect or affects positively the possibility to innovate, there is clearly a welfare gain: firms cannot derive private benefit from excluding other firms' customers from their network through incompatibility. However, if compatibility hinders innovation ("negative-correlation case"), the overall welfare effect is unclear: compatibility and a larger network come at the cost of less innovation.

I compare two scenarios:

- Firms compete without any restrictions.
- Firms are obliged to make their product compliant with a standard.

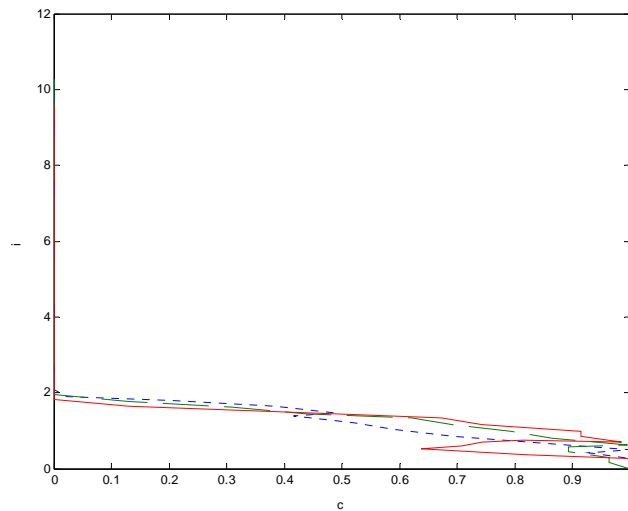
No Compatibility Restrictions

Initially a shake out and a "compatibility war" occurs between the companies. Firms with lower market shares fearing that their customers may be excluded from the network of compatible products initially sacrifice some innovation to preserve compatibility. Firms believing that they have the strength to take over the whole market will attempt to do so by innovating at the expense of compatibility. If they are lucky their timing is right, and they succeed to establish a monopoly (dashed line). If not (solid line and dotted line), they will be punished for deviating from the standard by their customers. They may try to retreat to the standard again (solid line).



At the beginning, there are welfare losses from the incompatibility, because a network of compatible products is destroyed. After one company has seized monopoly power, all

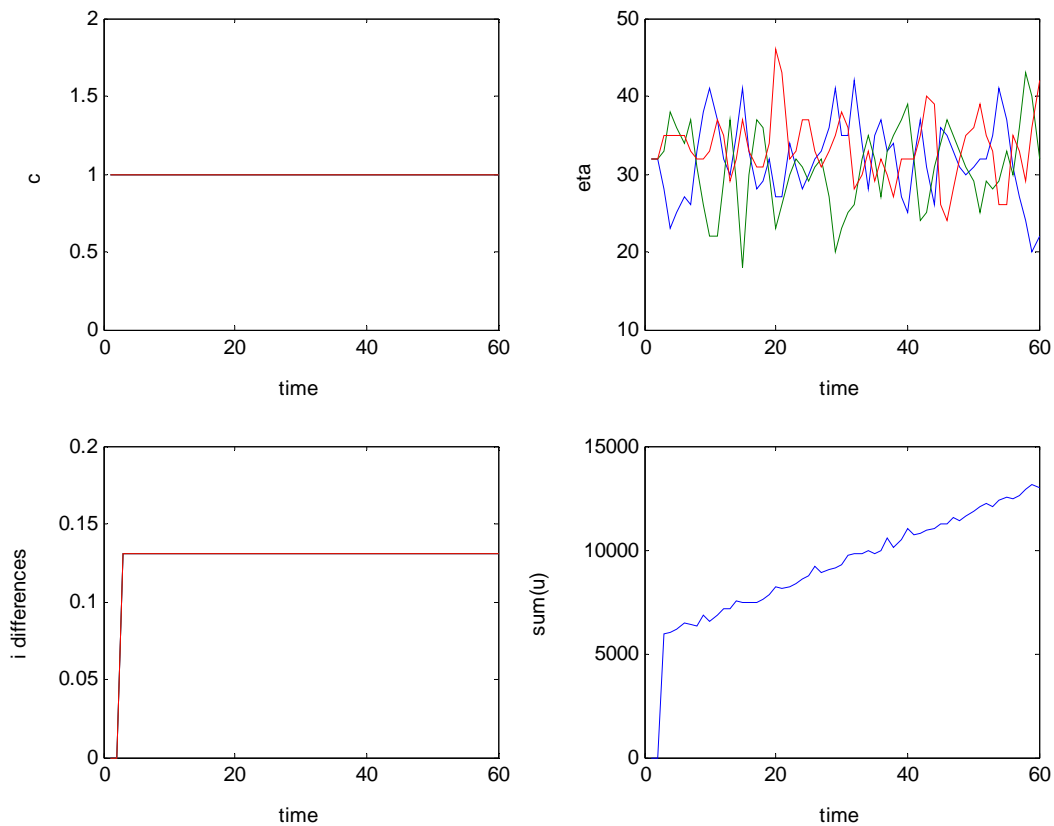
customers belong to the network and innovation can continue without concern for



