Autor / Author (E-Mail):	*Michael Beurskens, LL.M. / Dr. Dirk Zetzsche, LL.M.*
E-Mail:	dirk.zetzsche@uni-duesseldorf.de ; michael.beurskens@uni-duesseldorf.de

Titel / Title:	**Regulation of the Market for Standard Software:** **Waving Goodbye to "Efficiency"?**
CBC Nummer / Number:	CBC-RPS 0013 (06/2005)
SSRN Nummer / Number:	http://ssrn.com/abstract=747505
Schlagworte / Keywords:	market, standard software, regulation, legal frameworks, intellectual property, Chicago School concept, efficiency, antitrust law
JEL Classifications:	K10, K20, K21, K23, L10, L12, L13, L40, L41, L43

Regulation of the Market for Standard Software:
Waving Goodbye to "Efficiency"?[*]

Dirk Zetzsche[+] / *Michael Beurskens*[++]

Abstract: Due to the economics of the market for standard software, it is reasonable to assume that the market shows an inexorable tendency towards concentration. Whereas such a development might be considered efficient due to the market's nature as a natural monopoly, it could also require stronger regulation, similar to other concentrated industries. This paper analyzes to what extent the existing legal frameworks under U.S. and European intellectual property and antitrust law mitigate pro-concentration effects. It characterizes the standard software market as an industry in need of regulation and reconsiders the current efficiency-focused approach in favor of measures that strengthen competition as a source of rivalry. Rather than considering efficiency arguments as "all-or-nothing" justification for otherwise anti-competitive conduct, as under the Chicago School concept, the concept proposed in this paper considers the "efficiency" criterion to merely limit the remedies imposed by adjudicators.

This paper presents work in progress. Please don't hesitate to be critical.

Please send comments to:

michael.beurskens@uni-duesseldorf.de or dirk.zetzsche@utoronto.ca

[+] Dr. jur. (Ph.D./J.S.D. equivalent), Heinrich-Heine-University, Düsseldorf, Germany [HHU], LL.M. (University of Toronto), Habilitand at the Professorship for Civil Law, Commercial and Business Law (Prof. Dr. Ulrich Noack) and Research Fellow at the Center for Business and Corporate Law (CBC) at HHU; Researcher at the Center of Innovation Law and Policy at the University of Toronto

[++] LL.M. (Intellectual Property Law, University of Duesseldorf) und LL.M. candidate (University of Chicago), Researcher at the Center for Intellectual Property Law (Prof. Dr. Jan Busche) at HHU.

Table of Contents

I. Introduction

The battles of standard software[3] manufacturers fill the pages of major newspapers almost every day.[4] An increasing amount of literature[5] and high-profile cases such as antitrust measures against Microsoft[6] or the hostile takeover of PeopleSoft by its rival Oracle[7] evidence the increased regulation of the market for standard software. In this paper, we analyze the regulatory environment of the market for standard software under the premise that the market's economic characteristics drive the market towards a monopolistic structure. We analyze the peculiarities of this market from both the U.S. and the European perspectives, and argue that "efficiency" should merely limit the remedies imposed rather than function as a broadly used justification for otherwise anti-competitive conduct.

We will begin by providing a brief overview of the economics of the standard software market (sub II). Subsequently the paper will maintain that the current regulatory frameworks of intellectual property law and antitrust law increasingly drive the market towards monopolization (sub III). Our analysis of these

---

[3] We use the term „standard software" to define mass market software sold to a large group of consumers as contrasted to "individual software" designed for a specific company. Insofar Microsoft Windows and Microsoft Office are evident examples of standard software, whereas an accounting program specifically developed for a car-manufacturer would be individual software. The distinction is important as competition in the market for individual software is still extremely strong, however the increasing customizability of standard software poses a growing threat. We acknowledge that some software might be hard to categorize as there is a grey area of programs used by multiple customers, but still not marketed on the shelves of computer-stores or offered directly to the public.

[4] See for example The New York Times, *Microsoft Settles With Novell And Joins An Opponent* (November 9, 2004, Section C; Pg. 1); The Seattle Times, *Microsoft to face close scrutiny of Longhorn work* (January 27, 2005; Pg. E1); USA Today, *Microsoft gives in to European demands* (January 25, 2005, Pg. 1B); The Washington Post, *PeopleSoft Deadline Extended* (January 6, 2005; Pg. E02); The San Francisco Chronicle, *It's time to send the bill to Bill; Deadline to claim share of Microsoft settlement is Jan. 8* (December 29, 2004, Pg. C1); The Boston Globe, *Microsoft loses Antitrust Appeal – Firm will strip Media Player from Windows for European Market* (December 23, 2004, Pg. C1); The Times, *Microsoft should admit the war is over* (December 23, 2004, p. 33)

[5] *See e.g.* Jay P. Kesan & Andres A. Gallo, *Optimizing Regulation of Electronic Commerce*, 72 U. Cin. L. Rev. 1497 (2004); Barbara Crutchfield George & Lynn Vivian Dymally & Kathleen A. Lacey, *Increasing Extraterritorial Intrusion of European Union Authority into U.S. Business Mergers and Competition Practices: U.S. Multinational Businesses underestimate the Strength of the European Commission from G.E.-Honeywell to Microsoft*, 19 Conn. J. Int'l L. 571 (2004); Jeffrey A. Andrews, *Reversing Copyright Misuse: Enforcing Contractual Prohibitions on Software Reverse Engineering*, 41 Hous. L. Rev. 975 (2004); Boris Rotenberg, *The European Regulation of Communications Software: Building a "Platform" for Freely Interoperable Digital Expression?*, 8 Int'l J. Comm. L. & Pol'y 2 (2004); Joseph P. Liu, *REGULATORY COPYRIGHT*, 83 N.C.L. Rev. 87 (2004).

[6] See United States v. Microsoft Corp, 97 F. Supp. 2d 59 (D.D.C., 2000); United States v. Microsoft Corp., 346 U.S. App. D.C. 330 (D.C. Cir., 2001); European Court of First Instance, Order by the President of, decision T-201/04 R, 2004 ECJ CELEX LEXIS 390 [CFI-Order]; European Commission Decision, Decision of 24.03.2004, C(2004)900 final (published 21.04.2004), COMP/C-3/37.79 [EC, Microsoft-decision], and Press Release of March 24th, 2004, IP/04/382 {EC, Microsoft-Press Release].

[7] *See* In re PeopleSoft, Inc., Sec. Litig., 2000 U.S. Dist. LEXIS 10953 (D. Cal., 2000); United States v. Oracle Corp., 331 F. Supp. 2d 1098 (D. Cal., 2004): recently The San Francisco Chronicle, *Oracle takeover is now official; PeopleSoft no longer exists after $10.3 billion deal* (January 8, 2005, Pg. C1); The New York Times, *With Takeover Completed, Oracle Sends Layoff Notices* (January 15, 2005 Section C; Pg. 2); the European Commission cleared the merger October 26, 2004, see Press Release IP/04/1312.

frameworks shows that adhering to "**efficiency**" furthers the drive of the software market towards market concentration. We suggest addressing these weaknesses by treating the standard software market as a "regulated industry" - a step, which would enable regulators to push for the most efficient outcome for society regardless of the fact that this step might harm some market participants. However, in contrast to other regulated industries (that are typically characterized by idiosyncratic regulatory frameworks), we believe that the standard software market is best regulated implicitly, through means already provided by antitrust and intellectual law. However, we argue in favor of some adjustments of the current regimes. In particular, this implies assessing long-term effects of competitors' conduct and regulatory responses on competition and furthering competition as a process of rivalry instead of blindly adhering to short-term-efficiency considerations (sub IV). The "efficiency" criterion will retain a use in limiting potential remedies imposed by courts and regulators (sub V).

II. Characteristics of the Standard Software Market

The markets for standard software are characterized by a natural monopoly structure, but also has a public goods nature with network externalities and highly abstract products that are subject to rapid innovation.

The term "**natural monopoly**" describes a situation in which a single firm can produce a given level of output at a lower total cost than can any combination of multiple firms. This means that instant scalability of production coincides with high fixed costs and low marginal costs for a single product unit.[8] Any competitor who wants to enter into the market must undertake huge investments (entry-costs) before offering even a single unit.[9] Thus, natural monopolies are generally a supply side phenomenon. Further, in traditional natural monopoly industries, such as electricity and hard-wire telephony in the local loop, the incumbent's high fixed costs for production have been paid off, while new entrants face the burden of significant investments in order to even begin producing their first unit.

---

[8] Richard A. Posner, *Economic Analysis of Law*, Sixth Edition (2003), p. 361; recently Aditya Bamzai, The Wasteful Duplication Thesis in Natural Monopoly Regulation, *71 U. Chi. L. Rev. 1525* (2004); see also Richard A. Posner, Natural Monopoly and Its Regulation, 21 Stan L. Rev. 518 (1969).

[9] Even then, he will have to price his product much higher than the incumbent supplier, since the incumbent can regain his fixed costs by distributing them over a much larger number of units, due to established customer relationships, than the new entrant can do.

The markets for standard software exhibit the same characteristics: New market entrants face high initial costs to develop the essential software framework[10], whereas these costs have already been paid off by the incumbents. Under these conditions, rational investors will not provide new entrants with capital because the initial costs will inevitably exceed market prices - unless the new entrant's product exhibits significant technological advantages which the incumbent platform lacks. In addition, purchasers of a new basic platform require the ability to re-use their old documents and applications and must be able to exchange documents with others still employing the legacy framework.[11] This requirement of backward-compatibility due to interdependencies with third-party applications (like Word Processors) limits the possibility of change.[12] Consequently it is highly unlikely that any product will instantly reach the mass-market unless it provides high compatibility to others using the existing platforms – more simply put, existing monopolies tend to persist. Further, since the flexible design of standard software makes it fit the needs of a wide range of users, the initial high investments can be distributed among numerous sales.[13] Whether the effect of this characteristic on the market is significant, is contentious.[14] We believe, however, that no software firm would provide for various, at a first glance: useless, software functions to its customers essentially for free (assuming that customers would not pay for the additional functions at the moment of purchase) if this strategy did not

---

[10]   See Melissa A. Schilling, Technological Lockout: An Integrative Model of the Economic and Strategic Factors Driving Technology Success and Failure, 23 Acad. Mgmt. Rev. 267, 270 (1998); think of an operating system (e.g. Windows) Application Programming Interface (API) to run third-party applications (e.g. Office) on it or (increasingly important – Microsoft e.g. reports over 2.3 million developers for Office 2000 Developer) the interfaces for macros, plugins or add-ons to standard Office-Products.

[11]   This can be illustrated by the change in file-formats from Microsoft Office 95 to Microsoft Office 97 (especially with regards to Microsoft Word). Most purchasers of the new (and arguably better) version complained that they were unable to collaborate on documents or even exchange them with users who had not upgraded. Microsoft was quick to add improved export-filters later on and has assured that later versions of Office remained "backward-compatible".

[12]   See Mark Cooper, Antitrust as Consumer Protection in the New Economy: Lessons from the Microsoft Case, 52 Hastings L.J. 813 (2001) at p. 825; Hal Varian & Karl Shapiro, Information Rules (1999); e.g. Microsoft moved from 16-bit-Windows 3 to 32-bit-Windows 95 and NT and nowadays supports a new ".Net-Framework" while still keeping their systems compatible. Java and Linux have been struggling to convince Windows-users to move over due to lack of applications able to edit documents created by Microsoft Word, Excel or run expensive specialized applications (such as law firm management tools) created specifically for Microsoft Windows. Both issues have been remedied for the Linux-platform recently, by the release of the OpenOffice.org-suite (www.openoffice.org) and improvements in the Windows-compatibility-layer Wine (www.winehq.com), but still persist.

[13]   For example Windows includes code to allow users to run fast 3D-games (DirectX) which would be of little use on a secretary's PC only used to write letters. Windows also includes a complete set of network-management features which are useless on a PC which is not connected or code to print which is never used on a computer without a printer, see the list of features removable by the tools "98Lite" (http://www.litepc.com/matrix.html) and "XPLite" (http://www.litepc.com/xplite.html)

[14]   Stanley Liebowitz & Stephen Margolis, *Winners, Losers & Microsoft: Competition and Antitrust in High Technology* (Oakland, CA: Independent Institute, 2001), at *; but see also the convincingly argued counter-position in Ariel Katz, *A Network Effects Perspective on Software Piracy,* 55 U Toronto L J 155 (2005).

enhance overall demand for the program. The existence of this strategy itself accounts for the position that it has, in fact, an effect on the market.

Like all types of intellectual property, software exhibits some characteristics of **a public good**. In particular, it can be consumed without reducing any other person's consumption of it.[15] Whereas the costs of initially developing the software are high (as outlined above), the costs of producing additional copies are (almost) non-existent. This is unlike other "natural monopoly industries", where instant scalability is limited. For example, in electricity and wired telephony, the capacity of the wires limits the exercise of market power to a certain area, and data- or energy-transfer to a certain volume. Similar barriers do *not* exist in the software market, as far as the product itself is concerned. Software companies can reproduce their software to an unlimited extent, without affecting the product's quality.[16] Limits to instant scalabity (which may mitigate the producer's drive for growth) merely exist with respect to distribution of, as well as support and training for the software program. Since all of these functions may take place over the Internet, and support is required to a lesser extent as compared to other product types, the effects of these limits can be deemed insignificant.

Further, the standard software market exhibits **network externalities**. Network externalities occur when the utility that a user derives from consumption of a good increases with the number of other agents consuming the same good.[17] With regard to software the value of a standard increases with the number of people using it.[18] For example, software end-users benefit from learning to employ the user-interface of Microsoft Windows because they can assume this interface will appear on almost all computers they encounter.[19] Similarly, programmers develop the standardized Microsoft Windows Application Programming Interface to achieve the broadest possible market for their products.[20] Hardware manufacturers supply the

---

[15]   Posner, *Economic Analysis of Law*, *supra* note 8, p. 41.

[16]   However the broader the user-basis the higher risk for abuse, e.g. targeting by viruses or hackers – insofar the number of copies actually imposes a higher risk on the users (and thereby lowers the product's value).

[17]   Mark A. Lemley & David McGowan, *Legal Implications of Network Economic Effects*, 86 CALIF. L. REV. 479, 503-507 (1998); Michael L. Katz & Carl Shapiro, *Network Effects, Competition, and Compatibility* 75 Am. Econ. Rev. 424 (1985), at 424; see Posner, *Economic Analysis of Law*, *supra* note 8, p. 324; recently R. Polk Wagner, *On Software Regulation* 78 S. Cal. L. Rev. 457 (2005), at p. 476.

[18]   See Michele Kelber, Living in America: Antitrust Law and the Software Industry; What Can We Do?, 9 U. Balt. Intell. Prop. L.J. 155, 159 (2001); Jerre B. Swann, Jr., *Protecting Intellectual Property within Horizontal Exchange Relationships*, 2 J. Intell. Prop. L. 363 (1994) at p. 364; For the risks of Computer Software standardization see Douglas A. Barnes, *Deworming the Internet*, 83 Tex. L. Rev., 279 (2004) at p. 289-302.

[19]   See Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 Stan. L. Rev. 1045, at pp. 1066-1071.

[20]   See United States v. Microsoft Corp., 65 F. Supp. 2d 1, 12-14 (D.D.C. 1999).

necessary device-drivers only for major operating systems.[21] Finally, users and programmers benefit from the interchangeability of complementary products.[22] Network externalities in the software market increased significantly with the growth of the Internet in the late 1990s, which promoted exchange of electronic documents instead of printed hard-copies. Therefore, the "free" Internet ironically strengthened the market power of formats such as Word's ".doc" or Acrobat's ".pdf".

If network externalities exist, the presence of two incompatible products in a single market will create an unstable environment as they cannot profitably coexist. As a consequence, the market might eventually decide in favor of only one product ("**tipping**").[23] Thus, network externalities press the market towards a single (standard) product developed under standards proprietary[24] to the firm with a dominant position.[25]

One might argue that the same beneficial standardization can be reached by **open standards** (such as the Common User Application (CUA) defined by IBM,[26] the World-Wide-Web-Standards defined by the World Wide Web Consortium,[27] the standards for the C and C++ programming languages defined by the American National Standards Institute,[28] the standardized office-file-format promoted by OASIS or the definition for JavaScript and C# by ECMA International[29]). However the use of such public standards is limited in real life. One important problem of external standard setting by neutral parties, an industry

---

[21] Interestingly we do not face monopolization in the hardware business, where different companies compete for almost every element of the Personal Computer (PC); on the history of the separation of software and hardware industries see Ronald J. Mann, *Do Patents Facilitate Financing in the Software Industry?*, 83 Tex. L. Rev. 961 (2005) at pp. 968 seq.

[22] E.g. Add-ons, plug-ins and macros for Microsoft Office; applications, screensavers and icons for Microsoft Windows.

[23] Kelber, *supra* note 18, 159 (2001); see, however, on the other hand the evidence against the significance of these effects in the software industry provided by Liebowitz & Margolis, *Winners, Losers & Microsoft*, *supra* note 14.

[24] "Proprietary" generally refers to specifications developed and controlled by a single company. This can be achieved by intellectual property rights (i.e. the standard is legally "owned" by the company) or by keeping it secret and modifying it periodically to keep competitors away (i.e. the standard is only "factually" proprietary). The latter alternative prevails in practice since interfaces as such can often not be protected by copyright or patents due to lack of originality and non-obviousness.

[25] IBM was unable to move the market to OS/2 due to lack of applications and device drivers as well as issues with backward-compatibility. Even Microsoft was hard pressed to convince users to migrate to the Windows NT-platform (beginning with Windows NT 3.1 in and culminating in Windows XP in 2002) – mainly due to lack of device driver compatibility and support for legacy DOS and 16-bit-Windows apps (especially games).

[26] Defining the user-interface across platforms, e.g. Word for Dos 5.5/6.0, OS/2 and Windows 3.x (and with some additions also in Windows 95 to Windows XP).

[27] An international consortium with member organizations world-wide, <www.w3c.org>.

[28] A private, non-profit organization that administers and coordinates the U.S. voluntary standardization and conformity assessment system <www.ansi.org>.

[29] www.ecma-international.org/.

consortium or a single competitor is the rapid pace of innovation: On the one hand, if the standard setter cannot keep up with development, software developers tend to ignore those standards.[30] On the other hand, overly complex standards will not be fully implemented in real-life products.[31] Further, lack of precision often limits the use of standards as they may be interpreted differently.[32] Finally, intellectual property law often hinders adoption of standards (and open interfaces), as adherence to a standard is not a generally accepted defense to patent infringements.[33] This is a particularly dire situation, since incompatibility between products strengthens the position of dominant market players by preventing the exchange of data and by effectively forcing users to use their product(s). This might lead strong competitors to intentionally obscure their data-formats and interfaces to prevent substitution by a competitor's product.[34].

These network externalities also explain Microsoft's fight against "middleware" like Netscape's Navigator and Sun Microsystems' JAVA. Middleware software allows developers to create applications not depending on a specific operating system (such as Microsoft Windows) but instead building upon a standardized and shared middle-layer. Users are insofar able to continue using their existing, operating system dependent software programs, while developers can target the new independent platform instead. Applications that are designed for middleware platforms do not require a specific operating system. If a critical mass of these applications is available, this would significantly reduce the entry costs for new competing operating system manufacturers and the switching costs imposed on users. Some commentators[35] therefore assume that users will eventually switch to various (existing or new) competing operating systems. Since middleware inevitably erodes the lock-in effect that consumers face after having got accustomed to a specific software and therefore furthers flexible

---

[30]   The slow development of the HTML standard lead Netscape to introduce Tables (in Netscape Navigator 1.0) and Frames (in Netscape Navigator 2.0), which were subsequently added to the HTML-standard and are still supported by all web-browser even today. However the "dynamic"-HTML-additions in Netscape Communicator 4.0 had to compete with a different implementation of the same idea by Microsoft in Internet Explorer 4.0. The W3C standardized on a solution which is closer to Microsoft's implementation and Netscape was left out in the cold with a non standard conformant browser.

[31]   As is true with the CSS2-standard for formatting webpages, which is still only partially supported by both Internet Explorer and Mozilla Firefox. In general, many of the W3C-standards are in fact not supported by the current generation of web-browsers.

[32]   As seen in the Ansi C++-standard which is supported in different ways by different compilers.

[33]   See e.g. the reaction of the W3C to the recent lawsuit by Eolas against Microsoft < http://www.w3.org/2003/09/pag>; however see below for the possibility of compulsory licenses under the essential-facilities-doctrine of anti-trust-law.

[34]   This is the reason why neither Corel's Wordperfect nor Sun Microsystem's StarOffice or OpenOffice product can edit Microsoft Word-documents "perfectly" – i.e. without any changes in formatting or loss of other information.

[35]   Howard A. Shelanski, J. Gregory Sidak, *Antitrust Divestiture in Network Industries*, 68 U. Chi. L. Rev. 1 (2001), 37, 66-70; Amitai Aviram, *Regulation by Networks*, 2003 B.Y.U.L. Rev. 1179 (2003), 1231; Bruce Abramson, *Promoting Innovation in the Software Industry: A First Principles Approach to Intellectual Property Reform*, 8 B.U. J. SCI. & TECH. L. 75 (2002), 143-149.

customer choice, it helps mitigating network externalities which bind users to a specific operating system (such as Microsoft Windows on client computers or a specific Unix-version on servers).

Furthermore, unlike most other products, software as such is **highly abstract.** On the one hand, any number of previously separate features may be merged into a single product. On the other hand, any given product can be split into numerous smaller components. For example, Microsoft Word is, in essence, hundreds of potentially separate products merged into one – only the need for standardization (and thereby lower distribution and consumer transaction costs) drives software developers to include a common denominator of features.[36] Selling powerful combination products is highly efficient due to the lack of physical restraints and the extremely low marginal costs of adding functions. This is a different situation from that of, say, a car, where the essential elements (e.g. wheels and engine) are easily distinguished from unnecessary, optional extras (e.g. car radio). In fact, (independent) software manufacturers tend to separately offer "better" replacements for many of the built-in-components and for the provider of the core platform to integrate formerly separate features. The network externalities of a comprehensive platform have made this a common practice.[37] It allows third parties to build upon the included elements or allow their replacements to interact with other add-ons. However, if the newly integrated features fully satisfy consumer demand, competitors who base their business on providing better alternatives might be driven out of the market. This practice of extending the platform by integrating formerly separate elements makes the "product" as such almost impossible to define and makes it hard to detect illegal "bundling" by monopolists.

Finally the software market is characterized by its **rapid innovation**. Since product cycles hardly last longer than a few months market participants were traditionally required to recover costs within a few months or less. Investors would not provide capital to firms unless they were likely to recoup the investment by achieving a significant market share in exceedingly short timeframes.[38] Consequently (in the absence of a product "revolution") new entrants will generally avoid the risk of entering any market consisting of complex and/or costly products. This is particularly true with respect to operating systems that require a plethora of interfaces and interactions with user-level applications. In these markets, new entrance into the market is rare.

---

[36]   E.g., the ability to do mail-merges, to use drawing tools and tables, the option to use reviewing and outlining features and even the ability to do basic formatting are unnecessary for many users.

[37]   See United States v. Microsoft Corp., 253 F.3d 34, 95 (D.C. Cir., 2001) noting that "Our reading of the record suggests merely that integration of new functionality into platform software is a common practice…"

[38]   However, one must keep in mind that incumbent competitors can always reduce their prices to a price level that is close to zero, since this price level is their production cost for an additional unit of the software. And, incumbents may do so, since it is better to regain a little than nothing of one's investment.

Consequently, competition and innovation is merely exercised by few incumbents, and if one of these firms fails, no new entrant takes over its functions. It also means that regulators risk coming to late with regulatory measures, because rapid innovation may render certain strategy useless faster than regulators may adopt them.

These five characteristics (natural monopoly structure, public goods aspect, network externalities, abstraction and rapid innovation) *could* drive the standard software market towards further market concentration, and – eventually - monopolization. Whether this assumption is, in fact, true, is very contentious,[39] has been sufficiently discussed over the last decade, and cannot be decided at certainty neither here nor elsewhere. With regard to this question, time will eventually tell who is right or wrong. However, what would happen if the standard software market showed, in fact, an inexorable tendency towards concentration? On the long run, in the absence of regulatory interference, only one supplier controlling the market standard for any given type of standard software would persist. Smaller complementary products developed by formerly independent software firms would eventually be integrated into larger core platforms.

In the following, we will analyze whether there are legal factors already in place that mitigate such (hypothetical?) tendency towards monopolization. Further, we develop the consequences of such a scenario, as well as a possible regulatory response to the threats provided by further market concentration.

In the following, we only consider the market for standard software. These markets are distinguished from markets for highly customized, individual programs catered to a specific purpose or user. These customized programs usually serve a very tight market. In this field of customized software, service elements (like maintenance and support or individualization and customization or personalized design and development) often prevail. Further, the shared elements of the platform are limited, thereby reducing the entry-costs, whereas the marginal costs are high as any sale of individualized products requires a significant re-investment of human resources. Insofar customized software is not subject to the natural monopoly effect

---

[39]   See, on the one hand, for example, Microsoft's expert Richard Schmalensee, *Antitrust Issues in Schumpeterian Industries*, 90 Am. Econ. Rev. 192 (2000), arguing that these effects would spur, rather than mitigate competition *and* innovation *and* consumer welfare, and, on the other hand, the U.S. Department of Justice's expert Franklin M. Fisher, *Antitrust and Innovative Industries*, 69 Antitrust L.J. 559 (2000-2001), essentially positing that the opposite is true.

outlined above.[40] The same is true with respect to network effects. Due to its different characteristics, the markets for customized software products are not examined in this paper.

III. Mitigating Frameworks

Since persistent (in contrast to temporary) monopolies supposedly hamper product innovation and harm consumer welfare,[41] the law generally tries to destroy long-lasting monopolies in their infancy. To that end, two regimes significantly influence the allocation of market power in the software market – intellectual property laws and antitrust law. Whether these regimes mitigate the market's trajectory towards monopolization if, in fact, the economic characteristics of the standard software market push it towards monopolization, deserves closer examination.

A. Intellectual Property Law

The famous Open Source activist *Richard Stallman* (who e.g. recently fought devotedly against software patents in Europe[42]) has a dream: Everybody should be able to run, modify, redistribute and improve programs as soon as he or she receives a copy of a software product.[43] This Open Source dream would be perfectly realized by removing the *property* characteristic from what we understand as intellectual property.

Undeniably there would be less of a basis for monopolies in the software market in the absence of exclusive rights[44]. In a world without *any* legal protection of software, however, developers might move into an unnecessary and wasteful arms-race for effective protection by technical measures and try to protect their positions by restrictive contractual provisions. This situation seems undesirable in the long-run and

---

[40]   Under these conditions, small manufacturers in numbers of implemented products and revenue which possess a minimum uniform basis are able to implement custom solutions for specific clients which can compete on equal terms with large manufacturers. Conversely size does not always constitute an advantage in this market.

[41]   According to Joseph A. Schumpeter, *Capitalism, Socialism and Democracy* (1942), at 81-106, temporary monopolies achieved through cost-reducing or product-improving innovations are the driving force of the famous "gale of creative destruction", Schumpeter's concept of dynamic competition; see also Richard A. Posner, *The Social Costs of Monopoly and Regulation*, 83 J. Pol. Econ. 807 (1975)*.

[42]   See recently Richard Stallman, Bill Gates and other communists (2005) at <http://news.com/2010-1071_3-5576230.html>, Richard Stallman, Software patents – Obstacles to software development, <http://www.cl.cam.ac.uk/~mgk25/stallman-patents.html>.

[43]   A philosophy followed by the Open Source Community organized under the banner of GNU ("GNU is not Unix") http://www.gnu.org/philosophy/free-sw.html; see recently David S. Evans, *Anne Layne-Farrar, Software Patents and Open Source: The Battle Over Intellectual Property Rights*, 9 Va. J.L. & Tech. 10 (2004).

[44]   Though interestingly Redhat holds a large majority of the Linux-market in the U.S. even though they completely rely on open source software reusable by any potential competitor.

detrimental to both consumers and competitors.[45] Thus, granting *partially* exclusive rights for software provides the manufacturers with incentives to innovate while still keeping their products broadly available and usable. This is the state of contemporary intellectual property regimes around the world.

However, exclusive rights provided by intellectual property law may also serve anti-competitive purposes, if abused. This section examines whether intellectual property law provides the necessary balancing to compensate for the potential abuse of the exclusive rights it created. We will follow the historical development, which started with trade secret law, was later strengthened by copyright and utility patents and is nowadays also supported by trademarks, design patents, and sui generis rights in databases. The parallel applicability of these partly overlapping legal regimes raises many complex issues which cannot be sufficiently discussed here. At this point, we can only note that, while a sui generis right in software - for example, a patent with a significantly shorter protection term - would be suited to deal with these inconsistencies, the current framework is widely predetermined by international treaties.[46] Thus, chances of establishing such a sui generis right worldwide are, in fact, slim.