compatibility.

Obligatory Compatibility

When compatibility is enforced, consumers can benefit from a large network of compatible products from the beginning on. The negative side is that innovation is limited. Note that the fluctuation in the number of customers of a firm is only due to the random noise.



Comparison of Welfare Effects

In the short run, compatibility restriction lead to higher welfare, because they prevent harmful “standards wars”. In this example, in the long run the absence of restrictions leads to more

welfare, because a monopolist will emerge, establishing his proprietary technology as a de facto standard. He can innovate without being hindered by compatibility, because all the customers belong to his network.

Even though this result sounds plausible, one has to be careful. The simplifying assumption that a monopolist does have the incentives to innovate is problematic. One might argue that the threat of a catastrophic market entry forces him to do so, but due to the low chances of success incentives should be lower.

Interpretation of Results

Because the simulation is based on strong simplifications, the parameters chosen are arbitrary and for some parameter values the outcomes are non-stationary processes, the predictive power of the model is very limited. It is rather meant as an illustration of intuitions expressed implicitly or explicitly in the debate over the role of compatibility, innovation and monopolies in the software market.

The model illustrated how timing can play a crucial role in a competition *for* a market when compatibility is endogenous. It also shows why it can make sense for a company to temporarily reduce both the quality and the compatibility of their product, in order to seize monopoly power.

The marginalization of firms with highly innovative products due to lack of compatibility, a phenomenon often lamented by computer scientists, could also be demonstrated.

The model gives an intuition for the mixed welfare effects of standardization: if standards do not hold back innovation, they are welfare improving. If standards do interfere with innovation, a monopoly setting a de facto standard might be more efficient, as long as it has to fear that a competitor might take over the market.

Conclusions

It has been clearly stated by the courts that Microsoft did engage in anti-competitive behavior to stop Netscape and Java from jeopardizing its monopoly in the operating systems market. However, it lies in the very nature of the software markets and the catastrophic entries occurring in them that it is very difficult – when not impossible – to predict what would have happened if Microsoft had not engaged in illegal behavior. Therefore, it is not clear to what extent consumers have been harmed.

In software markets two problems have to be faced. On one hand, markets often seem to fail the socially optimal level of compatibility, possibilities for benefits arising from network externalities are not fully exploited. On the other hand, government intervention also bears the risk of mistake: enforcing compatibility might lead to suboptimal innovation in some cases, a break-up of a monopoly may lead to a harmful interim “standards war” before a new monopoly emerges.

Possible solutions to the problems could be the enforcement of compatibility where it obviously does not interfere with innovation (the publishing or licensing of Application Programming Interface and client/server protocol specifications as it is happening under the Microsoft-DOJ settlement; supervision of compliance with published standards, where a company has declared to observe them like the World Web Consortium’s XML or ECMA’s ECMAScript¹² and Common Intermediate Language).