1. Trade Secrets

Software manufacturers have relied on **trade secret protection** for the internal workings of their software since the first products were sold separately from computer hardware.[47] Since 1994 trade-secrecy is internationally guaranteed by Article 39 of the Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPs).[48]

Despite the fact that "real" intellectual property rights have recently been the focus of attention, trade secrets remain the most effective means for software protection. As a measure of self-help they generally avoid the risks and costs of legal enforcement. Artificially obfuscating source code, protocols and file-formats prevents third-parties from creating substitutable replacements for vendor-specific software products or illegal copies at least as well as court injunctions. While not granting exclusive property rights in the traditional sense, trade-secret law greatly reduces the risk of employing these methods. As long as secrecy is upheld, disclosure

---

[45] See Douglas Lichtman, *How the Law Responds to Self-Help*, U Chicago Law & Economics, Olin Working Paper No. 232 (2004), at 25-27; but see also Ariel Katz, *supra* note 14.

[46] For details see e.g. Viva R. Moffat, *Mutant Copyrights and Backdoor Patents: The Problem of Overlapping Intellectual Property Protection*, 19 Berkeley Tech. L.J. 1473 (2004).

[47] See MacGrady, *Protection of Computer Software - An Update and Practical Synthesis*, 20 Hous. L. Rev. 1033, 1045 (1983).

[48] See e.g. § 16 UWG in Germany or the Uniform Trade Secrecy Act (adopted by more than 40 states) in the US.

in a manner "contrary to honest commercial practices" (or "misappropriation") will **perpetually** give rise to both civil and criminal sanctions.

Trade secrets are not subject to the formal **limitations** (especially regarding the term of protection) of patent or copyright law.[49] Even the few existing restrictions were significantly reduced over time: Trade secret protection traditionally did not prevent others from patenting the same process and thereby enjoining any former users (including the developer of the trade secret) from further employing the process in the future. This risk was eliminated by the introduction of "first user rights" that allow anyone who used a patented process prior to application to continue that use even though someone else later acquired a patent on it.[50] Similarly, the legality of "reverse-engineering" (i.e. analyzing and re-implementing) features of publicly available software products[51] has been significantly limited by recent American legislation,[52] which coincides with a restrictive practice employed by the courts.[53] For example, Microsoft successfully enjoined Stac Electronics Co. from selling their only product (a disk-compression add-on to MS-DOS) because Stac had (arguably) used undocumented (and thus "secret") interfaces to MS-DOS. This injunction effectively cut Stac off from their revenue sources and provided Microsoft with sufficient leverage to reach a settlement in a patent suit brought by Stac.[54] Most recently, the Digital Millennium Copyright Act (DMCA)[55] introduced § 1201 into the U.S. Copyright Act. The new provision prohibits circumvention of any "technological measure that effectively controls access to a work protected under this title". It effectively eliminates the legality of reverse-engineering as long as some measure of protection is applied.[56] Since secrecy is generally based on

---

[49]   See also Daniel W. Park, Trade Secrets, *The First Amendment, and Patent Law: A Collision on the Information Superhighway*, 10 Stan. J.L. Bus. & Fin. 46 (2004) on the advantages of trade secrets compared to other rights.

[50]   See James R. Barney, *The Prior User Defense: A Reprieve for Trade Secret Owners or a Disaster for the Patent Law?*, 82 J. Pat. & Trademark Off. Soc'y 261(2000).

[51]   Unif. Trade Secrets Act 1 cmt. 2 (amended 1995), 14 U.L.A. 433, 438 (1990); see also Greg Weiner, *Reverse Engineering as a Method of Achieving Compatibility in the Computer Industry*, 6 U. Balt. Intell. Prop. L.J. 1 (1997).

[52]   See the Economic Espionage Act of 1996 (the "EEA"), Pub. L. No. 104-294, 110 Stat. 3488 (codified at 18 U.S.C. 1831-1839).

[53]   Lawrence D. Graham & Richard O. Zerbe, Jr., *Economically Efficient Treatment of Computer Software: Reverse Engineering, Protection, and Disclosure*, 22 Rutgers Computer & Tech. L.J. 61, 125 (1996).

[54]   By effectively cutting Stac off from its revenue source, Microsoft gained sufficient leverage to reach a settlement in a patent suit brought by Stac. Jim Carlton, *Microsoft, Stac End Battle With Pact, A 'Win-Win' Cross-Licensing Agreement*, Wall St. J., June 22, 1994, at B8; see infra III.A.3. for more on the Stac v. Microsoft-Case.

[55]   Pub. L. No. 105-304, 101, 112 Stat. 2860 (1998) (codified at 17 U.S.C.).

[56]   See Robert C. Denicola, *Fair's Fair: An Argument For Mandatory Disclosure Of Technological Protection Measures*, 11 Mich. Telecomm. Tech. L. Rev. 1 (2004).

obfuscating, i.e. hiding the internal processes of software by an "effective" measure a broad interpretation might lead to absolute protection. However, the courts are still careful in applying this potentially far-reaching provision.[57]

The only significant danger remaining to the user of trade secrets lies in the secret becoming *widely* public: As soon as the information is out, nothing can stop it from spreading.[58] This is also the reason for the general incompatibility of patents and trade secrets: Since publication of a patent application essentially lifts the veil of secrecy, an innovation will not be protected at all if the application is rejected. While applications were traditionally not published in the U.S., a 1999 amendment to the Patent Act requires publication within 18 months after filing (like the European Patent Convention).[59] This risk of completely losing any kind of protection might prevent software manufacturers from seeking patent protection at all.[60]

This seemingly limitless scope of trade secret protection prompted the European Commission to require Microsoft to disclose certain secret interfaces used in its software products to exchange data over a network. The Commission maintained that the "unilateral business decisions to keep things secret" does not lead to a "presumption of legitimacy".[61] Instead, it required balancing of these business decisions with the public interests at stake on a case-by-case basis. More specifically, it looked at the limitations of copyright in software and transferred the reasoning behind these limitations to trade secret law.[62] The settlement between Microsoft and the Antitrust Division of the U.S. Department of Justice in 2002 similarly required the disclosure of interfaces and other information to allow interoperability of Windows with middleware as well as

---

[57]   Chamberlain Group, Inc. v. Skylink Techs., Inc., 381 F.3d 1178 (Fed. Cir., 2004) held the provision inapplicable to the "circumvention" of a special protection mechanism by a company producing generic garage-door-openers; Lexmark Int'l, Inc. v. Static Control Components, Inc., 387 F.3d 522, 547 (6th Cir., 2004) held the provisions inapplicable to remanufactured printer cartridges.

[58]   See DVD Copy Control Ass'n v. Bunner, 116 Cal. App. 4th 241 (2004) at 253, holding that because of the "widespread publicity" publication of the DeCSS-code could not be effectively prevented, even though it was originally gained by "misappropriation".

[59]   See 35 U.S.C. § 122(b) (1) (A) – but also note the exceptions especially regarding applications for patents limited to the U.S.

[60]   Andrew Beckerman-Rodau, *The Choice Between Patent Protection and Trade Secret Protection: A Legal and Business Decision*, 84 J. Pat. & Trademark Off. Soc'y 371 at footnote 89.

[61]   European Court of First Instance, CFI-Order, *supra* note 6, at marginal 182.

[62]   id.

any protocol used to communicate with Microsoft's server operating systems.[63] However, the exact limits of trade secret protection still remain unclear both in Europe and the United States.

2. Copyright

 Both the United States[64] and the European Union[65] introduced copyright protection of software after intensive lobbying by the software-industry. This system was later globally affirmed by Article 4 of the WIPO Copyright Treaty (1996) and Article 10 (1) of TRIPs (1994). Unlike patent protection, copyright can exist parallel to trade secret protection as it does not require disclosure of the inner workings of a software product (the so-called 'source code') and allows developers to limit or prevent direct reproduction, distribution and creation of derivative works of their products.

 Copyright merely protects the specific "expression" of a generally unprotected "idea".[66] In applying this doctrine to software, U.S. courts have consistently limited the scope of protection, especially regarding the visible user interface.[67] A similar rule is expressly stated in the EC Software Directive.[68] Moreover, copyright *in software* is subject to several important limitations which, in theory, should effectively prevent monopolization. These include the right to decompile in order to find common interfaces and the right to examine the program to detect the underlying unprotected ideas and principles.[69] Additionally, recent cases of the European Court of Justice (*Magill* and *IMS Health*)[70] imposed further limitations specifically on *copyright*.

---

[63]   Final Judgment, United States v. Microsoft Corp., CA No. 98-1232 (CKK), filed Nov. 12, 2002 (D.D.C. 2002)) at III.D.

[64]   By amending the copyright act in 1980, see e.g. Central Point Software v. Nugent, 903 F. Supp. 1057, 1060 (E.D. Tex. 1995) ("That software programs are copyrightable material is beyond dispute").

[65]   EC Directive on the Legal Protection of Computer Programs of 14 May 1991 (91/250/EEC), 1991 O.J. L 122.

[66]   Art. 9 (2) TRIPS, see Baker v. Selden, 101 U.S. 99 (1880); Article 1 (2) EC Directive on the Legal Protection of Computer Programs of 14 May 1991 (91/250/EEC); with the traditional distinction between "ideas" and "expression" applied to software Computer Assocs. Int'l v. Altari, Inc., 982 F.2d 693 (2d Cir., 1992) at p. 703.

[67]   See Lotus Dev. Corp. v. Borland Int'l, 49 F.3d 807 (1st Cir., 1995); Apple Computer v. Microsoft Corp., 35 F.3d 1435 (9th Cir., 1994); Nicolas P. Terry, *GUI Wars: The Windows Litigation and the Continuing Decline of Look and Feel*, 47 Ark. L. Rev. 93 (1994); Linda Skon, Copyright Protection Of Computer User Interfaces: "Creative Ferment" In The Courts, 27 Ariz. St. L.J. 1063 (1995).

[68]   See Article 1 (2), 2nd sentence of the EC-Software-directive: "*Ideas and principles which underlie any element of a computer program, including those which underlie its interfaces, are not protected by copyright under this Directive*".

[69]   Article 6 EC Directive on the Legal Protection of Computer Programs of 14 May 1991 (91/250/EEC), as implemented in the laws of the member states, e.g. § 69 f. UrhG; Jaap H. Spoor, Copyright Protection and Reverse Engineering of Software: Implementation and Effects of the EC Directive, 19 U. Dayton L. Rev. 1063, 1077 (1994); Karen E. Georgenson, Reverse Engineering Of Copyrighted Software: Fair Use Or Misuse?, 5 Alb. L.J. Sci. & Tech. 291 (1996); Joe L. Gage, Jr., Copyright Law and Reverse Engineering: Have Recent Decisions Taken the Fair Out of Use?, 46 Baylor L. Rev. 183 (1994);

While these limitations significantly mitigate the anti-competitive effects of copyright protection vis-à-vis competitors, its core relevancy lies in the relationship to software-users and suppliers of add-on products: Copyright is the basis for the end-user license agreements (EULAs) between the developer and the users of any software title. Therefore copyright is the "life blood" of the Open Source community, allowing it to enforce its doctrine of openness and sharing upon any user of software published under a "Free software license".[71] Conversely, copyright protection is also the main threat to the Open Source operating system Linux in a suit brought by the SCO Group against IBM. In this suit, SCO claims that IBM contributed copyright-protected code to Linux, thereby making all Linux users copyright infringers.[72] In addition, copyright protects the software industry from a second-hand market by contractual clauses in license agreements that prevent the use or redistribution of applications (so called "OEM"-clauses). For example, Microsoft limits the use of some

---

see also Sega Enters. v. Accolade, Inc., 977 F.2d 1510 (9th Cir., 1992); Sony Computer Entertainment, Inc. v. Connectix Corp., 203 F.3d 596 (9th Cir., 2000).

[70]   See infra III.B.1.c.

[71]   See e.g. the GNU General Public License (GPL) at http://www.gnu.org/licenses/gpl.html: "2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions: … b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License. … These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it. …

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following: a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or, b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or, c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.) The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable. If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. …"

[72]   See Kerry D. Goettsch, SCO Group v. IBM: The future of Open-Source Software, 2003 U. Ill. J.L. Tech. & Pol'y 581 (2003).

of its applications to Windows; Apple limits its MacOS-operating system to Apple hardware. Licensing restrictions also prevented Microsoft from distributing a version of Sun's cross-platform JAVA that features Windows-specific, incompatible extensions. [73] Finally, copyright protects the right-owner from "drastic alterations" of his product made by other manufacturers.[74]

Copyright protects both the binary code (i.e. the software a user runs to achieve a specific result) and the source code (i.e. the editable and modifiable original programming effort, as the 'heart' of any software product which is needed to add new features and fix bugs). However, copyright protection for the source-code is largely irrelevant in practice[75] as the source code is generally subject to the (self-enforcing) protection of trade secrecy outlined above. Without access to the source code, competitors are unable to improve or modify another's work and disguise it as their own work. Under these conditions, illegal copies (termed "pirate copies" or "warez") are easy to detect. When the source-code of the original product is available (as required for Open Source products) it is hard to prove that a program (implementing some modifications) that is only available in binary form was based on a specific piece of the source code. Enforcing copyright for source code is therefore expensive and tedious, whereas the burden of proof is far lower for copyright-infringement in binaries. Still about a third of the standard software in use has been acquired in violation of copyright law (e.g. sharing of software among friends and family). This lead the industry to implement technical measures to prevent copying (e.g. the "product activation" of recent versions of Microsoft Windows and Microsoft Office) - something that (again) would not be possible (or easily removed) if the source-code was generally available.

Thus, copyright protection of mass-market standard software (like Microsoft Windows or Microsoft Office) does not mitigate the tendency towards concentration which is discussed in this article.

---

[73]   Sun Microsystems, Inc. v. Micorsoft Corp. (In re Microsoft Corp. Antitrust Litig.), 333 F.3d 517 (4th Cir., 2003).

[74]   This measure explains an important exception in the settlement of the American Microsoft antitrust case in 2002, allowing Microsoft to restrict the automatic launching of non-Microsoft-Middleware unless the replacement provides no user interface or an interface resembling the equivalent MS product. See: Final Judgment, United States v. Microsoft Corp., CA No. 98-1232 (CKK), filed Nov. 12, 2002 (D.D.C. 2002)) [hereinafter United States' & Settling States' Settlement] (judgment adopting plaintiffs' Third Revised Proposed Final Judgment), available at http://www.usdoj.gov/atr/cases/f200400/200457.htm at III.C.3.

[75]   Mann, *supra* note 21, at pp. 1012-20; but see the competition-oriented theory developed by Ariel Katz, *supra* note 14.

3. (Utility) Patents

As understood today, copyright in software offers strong protection against reproduction and distribution of binary end-user products without the manufacturer's consent, but little protection against re-implementation of innovative ideas by competitors in source-code. The desire to create protection against re-implementations required the industry to petition for patent protection. A patent grants protection not only against direct reproduction of an expression (like copyright) but also prohibits parallel inventions. The scope of a patent is further muddied by the doctrine of equivalency which expands infringement beyond the literal scope of the patent writ. Patents, however, do not replace copyright protection but complement it. The relevance of software patents gained extreme visibility in the ongoing lobbying battle in Europe between large software firms (who see Europe suffering from a competitive disadvantage relative to the U.S.) and the Open Source community (who believe that software patents will spell out their demise).[76]

On March 3, 1981 the United States Supreme Court accepted software as a patentable subject matter under § 101 of the *U.S. Patent Act*.[77] A few years later the Court of Appeals for the Federal Circuit extended patent protection to business methods.[78] These low statutory requirements have lead to a dense web of U.S. patents, often termed the "patent thicket".[79]

In contrast, Article 52 (2) (c) of the European Patent Convention [EPC][80] explicitly prohibits patents on "programs for computers", "to the extent to which a European patent application or European patent relates to such subject-matter or activities as such" (EPC Article 52 (3)). Patents on business methods are similarly prohibited.[81] The European Patent Office grants patents on software products only if the applicant

---

[76]   See the activist-sites at <http://swpat.ffii.org/>; <http://www.nosoftwarepatents.com/>; <http://www.softwarepatents.co.uk/>.

[77]   Diamond v. Diehr, 450 U.S. 175 (1981); previously see e.g. Gottschalk v. Benson, 409 U.S. 63 (U.S. 1972).

[78]   State St. Bank & Trust Co. v. Signature Fin. Group, 149 F.3d 1368 (Fed. Cir., 1998); cert. denied State St. Bank & Trust Co. v. Signature Fin. Group, Inc., 525 U.S. 1093 (1999).

[79]   See Michael J. Meurer Business Method Patents and Patent Floods, 8 Wash. U. J.L. & Pol'y 309 (2002); Oren Bar-Gill and Gideon Parchomovsky, The value of giving away secrets, 89 Va. L. Rev. 1857 (2003); Lawrence Lessig, The Future of Ideas: The Fate of the Commons in a Connected World (2001) at pp. 205-15; Mann, *supra* note 21, at footnote 5 estimates **10,000 new software patents** a year; see also the testimony given at the FTC hearings in Federal Trade Commission, To Promote Innovation: The Proper Balance of Competition and Patent Law and Policy (2003) at pp. 52 seq.

[80]   The EPC is not EC law. Rather, it is a treaty among most European states, including Switzerland as well as some non-European states, such as Turkey. It is important to note, though, that the Convention has no effect at all on the national patents granted by individual states according to their non harmonized national patent law, unless if it was granted in violation of the European rules, see EPC Article 138.

[81]   EPC Art. 52 (2) (c) excludes "schemes, rules and methods for performing mental acts, playing games or doing business".

proves that the potential invention constitutes a "*technical*" contribution beyond the fact that it runs on a "technical" computer.[82] However, their decisions remain very hard to categorize or predict. Insofar Europe is still a risky territory for software developers seeking patent protection. This might be clarified with the adoption of the (heavily debated) proposal for a EC Directive for the Protection of Software Implemented Inventions.[83] While the Directive cannot legally change the EPC as it is only binding on the EU Member States and not on other states that are only members of the EPC, it does have a strong de facto influence.

In theory, patent law furthers competition.[84] Patent law requires the inventor to publish technical inventions and forces patents into the public domain after 20 years. Without the exclusivity granted by patent law, inventions may remain secret. Thus, patents are supposed to spread new technologies and prevent the development or persistence of perpetual monopolies that would be otherwise sought to be protected by secrecy. It has often been argued that 20 years is too long with respect to the rapid pace of innovation in the software market.[85] The pace of innovation in other markets, however, (especially pharmaceuticals) is not significantly slower. The long duration will often be compensated for by the high initial requirements to acquire a patent: A patent is not granted for a product as such, but only for a specific new and non-obvious innovative process embedded in the product[86] – a fact that significantly weakens the efficiency of patent protection. In addition, patent claims are often specific enough to allow competitors to invent around them

---

[82]   John T. Soma, Kurt Leyendecker, Steven L. Webb, Software patents: A U.S. and E.U. comparison, 8 U. Balt. Intell. Prop. L.J. 1 (2000); Ann Marie Rizzo, The Aftermath of State Street Bank & Trust v. Signature Financial Group: Effects of United States Electronic Commerce Business Method Patentability on International legal and economic systems, 50 DePaul L. Rev. 313 at pp. 347-353 (2000); see more recently Heather Forrest, Europe: Open Market... Open Source?, 2003 Duke L. & Tech. Rev. 28 (2003)

[83]   COM(2002) 92 final (2002) at http://europa.eu.int/comm/internal_market/en/indprop/comp/com02-92en.pdf, see also the Council's Common Position (2004) at http://register.consilium.eu.int/pdf/en/04/st09/st09713.en04.pdf; pursuant to recital 3, the directive aims at harmonizing computer implemented inventions through Europe without extending the patentability on computer programs as such. See Forrest, Ibid.

[84]   Nuno Pires de Carvalho, The Primary Function of Patents 2001 U. Ill. J.L. Tech. & Pol'y 25 (2001); Simone A. Rose, Patent "Monopolyphobia": A Means of Extinguishing the Fountainhead?, 49 Case W. Res. 509 (1999); see also Julie E. Cohen, Mark A. Lemley, Patent Scope and Innovation in the Software Industry, 89 Calif. L. Rev. 1 (2001).

[85]   See e.g. Mark Aaron Paley, A Model Software Petite Patent Act, 12 Santa Clara Computer & High Tech. L.J. 301 (1996).

[86]   For example the patent in dispute in Eolas v. Microsoft (Patent No. 5,838,906) is not a patent for a web-browser but for a "distributed hyper media method for automatically invoking external application providing interaction and display of embedded objects within a hypermedia document". Microsoft attacked it for both anticipation (lack of novelty) and non-obviousness. The Court of Appeals for the Federal Circuit (Eolas Techs., Inc. v. Microsoft Corp., 399 F.3d 1325, 1332 (Fed. Cir., 2005)) allowed Microsoft to go forward with these defenses, thereby partially vacating the District Courts decision (Eolas Techs. v. Microsoft Corp., 2004 U.S. Dist. LEXIS 522 (D. Ill., 2004)). See also the University of California's FAQ at <http://www.ucop.edu/news/archives/2003/aug11art1qanda.htm>. This fact also seemed to be relevant in *Kodak v. Image Technical*, ib., 125 F.3d 1195.

and find unprotected alternatives.[87] Finally competitors might acquire a patent on improvements to the original invention, thereby theoretically excluding the original patent holder from the use of modern modifications of its own product. In practice, such "improvement patents" usually cause extensive cross-licensing agreements among the parties.

Furthermore, the patent system is designed to reward new market entrants for their inventiveness and protect them from theft of their ideas and is thereby meant to protect small competitors.[88] In theory, patent law opens competition by splitting the market among many smaller patent holders and thereby constrains overpowering monopolies. A patent case from the early 1990s illustrates this concept:[89] In the late 1980s Microsoft's market leading MS-DOS 5.0 was threatened when Digital Research began selling DR-DOS 6.0 which included technology to "double" hard-disk space by means of data-compression. Microsoft quickly entered into negotiations with Stac, the market leader in compression-technology at that time. Microsoft was about 100 times the size of Stac even then and it was clear that integration of compression-technology into MS-DOS would kill the market for Stac's separate add-on product, "Stacker". During negotiations the firms shared source-code, but eventually failed to reach an agreement. When Microsoft nevertheless included compression features ("DoubleSpace") in MS-DOS 6.0, Stac sued for patent infringement. A jury enjoined Microsoft from selling the infringing DOS-version (causing Microsoft to release MS-DOS 6.21 without the infringing compression feature) and required Microsoft to pay $ 120 million in damages.[90] However, the court also issued an injunction against Stac for violating Microsoft's trade secrets – thereby stopping sales of its only product "Stacker" – and damages in the amount of $ 13.6 million. Unlike Microsoft, Stac had no other sources of revenue and could not adapt its product. Thus, Stac could not afford long-term litigation. Eventually both companies dropped their claims and participated in a broad cross-licensing agreement. Microsoft paid Stac $ 43 million in royalties. In addition, Microsoft acquired $ 39.9 million in Stac preferred

---

[87]   Lemley & McGowan, *Legal Implications*, *supra* note 17, at 523-530.

[88]   See the results of the extensive study by Mann, *supra* note 21.

[89]   Stac Electronics Co. v. Microsoft Corp - No. 93-0413 (S.D. Cal. decided Feb. 23, 1994), appeal dismissed per stipulation, No. 941349, 1994 U.S. App. LEXIS 18042, (Fed. Cir. July 5, 1994).

[90]   John Burgess, *Microsoft Found Guilty Of Patent Infringement, Software Giant Ordered to Pay $ 120 Million*, Wash. Post, Feb. 24, 1994, at D11, D14; Amy Harmon, *Microsoft Loses Patent Lawsuit, Must Pay Rival $ 120 Million*, Los Angeles Times, Feb. 24, 1994, at D1, D5.

non-voting stock.[91] This (partial) "success-story" of patent law is, however, an exception. Only two more recent cases show that smaller outside-inventors can prevail against larger incumbents.[92]

Instead, it is reasonable to assume that patent law is abused as a means to further concentration in the market for standard software. The strong interdependency of software innovations and the costs of patent research can be used to block out new and small competitors. As new inventions tend to require use of previous patents (for backward-compatibility reasons) new firms that do not have access to patented rights may be kept out of the market. In addition, the broad claim language traditionally used in patent applications[93] tends to hinder alternative circumventive approaches in the software industry more than in other areas of technology. Consequently, new competitors depend on licenses from incumbent patent holders in order to enter the market. Incumbents can effectively protect their oligopolies by collectively denying licenses for essential patented interfaces. Thereby new entrants cannot offer backward-compatibility and cannot effectively compete due to a lack of exchangeable data. The leverage provided by these exclusionary rights allows strong market participants with valuable patents to require extremely favorable terms in negotiations, usually also including cross-licensing, giving them access to all inventions made by the smaller market participant.[94]

In order to resolve the dangers of abusive licensing terms, U.S. courts have developed the doctrine of "patent misuse" to keep patent protection within reasonable limits. Patent misuse occurs when the patentee has "impermissibly broadened the 'physical or temporal scope' of the patent grant with anti-competitive effect."[95] However the strength of that doctrine was recently limited by 35 U.S.C § 271 (d) which provides that a patentee is not guilty of misuse for deriving revenue from or licensing another to engage in acts that would constitute contributory infringement in the absence of the patentee's consent; enforcing the patent; refusing to license the patent or tying a license to acquisition of another license or product unless the patentee

---

[91]   Jim Carlton, *Microsoft, Stac End Battle With Pact, A 'Win-Win' Cross-Licensing Agreement*, Wall St. J., June 22, 1994, at B8

[92]   Eolas Techs. v. Microsoft Corp., 2004 U.S. Dist. LEXIS 522 (D. Ill., 2004) granted upstart Eolas Techs. $ 520,562,280.00 in damages for Internet-explorer-sales from 17 November 1998 through 30 September 2001 plus additional damages for the following years and enjoined Microsoft from further distributing Internet Explorer – however that decision was partially vacated and remanded in Eolas Techs., Inc. v. Microsoft Corp., 399 F.3d 1325 (Fed. Cir., 2005). In another case, a federal jury determined that Sun Microsystems, Inc's. JAVA violated patents held by (non-software-company) Eastman Kodak Co. – the parties subsequently settled for $ 92 Million in cash after Kodak threatened to demand $ 1 billion, see <http://www.eweek.com/article2/0,1759,1668324,00.asp>.

[93]   See Robert P. Merges, Richard R. Nelson, *On the complex Economics of Patent Scope*, 90 Colum. L. Rev. 839 (1990).

[94]   *.

[95]   Blonder-Tongue Lab., Inc. v. University of Ill. Found., 402 U.S. 313, 343 (1971).

possesses market power. Insofar the patent misuse defense nowadays primarily reflects the restrictions imposed on patents by antitrust law.

Assuming that there is (still) a problem, besides the "patent abuse" doctrine and similar devices, three measures could improve the current patent regime and ensure the existence of incentives to invent in a competitive market. The first is to impose **higher standards on patentability of software**. This includes requiring more specific and precise disclosure of inventions, and a greater distance from the previous state of the art as required for "non-obviousness" under the patent laws. However, instead of imposing higher standards on the patent offices' already overworked bureaucracies, legislatures should facilitate third party interventions in patent prosecution. While the European Patent Office allows anybody to challenge a patent in a quick and inexpensive opposition procedure, third party rights are very limited in the U.S. (no appeal for challenger, preclusion of argument in later trials, etc.).

Second, competition in the software-industry requires more efficient **compulsory licensing**. While this has traditionally been frowned upon and rarely employed, it evidently lowers barriers to market entry. For example, the German rules for compulsory licensing were recently amended[96]. Compulsory licensing generally requires proof of a supervening public interest. In contrast, the amended statute additionally provides for compulsory licenses whenever a third party needs a license to an older patent in order to make use of their own patented improvement if this improvement represents a major technological advance of high economic importance as compared to the older patent. Such changes might further the practical use of compulsory licensing and thereby protect smaller competitors. While the legislature tries to create broader rules for compulsory licensing, the courts reach similar results based on antitrust law. In a mixed antitrust /patent law constellation, the antitrust panel of the German Federal Supreme Court[97] generally accepted *compulsory licenses* based on the antitrust law's[98] anti-discrimination provisions for firms in a dominant position. While acknowledging that a patent owner, even in a dominant market position, is generally free to deny licenses, the senate (citing the European Court of Justice's decisions on copyright as essential facilities[99]) held that

---

[96]   German Patent Act, § 24 (2) as amended by *BGBl. I/2005, p. 146 seq..*

[97]   BGH, decision of 13 July 2004, KRZ 40/02 – Spuntfass.

[98]   Like the United States Germany employs the terminology of "Kartellrecht", i.e. "Antitrust Law", as "competition law" is generally understood as a reference to "unfair competition".

[99]   See ECJ Joined Cases C-241/91 & C-242/91 P, 1995 E.C.R. I- 743, 4 C.M.L.R. 718, *Radio Telefis Eireann (RTE) and Independent Television Publications Ltd. (ITP) v. Commission* (1995) [Magill] and ECJ Case C-418/01, *IMS Health GmbH & Co. OHG v. NDC Health GmbH & Co. KG* (2004) [IMS]; see also Donna M. Gitter, The Conflict In The European Community

significant challenges to free competition *might* require compulsory licenses. Especially when industry standards or official regulations require use of the patent for entering subsequent markets, strict scrutiny applies to the licensing practices of the patent holder. It is not necessary for the patent holder to endorse or even influence this standard as it is sufficient that he or she benefited from possessing the patent. Finally the court emphasized that antitrust law -based licensing obligations are not precluded by the compulsory license scheme provided for in the German *Patent* Act – both systems exist in parallel.