If compatibility can hinder innovation, ensuring a level playing field for two products *before* a catastrophic entry has happened improves the situation. The reason is that this causes innovativeness of a product to have more, and strategic behavior and chance less, impact on the market outcome. This is happening at the moment in Sun’s private anti-trust suit against

¹² Also known as JavaScript

Microsoft: if the current ruling is not reversed by an appeals court, Microsoft will be forced to ship not only .NET but also the newest version of Java with its Windows XP operating system.¹³

Appendix

Java Products and Their Competitors

Aspect	Java Product	Competitors
Programming language for Graphical User Interfaces (GUIs)	Java Standard Edition/Enterprise Edition/Micro Edition	MS Visual Basic and MS Visual C++ (only for MS Windows), Smalltalk, Python, Delphi (originally only for MS Windows, now a port for Linux is available), others
Programming language for web servers	Java Server Pages and Java Servlets (Java EE)	MS Active Server Pages (in combination with VBScript, JavaScript and PerlScript), MS ASP.NET (new version of ASP), Perl (CGI scripts, mainly Apache), PHP (Apache Web Server), Zope (based on Python), others
Programming languages in general	Java	MS VisualBasic, C/C++, Delphi, Perl, Python, Eiffel, Smalltalk, Matlab, PL/SQL, Assembler, others
Virtual Machine Platform	Java VM	native code, MS Java Virtual Machine, Smalltalk/Python/Perl bytecode, <i>MS Common Language Runtime (launched with .Net)</i>
Operating System	Java as an operating system independent middleware; Java OS	MS Windows, Linux/Sun Solaris/other Unix derivatives, Macintosh
Class/Function Library	Java Foundation Classes	MS Foundation Classes, class collections of Delphi, Naglib, LAPACK, others
Communication protocols/distributed components	Enterprise Java Beans (Java EE) with RMI-IIOP or Corba as communication layer; Remote Method Invocation (Java SE)	HTTP/HTML, Common Object Request Broker Architecture (CORBA, an open multiplatform multilanguage standard), MS Distributed Component Object Model (DCOM), MS Remote Procedure Calls, XML-RPC, <i>MS .NET</i>
Software Components	Java Beans	MS Component Object Model (COM/ActiveX), CORBA, Kpart (for Linux' KDesktop Environment), Objects in <i>.NET</i>
Database access protocols	Java Database Connectivity (JDBC)	Open Database Connectivity (ODBC, Windows only, although Linux ports have become available recently), proprietary products, <i>.NET database access protocols</i>
Software Platform for Mobile Devices	Java Micro Edition	<i>MS Common Language Runtime (part of .NET)</i> , various proprietary solutions

Table 1

¹³ see <http://www.computerworld.com/governmenttopics/government/legalissues/story/0,10801,77621,00.html>