Third, **patent enforcement** is a crucial factor. Whereas a patent can be enforced inexpensively and quickly in Germany, the jury-system imposes a higher burden on competitors in the United States or the United Kingdom. In general, only larger incumbents have the resources to get the optimum representation and expert evidence necessary to convince a jury of their case.[100] Furthermore, high litigation costs and lack of reimbursement rules make patent litigation an undesirable option for new market entrants.

In sum patent law currently fails to balance the impact of market forces in the software market to the extent it might do under ideal circumstances. While some developments might mitigate or even eliminate the insufficiencies currently experienced, we see no jurisdiction consistently pursuing an anti-concentration policy through means of patent law.

4. Trademarks, Design Patents and Databases

Ever seeking to expand their legal standing, software manufacturers try to complement trade secret, copyright and utility patent protection of their products with trademark law, design patents and the European Union's sui generis right in databases. [101]

(Registered) Trademarks[102] are identifiers for the source of a product (and an associated quality-level) – thereby reducing search costs for consumers. Even though this was not a classical trademark case, Microsoft referred to its trademarks as one of many objections to the Commission's requirement to unbundled its Media

---

[ ] Between Competition Law And Intellectual Property Rights: A Call for Legislative Clarification of the Essential Facilities Doctrine, 40 Am. Bus. L.J. 217, 300 (2003); Donna M. Gitter, Strong Medicine For Competition Ills: The Judgment Of The European Court Of Justice In The IMS Health Action and its Implications for Microsoft Corporation, 15 Duke J. Comp. & Int'l L. 153 (2004).

[100] *.

[101] See Michael J. Schallop, Protecting User Interfaces: Not As Easy As 1-2-3, 45 Emory L.J. 1533 (1996); Jane M. Rolling, No Protection, No Progress For Graphical User Interfaces, 2 Marq. Intell. Prop. L. Rev. 157(1998)

[102] Protected by the Lanham Act in the U.S. and as community trademarks under the EC-Community-Trademark-Regulation or as memberstate-trademarks under their laws (which were harmonized by the EC-Trademark-Directive)

Player from the Windows operating system in the European Microsoft case. When the European Commission required Microsoft to release an "unbundled" Windows XP-version without Windows Media Player, Microsoft feared "serious injury" to its "valuable Microsoft and Windows trademarks".[103] Since consumers associated those marks with the company, it could not be required to "sell a downgraded product inconsistent with its basic design concept". The argument was based on a long tradition in Trademark law preventing dilution by "**tarnishment**". The President of the European Court of First Instance, however, denied that argument and emphasized that trademarks only guaranteed a specific *source* of the product – and there would be no consumer confusion regarding the manufacturer as even the Windows-version without Windows Media Player would be produced by Microsoft.[104]

After the failure of copyright law in protecting the look and feel of software, there has been a move towards increased protection of ornamental elements of software user interfaces by **design patents**. For example, Apple Computers was granted patents on a photo-realistic image of a wastebasket and certain animation-effects in its Mac OS X software.[105] Similarly, (registered) **trademarks** (or more precisely, trade-dress) provide a viable option for the protection of user interfaces (e.g. the icons used in Windows or Office).[106] However, such protection is limited by the "functionality" doctrine.[107] This doctrine prevents trademark protection for elements which are essential to the use or purpose of the device or affect the cost or quality of the device especially due to lack of alternative designs. It also prevents trademarks from extending to basic elements that are part of the public domain.

Finally, the **sui-generis right** in databases (not to be mistaken for the hypothetical future sui generis right in software mentioned above),[108] which was established under the EC Database Directive,[109] has had

---

103   European Court of First Instance, Order by the President of, decision T-201/04 R, 2004 ECJ CELEX LEXIS 390 at marginal 351.

104   European Court of First Instance, Order by the President of, decision T-201/04 R, 2004 ECJ CELEX LEXIS 390 at marginal 469.

105   See United States Patent No. D470,860 – User interface for computer display (Wastebasket); United States Patent No. D453,166 - User interface for a computer display (transparent cascading dialogue sheets); D457,164 - User interface for computer display ("Genie Effect"), see also <http://www.macobserver.com/editorial/2003/03/05.1.shtml> and <http://www.macobserver.com/editorial/2003/03/04.2.shtml>.

106   See Mat D. Carlson, *Intellectual Property Protection for Computer Icons: The Trademark Alternative*, 31 U.S.F.L. Rev. 433 (1997).

107   See TrafFix Devices, Inc. v. Mktg. Displays, Inc., 532 U.S. 23 (2001).

108   Lionel M. Lavenue, *Intellectual Property for the Protection of Databases,* 38 SANTA CLARA L. REV. 1, 24 & n.113 (1997); Michael J. Schallop, *Protecting User Interfaces: Not as Easy as 1-2-3,* 45 EMORY L. J. 1533, 1535 (1996).

little impact on the software industry. This is primarily due to the fact that content-providers (e.g. LexisNexis and Westlaw) are (still) separate from providers of the underlying technology (e.g. Google or Microsoft).

5. Lack of Balancing Effect

So far, this section has shown that if, in fact, the economic characteristics of the standard software market tend to push it towards monopolization,[110] intellectual property rights currently may cement such monopolies, rather than mitigating its inherent tendencies caused by its economic characteristics. This is the case, even though a narrow definition of the scope of patents, a strengthened system of private enforcement, and mandatory licensing (or mandatory disclosure with regards to trade secrets) could theoretically mitigate the economic forces that drive the standard software market towards monopolization. Yet legislatures worldwide seem either ignorant of the problem or unwilling to consistently implement those measures.

Even with those hypothetical measures in place, it is unlikely (though not impossible) that intellectual property law can provide a fully satisfactory solution for the inherent monopolistic tendencies in the standard software market. Simply put, intellectual property rights are supposed to **grant** rather than take away **exclusivity**. Intellectual property rights will therefore inevitably provide the legal backup for the existing market power of incumbent software manufacturers. In addition, the private enforcement of copyrights and patents is subject to opportunistic behavior (individuals tend to give up on a claim whenever settlements result in a better outcome for themselves individually, leaving the general problem for competition and society behind) and collective action problems: *All* manufacturers would benefit from one manufacturer challenging the scope of another's intellectual property right, while only the litigator is bearing the risk. Finally, in the absence of significant changes to intellectual property laws, costs and legal barriers to efficient protection that prevent smaller competitors from gaining a level playing field with incumbents will remain.[111] Thus, there is little hope that intellectual property rights as such assist new entrants in effectively countering the position of dominant incumbents.

---

[109]   Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases; See Lionel M. Lavenue, Database Rights and Technical Data Rights: The Expansion of Intellectual Property for the Protection of Databases, 38 Santa Clara L. Rev. 1 (1997) on the relationship to existing IP-rights.

[110]   See our working assumption above, at II.

[111]   As noted above, whereas it is harder to gain patents in computer software in the European Patent Office, US patent holder face a higher costs in enforcing their patents.

B. Antitrust Law

The lack of effective balancing measures in intellectual property law would be irrelevant if antitrust law satisfactorily countered the economic forces that drive the market for standard software towards monopolization. While anyone can enforce the limits of intellectual property law by self-help - i.e. if a limitation applies no exclusive rights remain and anybody may commit otherwise infringing acts - antitrust law necessitates involvement of state agencies and / or courts which are usually slow and tedious. However, the risk of assessing legal limitations with respect to intellectual property rights lies with the acting competitor (and not with the government or the dominant market participant): If a supposed limitation does not apply, huge damages payments might be due. Consequently, competitors preferred other measures over self-help, in particular the enforcement of antitrust laws. Thus, the rules proscribing **abuse of market power** and **restraints of trade** were tested against dominant participants in the standard software market, and attempts to expand market concentration were subjected to the **merger control regime**.

1. Anti-Competitive Conduct by Monopolists

§ 1 of the *Sherman Act[112]* and Article 81 the EC Treaty prohibit agreements in restraint of trade, such as those relating to price fixing, market divisions and bundling of unrelated products. However, such agreements can be justified by a showing of positive welfare effects under the rule of reason applied in the US[113] or under the express exemptions of Article 81 (3).[114] Further, § 2 of the *Sherman Act[115]* and Article 82 of the EC Treaty proscribe anti-competitive behavior by monopolists.[116] To distinguish exclusionary acts from competitive acts

---

[112] 15 USCS § 1 (2005).

[113] see e.g. *United States v. Columbia Steel Co.*, 334 U.S. 495, at 508 (1948); *Broadcast Music, Inc. v. Columbia Broadcasting System, Inc.*, 441 U.S. 1, at 16-24 (1979); for details on the "rule of reason" see Posner, *Antitrust Law* (University of Chicago Press, 2d ed. 2001), at 38-39.

[114] Dirk Zetzsche, *An Antitrust Analysis of the Software Industry – Comparative View*, Bell University Laboratories at the University of Toronto Innovation Law and Policy Research Lab Working Paper (5_2004), online www.innovationlaw.org/lawforum/pages/Zetzsche_csm.pdf [Zetzsche, *Software Industry*] at 14, 80, 106 et seq

[115] 15 USCS § 1 (2005).

[116] A dominant position is such that a firm or group of firms would be in a position to behave to an appreciable extent independently of its competitors, customers and ultimately of its consumers. European Court of Justice, *Hoffmann-La Roche*, 13.02.1979, case 85/76, [1979] ECR I-217; for an introductory overview, see Per Jebsen & Robert Stevens, "Assumptions, Goals, and Dominant Undertakings: The Regulation of Competition Under Article 86 of the European Union", (1996) 64 Antitrust L.J. 443.

the courts look to the efficiency of the specific conduct in question, i.e. whether or not the specific conduct has a welfare enhancing effect.[117]

In the past few years the antitrust authorities and courts have tried to apply these rules to Microsoft's almost ubiquitous operating systems. The saga began as early as 1994, when both the Antitrust Division of the U.S. Department of Justice [DoJ] and the European Commission challenged Microsoft's restrictive terms in license- and nondisclosure-agreements for MS-DOS.[118] Both investigations ended in identical consent decrees (resp. an "undertaking" in European terminology),[119] which kept the authorities at bay for only three years: In 1997 the DoJ sued Microsoft for violating the integrated products exception clause of the decree by bundling Internet Explorer and Windows. However, the D.C. Circuit held that the bundling was not prohibited by the consent decree and did not rule on the potential antitrust issues. [120] Shortly thereafter, the DoJ joined up with several states for another legal battle – the multi year trial and its settlement are legendary.[121] In June 2000 Judge Thomas Penfield Jackson ordered Microsoft to split into two separate legal entities.[122] It based its decision primarily on the attempt to monopolize the Internet Browser market in violation of the *Sherman Act* § 2 and the resulting "per se" illegality of the bundling of Internet Explorer and Windows under the *Sherman Act* § 1. The Court of Appeals for the D.C. Circuit vacated the breakup order and remanded the case. [123] While the federal government and most of the states agreed on a settlement, [124] others further pursued the case. However, Judge Colleen Kollar-Kotelly's decision was almost completely identical to the settlement[125] and an appeal by Massachusetts was dismissed en banc by the Court of Appeals for the D.C Circuit in June 2004.[126]

---

[117]   See Alan J. Meese, Price Theory, Competition, and the Rule of Reason, 2003 U. Ill. L. Rev. 77 (2003).

[118]   United States v. Microsoft Corp., 56 F.3d 1448 (D.C. Cir. 1995) [Microsoft I]; Proposed Final Judgment and Competitive Impact Statement, United States v. Microsoft, 59 Fed. Reg. 42,845, 1995-2 Trade Cas. (CCH) P50,764 (1994); Bulletin of the European Union, 1994, vol. 7/8, p. 130.

[119]   United States v. Microsoft Corp., *consent decree issued*, 1995 WL 505998 (D.C. Cir. Aug. 21, 1995); EC*.

[120]   United States v. Microsoft Corp., 147 F.3d 935 (D.C. Cir. 1998).

[121]   United States v. Microsoft Corp., 87 F. Supp. 2d 30 (D.D.C. 2000).

[122]   United States v. Microsoft Corp., 97 F. Supp. 2d 59, 64-65 (D.D.C. 2000).

[123]   United States v. Microsoft Corp., 253 F.3d 34 (D.C. Cir. 2001).

[124]   United States v. Microsoft Corp., 231 F. Supp. 2d 144 (D.D.C. 2002).

[125]   New York v. Microsoft Corp., 224 F. Supp. 2d 76 (D.D.C., 2002).

[126]   Massachusetts v. Microsoft Corp., 362 U.S. App. D.C. 152 (D.C. Cir., 2004).

The recent wave of U.S. litigation ended with a private antitrust action against Microsoft brought by a number of consumers and competitors.[127]

Most recently, the European Commission ruled in 2004 that Microsoft violated Articles 81 and 82 of the EC Treaty by including Windows Media Player in its Windows operating system and obfuscating necessary interfaces.[128] While Microsoft's preliminary motion for a stay of the Commission's order was denied by the President of the European Court of First Instance,[129] the order as such is still pending.

As constellations involving Microsoft account for examples for almost every type of anti-competitive behavior discussed in literature,[130] Microsoft cases will also provide the majority of examples used in this analysis. Specifically, these include predatory pricing, tying, some licensing practices, as well as threats, bribes and fraud.

a. Predatory Pricing

Antitrust law generally permits discounting. However "predatory pricing" by dominant market participants below average *variable* costs to exclude rivals or deter entrants, and thereby earn higher profits in the future is prohibited.[131] The predation itself can occur by different means, such as expansion in unused capacity, increases in the frequency of services and by improving quality without the price increases necessary to recover costs.[132] Furthermore, predatory pricing might result in a "predatory reputation" that deters potential competitors from entering into the market, or investors from financing businesses that intend to compete with these aggressive firms.[133] Chicago School economists, however, assume that predation is

---

[127]   In re Microsoft Corp. Antitrust Litig., 127 F. Supp. 2d 702, 704 (D. Md. 2001).

[128]   Sun Microsystems v Microsoft Corp, ECComm 1 (Comp/C-3/37.792) (Mar 24, 2004)

[129]   European Court of First Instance, CFI-Order, *supra* note 6.