Matlab Function 'compatibility_competition'

```
function
[eta,Eta,c,i,u]=compatibility_competition(nsteps,p_n_firms,Eta,eta,c,i,n_fi
rms_choices,std_epsilon,max_comp,firms_choices,p_weight_i,min_comp)
% function running the actual simulation with the parameters given
% (see run_compatibility_competition.m for details)
global n_firms
global weight_i
weight_i=p_weight_i;
n_firms=p_n_firms;
neweta(1,:)=ceil(eta(1,:)/2);
neweta(2,:)=ceil(eta(1,:)/2);
% auxiliary variable for 'approximate equalities'
very_small=0.00001;

for j=3:nsteps
% number of consumers assumed to stay equal throughout
Eta(j)=Eta(1);
% random noise for one period
epsilon=random('Normal',0,std_epsilon,Eta(j),n_firms);
% make choice for every firm
for p=1:n_firms
% vector of (profit,c,i) with maximal profit
maxprofit=[0;0;0];
% vector of (c,i) with possible choice
mychoice=[c(j-1,p);i(j-1,p)];
% iterate through all possible choices
for fc=1:n_firms_choices
% if choice feasible (c not too high)
if (mychoice(1)+firms_choices(1,fc)<=max_comp+very_small)
% if choice still feasible (c not too low)
if (mychoice(1)+firms_choices(1,fc)>=min_comp-very_small) |
(min_comp==0)
% find out whether mychoice is optimum
mychoice=[c(j-1,p);i(j-1,p)]+firms_choices(:,fc);
if mychoice(1)>max_comp-very_small
mychoice(1)=max_comp;
elseif mychoice(1)<min_comp+very_small
mychoice(1)=min_comp;
end
prof=profit(mychoice,eta(j-1,:),c(j-1,:),i(j-
1,:),p,epsilon,Eta(j));
if prof>=maxprofit(1)
maxprofit(1)=prof;
maxprofit(2)=mychoice(1);
maxprofit(3)=mychoice(2);
end
end
end
end
c(j,p)=maxprofit(2);
i(j,p)=maxprofit(3);

% check if natural lower boundery (0) of compatibility reached
% a firm can achieve maximal innovation even if its choice is
% restricted by c>=0 (after having reached full incompatibility)
if (min_comp==0) & (mychoice(1)==0)
temp=max(firms_choices');
i(j,p)=i(j-1,p)+temp(2);
end
end
end
% choice of firms has been calculated
```

```

    % now calculate choice of consumers
    neweta(j,:)=zeros(1,n_firms);
    epsilon=random('Normal',0,std_epsilon,Eta(j),n_firms);
    for j_cons=1:Eta(j)
        [choice,util]=consumer_choice(eta(j-
1,:),c(j,:),i(j,:),epsilon(j_cons,:));
        neweta(j,choice)=neweta(j,choice)+1;
        u(j,j_cons)=util;
    end
    eta(j,:)=neweta(j,:)+neweta(j-1,:);
end

% estimate profit of firm p given the other parameters
function prof=profit(choice,p_eta,p_c,p_i,p,epsilon,Eta)
global n_firms
tempc=p_c;
tempi=p_i;
tempc(p)=choice(1);
tempi(p)=choice(2);
prof=0;
% estimate choice of each consumer
for i=1:Eta
    [choice,util]=consumer_choice(p_eta,tempc,tempi,epsilon(i,:));
    if choice==p
        prof=prof+1;
    end
end
end

% calculate choice and maximal utility for a consumer
function [choice,maxutil]=consumer_choice(eta,c,i,epsilon)
global n_firms
global weight_i
%% scalar way to find product maximizing utility
% maxutil=-1000;
% for p=1:n_firms
%     u=eta(p)+c(p)*(c*eta'-c(p)*eta(p))+weight_i*i(p)+epsilon(p);
%     if u>=maxutil
%         maxutil=u;
%         choice=p;
%     end
% end
%% however the vector notation is faster:
u=eta+c.*(c*eta'*ones(1,n_firms)-c.*eta)+weight_i*i+epsilon;
[maxutil, choice]=max(u);

```

Matlab M-Script 'run_compatibility_competition'

```

for i=1:5
% name of case to be simulated
sim_case='wo_low';
sim_num=i;

% output files
variable_file=sprintf('essay/variables_%s_%d.mat',sim_case,sim_num);
comp_fig=sprintf('essay/comp_fig_%s_%d.fig',sim_case,sim_num);
ci_fig=sprintf('essay/ci_fig_%s_%d.fig',sim_case,sim_num);

% number of steps calculated, t=1..nsteps
nsteps=60;
% intial number of firms/products
n_firms=3;
% size of networks in each period considered

```

```

eta=[];
% total number of consumers in each period
Eta=[];
% number of new consumers arriving
neweta=[];
% compatibility levels for each firm in each period
c=[];
% innovativeness for each firm in each period
i=[];
% initial values
neweta(1,:)=[16 16 16];
neweta(2,:)=neweta(1,:);
eta(1,:)=2*neweta(1,:);
eta(2,:)=2*neweta(1,:);
Eta(1)=sum(neweta(1,:));
Eta(2)=sum(neweta(2,:));
% technological maximum of compatibility
max_comp=1;
% minimal level of compatibility (e.g. set by law, contracts, etc.)
min_comp=0;
% initial values for c and i
c(1,:)=1*ones(1,n_firms); %max_comp*ones(1,n_firms);
i(1,:)=zeros(1,n_firms);
c(2,:)=c(1,:);
i(2,:)=i(1,:);
% number of choices a firm faces, the circle (or ellipse) is
% approximated as a collection of n_firms_choices points on
% its outline
n_firms_choices=16;
% standard deviation of white noise epsilon
std_epsilon=30;
% maximal change in compatibility/innovativeness a firm can achieve
% (=radius of circle)
change_per_period=.1;

temp=[0:pi/n_firms_choices:2*pi];
% term making the choice circle to an ellipse with "positive correlation"
pos_corr=cos([-pi/4:pi/n_firms_choices:2*pi-pi/4;-
pi/4:pi/n_firms_choices:2*pi-pi/4]);
% term making the choice circle to an ellipse with "negative correlation"
neg_corr=cos([pi/4:pi/n_firms_choices:2*pi+pi/4;-
3*pi/4:pi/n_firms_choices:2*pi-3*pi/4]);
% circle (or ellipse) approximated by a set of numbers
firms_choices=change_per_period*( [cos(temp);sin(temp)]+0*pos_corr+1*neg_cor
r);
% weight of innovativeness in consumers' utility function
weight_i=20;

% run simulation
[eta,Eta,c,i,u]=compatibility_competition(nsteps,n_firms,Eta,eta,c,i,n_firm
s_choices,std_epsilon,max_comp,firms_choices,weight_i,min_comp)

% save output variables
save(variable_file)

% create figure with plots of compatibility,
% innovativeness, network sizes and welfare function
fig=figure;
hc=subplot(2,2,1);
plot([1:nsteps]',c);
set(hc,'YLabel',XLabel('c'));
set(hc,'XLabel',XLabel('time'));
idiff=i-[zeros(1,n_firms);i(1:nsteps-1,:)];

```

```
hidiff=subplot(2,2,3);
plot([1:nsteps]',idiff);
set(hidiff,'YLabel',XLabel('i differences'));
set(hidiff,'XLabel',XLabel('time'));
heta=subplot(2,2,2);
plot([1:nsteps]',eta);
set(heta,'YLabel',XLabel('eta'));
set(heta,'XLabel',XLabel('time'));
hu=subplot(2,2,4);
plot([1:nsteps]',sum(u)');
set(hu,'YLabel',XLabel('sum(u)'));
set(hu,'XLabel',XLabel('time'));
set(fig,'FileName',comp_fig);

% create figure with compatibility/innovativeness path
fig2=figure;
h2=subplot(1,1,1);
plot(c,i);
set(h2,'YLabel',YLabel('i'));
set(h2,'XLabel',XLabel('c'));
set(fig2,'FileName',ci_fig);

end
```

Arthur, W. Brian. 1989. Competing Technologies, Increasing Returns, and Lock-In by Historical Events. *The Economic Journal*

Coase, R. 1937. The Nature of the Firm. *Economica* n.s. 4: 386-405. Reprinted in *Readings in Price Theory*, ed. G. Stigler and K. Boulding (Homewood, Ill.: Irwin, 1952)

Economides(www): (Nicholas Economides' website):

http://www.stern.nyu.edu/networks/Analysis_Appeals_Court_Decision.htm

Farrell, J., and G. Saloner. 1985. Standardization, Compatibility, and Innovation. *Rand Journal of Economics* 16: 70-83

Farrell, J., and G. Saloner. 1986a. Standardization and Variety. *Economics Letters* 20: 71-74

Farrell, J., and G. Saloner. 1986b. Installed Base and Compatibility: Innovation, Product Preannouncements, and Predation. *American Economic Review* 76: 940-955

Katz, M., and C. Shapiro. 1985a. Network Externalities, Competition, and Compatibility. *American Economic Review* 75: 424-440

Katz, M., and C. Shapiro. 1986a. Technology Adoption in the Presence of Network Externalities. *Journal of Political Economy* 94: 822-841

Katz, M., and C. Shapiro. 1986b. Product Compatibility Choice in a Market with Technological Progress. *Oxford Economic Papers* 38: 146-165

Shapiro, C. and H. R. Varian. 1999. Information Rules

Shy, O. 1996. "Technology revolutions in the presence of network externalities." *International Journal of Industrial Economics* 14: 785-800.

Shy, O. 2001. The Economics of Network Industries. *Cambridge University Press*

Tirole, J. 1994. The Theory of Industrial Organization

Williamson, O. 1975. Markets and Hierarchies: Analysis and Antitrust Implications.

New York: Free Press