[130]   The literature on the Microsoft cases is vast. Among the seminal pieces are Tim Brennan, "Do Easy Cases Make Bad Law? Antitrust Innovation or Missed Opportunities in *U.S. v. Microsoft*" (2001) G. Wash. L. Rev. 69; David Evans, Franklin Fisher, Daniel Rubenfeld & Richard Schmalensee, *Did Microsoft Harm Consumers? Two Opposing Views* (Washington, DC: AEI-Brookings Joint Center for Regulatory Studies, 2000); Liebowitz & Margolis, *Winners, Losers & Microsoft*, *supra* note 14; Michael Katz & Carl Shapiro, "Antitrust in Software Markets", in: Jeffrey Eisenach & Thomas Lenard (eds.), *Competition, Innovation and the Microsoft Monopoly: Antitrust in the Digital Marketplace* (Boston, MA: Kluwer Academic, 1999).

[131]   This may happen either in the very market in which the pricing conduct occurs, or in other related markets. Geoff Edwards, *The perennial problem of predatory pricing: a comparison and appraisal of predatory pricing laws and recent predation cases in the United States and Australia* 30 ABLR 170 (2002), at 171.

[132]   Geoff Edwards, *The hole in the section 46 net*, ABLR 31, 151 (2003), at 152.

[133]   Michael J. Meurer, *Controlling Opportunistic and Anti-Competitive Intellectual Property Litigation*, 44 B.C. L. Rev 509 (2003), Peter H. Huang, *Still Preying on Strategic Reputation Models of Predation*, 3 Green Bag 2d 437 (2000).

irrational behavior, and hence so rare that a legal rule against predatory pricing would do more harm than good by providing false signals to the market.[134] In contrast, some commentators[135] argue that the economic models of the Chicago School rely on an erroneous framework of perfect information, while in a dynamic world of imperfect and asymmetric information, strategic conduct can be profitable.

In light of this disagreement among experts, courts and regulators have hesitated to interfere in the market for standard software on the grounds of predatory pricing claims on either side of the Atlantic. The District Court for the District of Columbia noted that Microsoft charged $ 89 for the Windows 98 upgrade as the "revenue-maximizing price," even though Microsoft had found in an internal study that a price of $ 49 would have been profitable. In addition, Microsoft charged different prices to different Original Equipment Manufacturers.[136] Further, Microsoft's internal emails proved that Microsoft steeply discounted its software expressly as a means of preventing the growth of Linux in developing markets. The D.C. Circuit, however, maintained that this pricing strategy, and also the free distribution of Internet Explorer both over the Internet as well as bundled with Microsoft's Windows operating system was "pro-competitive" behavior and therefore legal, even though it effectively dried out Netscape's browser market. It held: "The antitrust laws do not condemn even a monopolist from offering its product at an attractive price and we therefore have no warrant to condemn Microsoft for offering [Internet Explorer] free of charge."[137] Like the U.S. courts, the European Commission abstained from challenging the conduct under the predatory pricing claim.[138]

The effectiveness of predatory pricing rules is, in fact, questionable in markets in which the products primarily consist of intellectual property: If variable costs are almost non-existent (close to zero) due to low costs of reproducing the product, pricing below *average variable costs*, the precondition of a predatory pricing

---

[134]   Robert H. Bork, *The Antitrust Paradox: A Policy at War with Itself*, 2cd Ed. (New York, NY: 1993) [Bork, *The Antitrust Paradox*]; Frank H. Easterbrook, "Predatory Pricing Strategies and Counterstrategies" (1981) U. Chic. L. Rev. 50, 336; the same result achieved John S. McGee, "Predatory Price cutting: The Standard Oil Case" (1958) J. of L. & Econ. 1, 137.

[135]   See Patrick Bolton, Joseph F. Brodlay & Michael H. Riordan, "Predatory Pricing: Strategic Theory and Legal Policy" (2000) Georgetown L. R., and "Predatory Pricing: Response to Critique and Further Elaboration", [2001] Boston Univ. School of Law Working Paper No. 01-23, published in (2001) Georgetown L. J. 89, 2495.

[136]   United States v. Microsoft Corp., 65 F. Supp. 2d 1, 17 (D.D.C. 1999).

[137]   United States v. Microsoft Corp., 253 F.3d 34, 90 (D.C. Cir. 2001) .

[138]   EC, Microsoft-Decision, *supra* note 6, at [464.].

claim, can almost never be shown. This is particularly true with respect to the market for standard software.[139] Under these conditions, regulation on the basis of predatory pricing is ineffective.

b. Tying / Bundling

Tying is the sale or lease of one product made dependent on the sale or lease of another. [140] In the software market this can be achieved by both technical ("technological tying") and legal means ("contractual tying"), for example when licensing terms "connect" the basic and the additional software in a way that the licensee cannot acquire one part of the product without the other.[141] "Contractual tying" is unlawful under the *Sherman Act* § 1 if a company with market power in the market of the principal good ties it to another good, offers consumers no choice to purchase both products separately, and thereby forecloses a significant share of commerce with its arrangement.[142] It is similarly subject to antitrust scrutiny under Article 81 of the EC-Treaty. "Technological tying" is only subject to the provisions against abuse of market power (Article 82 of the EC Treaty and the *Sherman Act* § 2 respectively).

One might argue that a rational market participant would re-price the wanted product instead of bundling it with an unwanted product.[143] Further, tying might even create higher value products consisting of several lower value products by providing the benefits of technical integration and lower transaction costs.[144] Despite these potentially beneficial effects (that account for strong support for the legality of some tying

---

[139]   With respect to the economic characteristics of standard software, see *supra* II.

[140]   See e.g. Andrew Chin, Decoding Microsoft: A First Principles Approach; 40 Wake Forest L. Rev. 1 (2005); David A. Heiner, Assessing Tying Claims in the Context of Software Integration: A Suggested Framework for Applying the Rule of Reason Analysis, 72 U. Chi. L. Rev. 123 (2005); Warren S. Grimes, "Antitrust Tie/In Analysis After Kodak: Understanding the Role of Market Imperfections", 62 Antitrust L.J. 263 (1994); Keith N. Hylton & Michael Salinger, "Tying Law and Policy: A Decision-Theoretic Approach, 69 Antitrust L.J. 469 (2001), and the response by Warren S. Grimes, "The Antitrust Tying Law Schism: A Critique of Microsoft III and a response to Hylton and Salinger", 70 Antitrust L.J. 199 (2002-2003), with reply by Keith N. Hylton & Michael Salinger, "Reply to Grimes: Illusionary Distinctions and Schisms in Tying Law", 70 Antitrust L.J. 231 (2002-2003); Gordon B. Spivack, "The Chicago School Approach to Single Firm Exercises of Monopoly Power: A Response", 52 Antitrust L.J. 651 (1983), at 663 et seq., as well as the references cited infra and *id.*

[141]   See Zetzsche, *Software Industry*, *supra* note 114, at 115 et seq.

[142]   See United States v. Microsoft Corp., 253 F.3d 34, 85 (D.C. Cir. 2001); see also Eastman Kodak Co. v. Image Tech. Servs., Inc., 504 U.S. 451, 461-62 (1992)..

[143]   Bork, *The Antitrust Paradox*, *supra* note 134, at p. 372-374; Posner, *Antitrust Law*, *supra* note 113; George Stigler, "United States v. Loew's Inc.: A Note on Block Booking" (1963) Sup. Ct. Rev. 152.

[144]   Melissa Hamilton "Software Tying Arrangements Under the Antitrust Laws: A more Flexible Approach" (1994) 71 Denv. U.L.Rev. 607, with the argument that bundling allows the manufacturer further development, and upgrading of a software product, which the restrictive view would not allow.

arrangements in literature),[145] judicature considers tying generally to be illegal per se because tying arrangements pose an unacceptable risk of stifling competition.[146] For example, with respect to standard software, competitors in the market for additional software (such as applications) are unable to sell their products, if a comparable product already comes bundled with the prevailing product in an important market, such as operating systems.[147] Thus, a monopolist in one market might (try to) expand his market power to another unrelated market, a behavior termed "leverage".[148]

The problem of tying initially achieved visibility in the hardware market. In the 1970s, manufacturers of plug-compatible peripheral products (like disk-drives or memory) sued IBM for integrating functions of previously separate products into a single combined product.[149] The courts, however, dismissed the suits either by not distinguishing two separate products or they justified the tying as a beneficial combination due to the combined product's technical superiority.

While Microsoft's inclusion of Internet Explorer in Windows was reviewed by the D.C. Circuit in 1998 (when the DoJ claimed a violation of the 1994 consent decree without success), in 2001 (where the DoJ attacked the tying on the basis of § 1 and § 2 of the Sherman Act) and 2004 (when the Circuit Court reviewed the District Court decision which was almost identical to the settlement), only the 2001 proceeding discussed questions of relevance in this context. Microsoft had used three technological methods to effectively bind its browser to Windows: Microsoft excluded the browser from the "Add/Remove Programs" control panel; it designed Windows to override the user's choice of a default browser in certain cases; and it commingled code related to browsing and other code in the same files, so that any attempt to delete files containing the browser would necessarily cripple the operating system. The court noted that "antitrust scholars have long recognized the undesirability of having courts oversee product design" and that "any dampening of technological innovation would be at cross-purposes with antitrust law."[150] It emphasized that the technical advantages of a

---

[145]   *.

[146]   Jefferson Parish Hosp. Dist. No. 2 v. Hyde, 466 U.S. 2, 9 (1984).

[147]   E.g. applications like Windows Media Player coming bundled with operating systems like Microsoft Windows.

[148]   Mark Lemley at al., *Software and Internet Law*, 2nd ed. (New York: Aspen Law & Business, 2003), at p. 439 et seq.; Louis Kaplow, "Extension of Monopoly Power Through Leverage" (1985) 85 Colum. L. Rev. 515. See also Edward J. Iacobucci, "Tying as Quality Control: A Legal and Economic Analysis" (2003) 32 J. of Leg. St. 435.

[149]   See, e.g. In re IBM Peripheral EDP Devices Antitrust Litigation, 481 F Supp 965, 1003-04 (ND Cal 1979); Telex Corp v IBM Corp, 367 F Supp 258, 341-42 (ND Okla 1973).

[150]   United States v Microsoft Corp, 147 F3d 935, 948 (DC Cir 1998).

combined product (as achieved through the expansion of the Windows platform by integrating Internet Explorer into Windows 98) over the functionalities of separately bought items generally justify tying. It noted that the application of a per se analysis would stunt valuable innovation and that tying takes place in fully competitive markets as well.[151] Finally it emphasized that it is impossible to distinguish operating system code from non-operating system code (providing e.g. "middleware-functionality") and that removal of features would be to the detriment of both consumers and developers.[152] The tying claim was dropped on remand. The final settlement did not require Microsoft to un-bundle Internet Explorer from Windows, as it was considered sufficient that end users or computer retailers can replace Internet Explorer with a competing browser.[153] (Ironically, Microsoft's bundling of Internet Explorer with Windows might backfire as Microsoft may have violated a patent held by Eolas Techs, Inc. Thus, the bundling may eventually result in huge damages payments and royalties for the numerous free copies distributed by Microsoft.)[154]

In contrast to the U.S. courts, the European Commission found that Microsoft contravened Art. 82 of the EC Treaty by leveraging its near monopoly in the market for PC operating systems into the markets for both work group server operating systems, and media players.[155] It focused its analysis on consumer demand. It contended that the inclusion of Windows Media Player in the ubiquitous Windows operating system induced content providers and software developers to rely upon Windows media formats and application programming interfaces, thereby increasing their usage and potentially foreclosing competition in the market for media players in the future. Finally, the Commission emphasized that the demand for media players is evidently separable from the market for operating systems. In order to restore competition, the Commission initially considered a "must-carry regime" for competing Media-players[156] but decided against it as it found no viable method of determining which competing products to include. Instead, Microsoft was required to offer a version of Windows without Windows Media Player[157], to price it separately, and to refrain from using any commercial, technological or contractual terms that would have the effect of rendering the unbundled version

---

[151]   United States v Microsoft Corp, 253 F3d 34, 92-94 (DC Cir 2001).

[152]   Massachusetts v Microsoft Corp, 373 F3d 1199, 1208 (DC Cir 2004).

[153]   See United States v. Microsoft Corp., 231 F. Supp. 2d 144, 179-181 (D.D.C. 2002).

[154]   See Eolas Techs., Inc. v. Microsoft Corp., 399 F.3d 1325 (Fed. Cir., 2005).

[155]   EC, Microsoft-Decision, *supra* note 6, at *; EC, Microsoft- Press Release, *supra* note 6.

[156]   Commission Gives Microsoft Last Opportunity to Comment Before Concluding its Antitrust Probe, IP/03/1150, Aug. 6, 2003, at http://europa.eu.int/rapid/start/cgi/guesten.ksh?p_action.gettxt=gt&doc=IP/03/11500RAPID&lg=EN.

[157]   Named "Windows N", after the Commission declined Microsoft's suggestion of "Windows XP Reduced Media Edition".

of Windows less attractive than the bundled version. It superseded Microsoft's warnings that rewriting Windows to remove Media Player might result in a substandard Windows version[158] after a demonstration of the already existing Windows XP Embedded, a version of Windows without Media Player.[159] The underlying rationale of this remedy is that other media player vendors should be able to convince computer manufacturers to install their media player technology in the place of Microsoft's product. As a result, Microsoft would no longer be guaranteed ubiquity – unless their Media Player succeeds in the competition with other manufacturers. Therefore, the Commission assumed that competitors would approach the Original Equipment Manufacturers and induce them (e.g. by paying a fixed amount) to bundle their products instead of Microsoft's.

The European approach is more complicated than the traditional per se illegality of tying arrangements in the U.S.,[160] as the European Commission deems tying by a dominant firm an abuse only *if not justified by commercial usage or the nature of the product*.[161] Based on this premise, the European Commission condemned tying in various cases. Those tried were upheld on appeal[162] if the EU had jurisdiction over the case.[163] In the most recent Tetrapak *II* case, which concerned two cases of tying (the firm was partly super-dominant, and partly not dominant in the specific markets in question), the ECJ confirmed that tying was an abuse of a dominant position with respect to both cases. Thus, European practice makes decisions similar to the American per se rule, but on different legal grounds. Since economists have become more and more critical towards a strict rule on tying, European commentators are also hopeful that the European Commission " has clearly become less hostile [with regard to tying] over the last 25 years."[164]

---

[158]   Reuters, Microsoft Warns EU Rewrite Would Be Inferior, CNET News.com, Nov. 13, 2003, at http://news.com.com/2100-1016-5106777.html.

[159]   Reuters, Microsoft Rivals: Modified OS Isn't Mediocre, ZDNet News, Nov. 14, 2003, at http://zdnet.com.com/2100-1104-5107673.html.

[160]   See Jefferson Parish Hosp. Dist. No. 2 v. Hyde, 466 U.S. 2, 9 (1984).

[161]   Article 82 declares tying to be generally illegal. The European Commission's Guidelines on Vertical Restrains further define the exceptions, see COM(2000/C 291/01) – O. J. C 291 of 13.10.2000, online http://europa.eu.int/scadplus/leg/en/lvb/l26061.htm. For the European doctrine on tying, see Valentine Korah, *An Introductory Guide to European Competition Law and Practice* (Hart, 8th ed.: 2004) [Korah, *EC Competition Law and Practice*], at 140 et seq.

[162]   For example, in *Telemarketing*, Case 311/84, *CBEM v CLT and IPB* [1985] ECR 3261.

[163]   This argument helped Hilti eventually avoid being sentenced for illegal tying, see *Hilti*, Case 138/88, confirmed by the CFI (T-30/89) [1991] ECR II – 1439.

[164]   Korah, *EC Competition Law and Practice*, *supra* note 161, at 146.

c. Anti-competitive Licensing

Licensing allows the owner of an intellectual property right to realize its value without the necessary capital, geographic presence or market power. First, licensing provides a revenue stream without investment in the manufacturing of the good. Similarly, it allows worldwide exploitation of an invention, although the owner does not have a global distribution network. Furthermore, licensing facilitates the integration of licensed property with complementary factors of production.[165] Finally, licensing helps the owner to make his technologies a standard by increasing dissemination, even if his own product does not have a significant market share. Thereby, licensing facilitates the diffusion of inventions, reduces duplication of research and development, strengthens the incentives to invent, spurs incremental innovation and generates product market competition. Licensing as such is consequently pro-competitive.

When multiple competitors provide access to a specific technology, negotiations will result in fair and reasonable terms. However, a market participant with a dominant position can negotiate a contract in which the anti-competitive effects will outweigh the efficiency enhancing and pro-competitive effects of the license. Such anti-competitive effects can include denial of access to specific technologies, or limit the licensor's use of the licensed rights, e.g. by limiting the licensee's freedom to sell, resell, purchase, or otherwise. Thus, licensing might restrain trade on a vertical axis. Furthermore, the license terms themselves may harm competition, for example by forcing the licensee to bundle additional unpopular products with a popular one, [166] to prominently display unwanted products, or by contractually prohibiting separating out the unwanted product.[167]

While U.S courts accept the general notion that unilateral refusal to license copyrights can constitute illegal exclusionary conduct,[168] the First Circuit nevertheless presumed that an author's desire to exclude others from using copyrighted works provides valid justification to any harm done to consumers unless the

---

[165]   Federal Trade Commission, Antitrust Guidelines for the Licensing of Intellectual Property, April 6, 1995, reprinted in 4 Trade Reg. Rep. (CCH) ¶13,132, at 2.3.

[166]   Under this condition, licensing raises antitrust issues that are analogous to contractual tying, see *supra* III. 2. b) (2).

[167]   For an overview of the various practices, which Microsoft used to "promote" its unpopular product "Internet Explorer" during the time of the "Browser War" see Timothy F. Bresnahan, "The Economics of the Microsoft Case", John M. Olin Program in Law and Economics Working Paper 232 (March 2002), http://ssrn.com/abstract=304701, at p. 16.

[168]   *Eastman Kodak Company v. Image Technical Services, Inc.*,125 F.3d 1195; *1997 U.S. App. LEXIS 22608; 44 U.S.P.Q.2D (BNA) 1065; 1997-2 Trade Cas. (CCH) P71, 908* (9th Cir. 1997). The antitrust concerns are explicitly cited in the Federal Circuit's decision in *Intergraph v. Intel*, 195 F.3d 1346, at 1358; *1999 U.S. APP. LEXIS 29199; 52 U.S.P.Q.2D (BNA) 1641; 1999-2 TRADE CAS. (CCH) P72,697* (Fed. Cir. 1999): refusals to deal "may raise antitrust concerns when the refusal is directed against competition and the purpose is to create, maintain, enlarge a monopoly."

plaintiff proves special circumstances.[169] Similarly, a patentee may generally exclude competitors from the use of its patents despite the monopolistic nature of his or her activity.[170] However, power gained through legal advantages such as copyright can give rise to liability if a seller with a dominant position in one market seeks to leverage its position into an after market.[171] Furthermore, the "essential-facilities doctrine"[172] may provide a legal basis for requiring licensing on non-discriminatory terms. According to the Supreme Court, however, the doctrine only prevents a dominant firm from refusing *to continue* a contractual relation necessary for a rival to compete, but it does not prohibit a refusal by a dominant firm *to create* such a relationship in the first place. [173] In the vein of the general reluctance to require compulsory licensing expressed by American courts, the settlement in the Microsoft case does not provide for a wide range of compulsory licenses, but only requires Microsoft to provide access to communication protocols under "reasonable and nondiscriminatory terms to be determined by Microsoft.[174]

In contrast, European law appears more responsive to concerns of competitors requiring access to new technologies. In *Magill*,[175] the European Court of Justice emphasized that the exercise of an exclusive right by the proprietor may constitute abusive conduct if the following three criteria are fulfilled: (1) the appellant's refusal of the license prevents the appearance of a new product; (2) there is no sound justification for such refusal to deal; and (3) by enforcing their legal rights, the appellants reserved to themselves the secondary market by excluding all competition on that market.[176] The *Ladbroke*[177] and *Bronner*[178] decisions

---

[169]   Data Gen. Corp. v. Grumman Sys. Support Corp., 36 F.3d 1147, at 1151, 1184 et seq. (1st Cir., 1994).

[170]   See for example R*e Independent Service Organizations Antitrust Litigation*, 203 F.3d 1322, at 1326; *2000 U.S. App. LEXIS 2303; 53 U.S.P.Q.2D (BNA) 1852; Copy. L. Rep. (CCH) P28, 026* (Fed Cir. 2000).

[171]   See e.g. Eastman Kodak Co. v. Image Technical Services Inc., 504 U.S. 451, at 479 n. 49 (1992)

[172]   Developed in *United States v. Terminal Ry.*, 212 U.S. 1 (1912); *Aspen Skiing Co. v. Aspen Highlands Skiing Corp.*, 472 U.S. 585; 472 U.S. 585; 105 S. Ct. 2847; 86 L. Ed. 2d 467; 1985 U.S. LEXIS 115; 53 U.S.L.W. 4818; 1985-2 Trade Cas. (CCH) P66, 653 (1985); however, Areeda, "Essential Facilities: an Epithet in Need of Limiting Principles" (1989) Antitrust L. J. 58, 841 shows that the scope of the essential facilities doctrine is limited; for software programs in particular, see David McGowan, "Regulating Competition in the Information Age: Computer Software as an Essential Facility under the *Sherman Act*" (1996) 18 Hastings Comm. & Ent.L.J. 771, at 836-841.

[173]   *Verizon Communication, Inc. V. Law Offices of Curtis Trinko*, 124 S. Ct. 872 (2004).

[174]   Final Judgment, United States v. Microsoft Corp., CA No. 98-1232 (CKK), filed Nov. 12, 2002 (D.D.C. 2002)) at III.I..

[175]   *Magill, supra* note 99, at [46].

[176]   The judgment was unclear, however, as to the question of whether these criteria applied alternatively or cumulatively, see Estelle Derclaye, "Abuses of dominant position and intellectual property rights: a suggestion to reconcile the Community courts case law", (2003) W. Comp. 26(4), at 685, and "The IMS Health Decision: A Triple Victory" (2004) W. Comp. 27(3), 397, 399.

[177]   Case T-504/93, *Tierce Lasbroke SA v Commission* [1997] ECR II-923; [1997] 5 CMLR 309 (CFI).

further refined these requirements. In *IMS-Health*,[179] the European Court of Justice expressly held that a dominant competitor's refusal to grant a license may constitute an abuse of market power if the **license is indispensable for operating on a secondary market**. The license applicants must have the intent to provide new goods or services that the owner of the right does not offer and for which there is a **potential consumer** demand. **Objective considerations** may justify the refusal to license.

On the basis of this case law, the European Commission found that Microsoft had abused its market power and *limited technological development to the prejudice of consumers*[180] – a criterion found in Art. 82 (b) of the EC Treaty by denying third parties access to (subsequently changed) information on the network interfaces used in Windows. Changes in interface definitions are not reason for concern per se, as they are often inevitable in order to enable innovation and allow further development of software platforms. *Secret* change of interfaces, however, will render other manufacturers' products incompatible, as these manufacturers cannot adjust their products to these changes. Even a time-lag between implementation of changes in end-user-software and publication can severely damage a competitor's reputation.

In the case of Microsoft, these undisclosed interfaces were necessary to exchange data between computers running other operating systems (like Linux, Unix or MacOS) and computers running Microsoft Windows. The resulting interoperability problems between Windows systems employed by most users and (server) operating systems not manufactured by Microsoft (especially Unix) caused consumers to choose Microsoft server products to complement its ubiquitous Windows desktop operating system. While Microsoft pointed out that older versions of its interfaces were licensed and that it granted licenses under the settlement with the DoJ, the Commission noted that these licenses either did not cover the current versions of the software, were incomplete or were limited to specific connections (server to client, thereby leaving out client

---

[178]   Case C-7/97, *Oscar Bronner GmbH & Co. KG v. Mediaprint … GmbH & Co. KG* [1998] ECR 7791 (ECJ).

[179]   *Supra* note 99. See also the papers by Derclaye, *supra* note 176.

[180]   EC, Microsoft-Decision, *supra* note 6, at [693. et seq.], and [694]: "Due to the lack of interoperability that competing [work group server systems] products can achieve with the Windows domain architecture, an increasing number of consumers are locked into a homogeneous Windows solution at the level of [work group server systems]. This impairs the ability of such customers to benefit from innovative [work group server systems] features brought to the market by Microsoft's competitors. In addition, this limits the prospect for such competitors to successfully market their innovation and thereby discourages them from developing new products. … If Microsoft's competitors had access to the interoperability information that Microsoft refuses to supply, they could use the disclosures to make the advanced features of their own products available in the framework of the web of interoperability relationships that underpin the Windows domain architecture.

to client and server to server connections). On these grounds the Commission required Microsoft to disclose and license the necessary interface information on a timely basis.

The Commission explicitly acknowledged that "ordering Microsoft to disclose such specifications and allow such use of them by third parties restricts the exercise of Microsoft's intellectual property rights"[181] and therefore applied the European Court of Justice's test outlined above. The Commission found that Microsoft "*disrupted the previous level of supply*" by switching from (relatively) open disclosure to incomplete disclosure of interfaces in Windows 2000.[182] It also maintained that Microsoft's behavior "*risked eliminating competition*" – a criterion from *Magill*, that is based on earlier EC case law in *Commercial Solvents* and *Telemarketing*.[183] Finally, it established that the disclosure of interface information was "*indispensable to carry on business.*"[184]

d. Threats, Bribes and Fraud

In addition to the contentious practices discussed above, Microsoft employed two evidently illegal and unjustifiable means to protect and expand its dominant position: Threats and bribes to third parties who were dependent on Microsoft, and fraud on software developers. Those measures made it an easy target for the antitrust authorities' claims of illegal monopolization

As an example of a threat, Microsoft suggested that it might optimize Windows for AMD-CPUs if Intel did not stop supporting Sun in developing a virtual machine for its cross-platform JAVA technology.[185] Similarly it warned Apple Computer that it might stop supporting and developing Microsoft Office for Macintosh Operating System (MacOS), if Apple continued bundling Netscape as the default browser for MacOS.[186] As bribes, internet access providers who offered Internet Explorer were included in Microsoft's Internet Connection Wizard, or even allowed to distribute their connection software with Microsoft Windows.[187] The settlement between the DoJ and Microsoft in 2002 seeks to prevent similar practices in the

---

181   EC, Microsoft-Decision, *supra* note 6, at [546.].

182   *Ibid*, at [578.].

183   Cases 6/73 and 7/73, *ICI Commercial Solvents v Commission* [1974] ECR 223 [1974] 1 CMLR 309 (ECJ), and Case 311/84, *CBEM v CLT and IPB* [1985] ECR 3261.

184   EC, Microsoft-Decision, *supra* note 6, at [588.], citing *Bronner*, *supra* note 178. The Commission held that both criteria were fulfilled due to (1) the extraordinary market strength of Microsoft in COS, Ibid, at [590. et seq.] and (2) the significant importance of interoperability between client operating systems and work group server systems, Ibid, at [637. et seq.].

185   United States v. Microsoft Corp., 253 F.3d 34, 77 (D.C. Cir. 2001).

186   United States v. Microsoft Corp., 253 F.3d 34, 72-74 (D.C. Cir. 2001).

187   United States v. Microsoft Corp., 65 F. Supp. 2d 1, 63-64, 68-66 (D.D.C. 1999).

future by explicitly prohibiting retaliation against third parties for distributing, supporting, promoting or developing competitors' software. It also requires uniform licenses to prevent compulsion for using Microsoft products.[188] Original Equipment Manufacturers are now free to install and display icons, shortcuts and menu entries for competing software in any size or shape. Further, these entities, along with end users, can designate the competing software as a default.[189] On a similar notion, the European Commission ordered Microsoft not to give PC manufacturers a discount that is conditional on buying Windows together with Windows Media Player.[190]

Microsoft developed an especially devious ploy to slow widespread adoption of Sun Microsystems' JAVA.[191] Initially, Microsoft acquired a license from Sun to provide proprietary versions of both the Java "programming language" (Microsoft J++) and the "virtual machine". However, Microsoft extended Sun's standard by allowing direct access to many elements of its Windows operating system – thereby rendering applications developed using Microsoft J++ incompatible with third parties' "virtual machines". Software developers received preferred access to and support for Microsoft J++. Microsoft "deceptively failed to tell the developers that if they used these tools their software would run only on"[192] Microsoft's virtual machine. Thus, "developers who were opting for portability … unwittingly [wrote] Java applications that [ran] only on Windows".[193] The Court of Appeals for the Fourth Circuit denied Sun an injunction requiring Microsoft to carry their most current, standard-conformant version, but prevented Microsoft from further distributing their incompatible version on the basis of copyright law (not antitrust law!).[194]

While the previous section has shown that - especially in the U.S. - antitrust law generally has difficulties in accounting for limits on Microsoft's quest for monopoly, this obviously anti-competitive conduct made Microsoft an easy target for enforcement agencies' claims of illegal monopolization.

---

[188]   Final Judgment, United States v. Microsoft Corp., CA No. 98-1232 (CKK), filed Nov. 12, 2002 (D.D.C. 2002)) at III.A / III.B.

[189]   Final Judgment, United States v. Microsoft Corp., CA No. 98-1232 (CKK), filed Nov. 12, 2002 (D.D.C. 2002)) at III.C.1-2.

[190]   EC, Microsoft-Decision, *supra* note 6, at *, and Press Release, *supra* note 6, at 2.

[191]   JAVA is the middleware of which many believed that it could overcome Microsoft's monopoly in the operating system. See *supra* at II on the reasons for Microsoft's fear of "Middleware" such as web-browsers and JAVA.

[192]   Lemley et al., *Software and Internet Law, supra* note 148, at 416.

[193]   *United States v. Microsoft Corp.* (II), *supra* note 137, 253 F.3d 34, at 56 et seq.

[194]   Sun Microsystems, Inc. v. Microsoft Corp. (In re Microsoft Corp. Antitrust Litig.), 333 F.3d 517 (4th Cir., 2003).

2. Mergers

A merger of two competitors is subject government scrutiny because it effectively reduces the number of market participants. Section 7 of the US *Clayton Act* prohibits mergers that may "substantially lessen competition" (the so-called "SLC-test"). The EC Merger Control system is grounded in the recently revised Council Regulation (EC) No 139/2004 of 20 January 2004. Article 2 (2) states that "a concentration which would not *significantly impede effective competition* in the common market or in a substantial part of it, in particular as a result of the creation or strengthening of a dominant position, shall be declared compatible with the common market." Both concepts are essentially equivalent to each other.

In the early 1990s, Intuit Inc.'s Quicken was by far the most successful software for personal financial management (featuring an amazing 70% market share as compared to 20 % using "Microsoft Money" [195]). In April 1994, after failing to reach an (illegal) market division agreement,[196] Microsoft decided to purchase Intuit for about US-$ 2 billion in shares. Microsoft was even willing to give away significant parts of its own product ("Microsoft Money") to its then competitor Novell essentially for free.[197] However, the DoJ objected to Microsoft's merger proposal. In May 1995, the DoJ filed suit in federal court to stop the merger.[198] The DoJ alleged that new entry into the personal finance software market is particularly difficult, because the range of possible features is limited and users therefore have little incentive to switch products from an incumbent to a new product. Further, a successful product required cooperation with financial service providers which would be unwilling to enter into agreements with insignificant new competitors. Thus, even though it was possible to develop a feature-competitive product, its acceptance in the market significantly depended on the negotiation power of the manufacturer vis-à-vis banking institutions. It was held to be unlikely that any other firm (not even Novell) would have a position sufficiently strong to compete with Microsoft or Quicken (a firm with a significant first mover advantage). Thus, the DoJ expected prices to increase following the acquisition. The transfer of "Money" to Novell was held to be insufficient to prevent the anti-competitive effects of the Intuit acquisition. In the absence of the acquisition, the DoJ expected Microsoft to invest more in improving MS

---

[195] These numbers are based on the narrow market division by the government - Microsoft argued for a broader market that covered all ways of managing money including pens and calculators.

[196] See <http://www.businessweek.com/microsoft/updates/up90216a.htm>

[197] At that time Novell offered both a client operating system (Novell DOS, formerly DR DOS, later OpenDOS), an Office suite (PerfectOffice, formerly WordPerfect Office, later Corel Office), a Server operating system (Novell Netware) and it was rumoured that they might eventually enter the market for graphical user interfaces.

[198] U.S. v. Microsoft Corporation and Intuit, Inc. (N.D. Cal.) No. C 95-1393 WHO at <http://www.courttv.com/archive/legaldocs/cyberlaw/microsoft/intuit2.html>

Money than Novell could do. Consequently, there would be a stronger (beneficial) rivalry between "Microsoft Money" and "Intuit Quicken", than competition between "Microsoft Quicken" and "Novell Money". From a U.S. perspective, remarkably, the DoJ emphasized the need to *protect competition* in the emerging home banking market in the future. Controlling the market for personal finance software market in addition to its dominance in the operating systems market would have enabled Microsoft to seize control of the markets of the future. By keeping Quicken independent, choice was preserved for financial institutions and consumers. Judge Orrick of the U.S. District court for Northern California set an expedited trial date for June 26, 1995.[199] On May 20, 1995, however, Microsoft backed out of the deal with Intuit by paying a termination fee of $146.5 million. Having settled antitrust allegations with regard to Windows 95 with the DoJ and the European Commission (the consent-decrees)[200] shortly in advance, Microsoft was not willing to spend years in the (public) courtroom again.

A similar issue might arise from the recently announced acquisition of Macromedia, Inc. by Adobe, Inc.[201] Both companies together cover an enormous share in certain products of the application market with competing products: Adobe's GoLive competes with Macromedia's Dreamweaver; Macromedia's Freehand goes up against Adobe's Illustrator; and Fireworks compares to Photoshop.[202] It seems highly unlikely that a merged Adobe-Macromedia will continue to develop all these products. By reducing choice, a lessening of competition seems inevitable. Adobe, however, points at Open Source developers and Corel (which outsells both firms in some markets) to warrant future competition in the market.[203] In addition, Adobe emphasizes that Microsoft can enter into its market at any time and thereby provides "potential competition". To exemplify this, Adobe refers to certain document exchange facilities (codenamed "Metro") which will be implemented in the next version of the Windows operating system (codenamed "Longhorn") and thereby threaten Adobe's popular Acrobat-product (which creates and displays the ever-popular PDF-files). On first sight these arguments seem to be based on a reasonable application of traditional antitrust-principles. One

---

[199]   U.S. v. Microsoft Corporation and Intuit, Inc. (N.D. Cal.) No. C 95-1393 WHO at
<http://www.courttv.com/archive/legaldocs/cyberlaw/microsoft/intuit1.html>.

[200]   *Supra* note 119.

[201]   Laurie J. Flynn, Adobe Buys Macromedia for $3.4 Billion, The New York Times, Section C; Column 4; Business/Financial Desk; Technology; Pg. 7 (April 19, 2005);

[202]   David Frith, Staying ahead of Microsoft, The Australian, FEATURES; Double ClickIT Alive / Computers; Pg. T17 (April 26, 2005).

[203]   Scarlet Pruitt and Martyn Williams, Adobe to buy Macromedia for US$3.4 billion,
<http://www.pcworld.idg.com.au/index.php/id;1197843252;fp;2;fpid;1>

might indeed argue that potential competition by Microsoft would in fact justify almost any lessening of competition imaginable. However, Microsoft's own antitrust worries with regard to implementing new features in Windows significantly remedy the danger of its market-entry. Instead the peculiarities of the software-market require an analysis of the merger effects beyond pure redistribution of market shares. The antitrust authorities' response to the merger proposal will be interesting to examine.

On the other hand, both the Court for the Northern District of California[204] and the European Commission[205] allowed PeopleSoft and Oracle to merge, in well-reasoned decisions. This occurred despite the fact that EC authorities applied a narrow market definition of "high function" Human Resources Management and Financial Management Services software. This type of software is exclusively purchased by large and complex enterprises requiring high standards of performance and support. Even though PeopleSoft, Oracle and SAP held big market shares in these markets, their respective bidding behavior was not affected by the presence of any of the other three firms as rivals in the final rounds of a given bidding contest. Consequently, other firms could sufficiently influence pricing by incumbents in order to warrant competition in the future. Further, "the heterogeneity of the products, the asymmetries in the market shares of the different players and the lack of price transparency also rendered coordinated effects implausible in the industry."[206] Thus, the antitrust authorities correctly recognized the characteristics of the market for customized software. Due to low economies of scale and network externalities in this market, small competitors may effectively compete with large participants. Regulatory interference was not justified.

3. Pro-Concentration Effect of Efficiency

Government agencies and courts on both sides of the Atlantic have tried their best to balance the need for competition in an increasingly monopolistic industry with the potential harm to innovation. While antitrust law provides some frameworks to solve the conflict between free competition and intellectual property,[207] the

---

[204]   *United States of America et al v Oracle Corporation*, No C 04-0807 VRW (Northern District of California), available online from the website of the U.S. Federal Department of Justice, www.dpj.us.gov , at 124 et seq.

[205]   European Commission, Press Release "Commission clears Oracle's takeover bid for PeopleSoft", IP/04/1312, 26 October 2004.

[206]   Ibid.

[207]   Federal Trade Commission, Antitrust Guidelines for the Licensing of Intellectual Property, April 6, 1995, reprinted in 4 Trade Reg. Rep. (CCH) ¶13,132; Europe: Commission Regulation (EC) No. 772/2004 of 27 April 2004 on the application of Article 81 to technology transfer agreements, OJ L 123/11, 27.04.2004 - The Technology Transfer Block Exemption [TTBER], and the Commission Regulation (EC) No 2659/2000 of 29 November 2000 on the application of 81 (3) of the Treaty to categories of specialization agreements, OJ L 304/7 of 5.12.2000 [R&D Regulation]. For details, see Mauritis

efficiency criterion remains the nucleus of any antitrust case regarding standard software: Under the provisions against anti-competitive agreements (§ 1 of the U.S. *Sherman Act*[1] and Article 81 (1) of the EC Treaty) courts are required to determine "unreasonableness" by assessing whether the restraint in question contributes to an efficiency enhancing integration of economic activity,[208] or whether it should be exempted from the general ban under Article 81(3) of the Treaty due to positive welfare effects.[209] Under the monopolization provisions (§ 2 of the *Sherman Act* and Article 82 of the EC Treaty), courts and agencies have to distinguish exclusionary acts, which reduce social welfare, from competitive acts, which increase it. Even though the distinctive criteria in question are far from clear,[210] both the U.S. and the EC law emphasize the efficiency of the specific conduct in question, i.e. whether or not the specific conduct has a welfare enhancing effect.[211] Under the merger control schemes (§ 7 of the US *Clayton Act* and Council and Regulation (EC) No 139/2004 of 20 January 2004), if a merger substantially lessens competition or *significantly impedes effective competition*, respectively, the defendant may rely on an "efficiency defense" and justify the negative effects of his or her conduct on the competition in the market.[212] How does this efficiency defense influence the market for standard software?

In the market for standard software, increasing market power necessarily causes short term "efficiency gains" for the individual manufacturer (due to the natural monopoly, economies of scale) *and* the consumers (due to network externalities and lower prices, possibly resulting from economies of scale). Microsoft's bundling of Internet Explorer with Windows and providing it as a free download essentially forced Netscape to provide its browser for free as well. The huge investments in research and development significantly improved both products. Insofar these actions had a beneficial effect to consumers.[213] In standard software

---

footnotes

Dolmans & Anu Piilola, "The New Technology Transfer Block Exemption: A Welcome Reform, After All", (2004) 27 World Competition 3, 351; Zetzsche, *Software Industry*, *supra* note 114, at 69 et seq.

[208] *Broadcast Music, Inc. v. Columbia Broadcasting System, Inc.*, 441 U.S. 1, at 16-24 (1979).

[209] European law regulates in detail which agreements are considered to enhance efficiency, see Korah, *EC Competition Law and Practice*, *supra* note 161, at 87 et seq. See also Zetzsche, *Software Industry*, *supra* note 114, at 14, 80, 106 et seq.

[210] Einer Elhauge, "Defining Better Monopolization Standards", (2003) Stanf. L.Rev. 253. Regulators rely on the "sacrifice test", the "as-efficient competitor test", or the "consumer-harm test", all of which have their pitfalls, see e.g. John Vickers, "Abuse of Market Power" (speech, 3 Sept 2004), online http://www.oft.gov.uk/News/Speeches+and+articles/2004/spe03-04.htm, at 13.

[211] See Alan J. Meese, *Price Theory, Competition, and the Rule of Reason*, * U. Ill. L. Rev. 77 (2003).

[212] EU: *; U.S: *FTC v Tenet Health Care Corp*, 186 F3d 1045, 1054-1055 (8th Cir 1999); *FTC v Staples, Inc*, 970 F Supp 1066, 1088 (D DC 1997).

[213] *United States v. Microsoft Corp.*, 65 F. Supp. 2d 1, 46 (D.D.C. 1999).

markets, which are deemed a natural monopoly, a high level of market concentration will therefore always be "more efficient" than small competitors struggling with each other. Insofar, the software industry resembles (the early days of) telephony[214]: A traditional understanding of "efficiency" would actually require regulators and adjudicators to support the market's inherent monopolization tendencies. Similarly, it is unlikely that either consumers or competitors can distinguish damages caused by anti-competitive conduct from profits lost due to legitimate competition. The loss in consumer welfare by lack of innovation or choice is almost impossible to quantify or remedy.[215]

This result, however, is in line with the Chicago School's "efficiency"-oriented approach to antitrust law. From that perspective, competition, rather than government intervention, provides the best consumer protection and warrants the greatest economic efficiency.[216] Under this reasoning antitrust authorities have softened the "iron hand of antitrust law" in favor of efficiency, thereby allowing monopolists to flourish. Going even further, an "economic rationality"-standard was occasionally applied to mergers based on the belief that business expansion through mergers created beneficial growth in the overall economy.[217] Under this standard it would be "irrational" for a small company (like Netscape) to "wastefully compete" in the market instead of surrendering to a lucrative buy-out by a major participant (like Microsoft) for a huge premium. The final settlement between the U.S. Department of Justice and Microsoft reached in 2001 was unsatisfactory insofar as the remedies imposed were evidently insufficient to counter Microsoft's market power.

The antitrust courts and authorities of the European Union, however, do not adhere to efficiency to the same extent as their U.S. counterparts. While emphasizing that creativity is mainly stimulated by intellectual property law, the European Commission emphasized the public good aspect of the resulting legal rights. It maintains that competition warrants a higher possibility of innovation and consumers will therefore benefit from a fierce process of rivalry.[218] The above antitrust analysis nevertheless demonstrated that even the EC authorities' mainly focused on Microsoft's aggressive and evidently illegal actions against competitors.

---

[214]   Kelber, *supra* note 18, at 159.

[215]   John E. Lopatka & William H. Page, Who Suffered Antitrust Injury in the Microsoft Case?, 69 Geo. Wash. L. Rev. 829 (2001).

[216]   David P. Baron, Business and its Environment (3d ed. 2000) at p. 278.

[217]   Charles R. Geisst, Monopolies in America (Oxford University Press 2000) at p. 286.

[218]   EC, Microsoft-Decision, *supra* note 6, at pp. 694, 700, 711.

In light of this observation, it was Bill Gates' and his managers' (maybe only) strategic failure to subject his company to state scrutiny through aggressive statements issued both internally and externally,[219] as it was doomed to supersede other manufacturers under the inevitable rules of the economics that characterize the standard software industry anyway.

---

[219]   See the interesting evidence gathered in the various Microsoft cases, for example EC, Microsoft-Decision, *supra* note 6, at *.

IV. Regulating the Future

As shown above, neither intellectual property laws nor antitrust laws are currently effective in countering the market's inherent drive towards concentration. For the future, three options for handling standard software markets exist:

- We may treat it the way we currently do and abstain from any tighter regulation (discussed under A below);

- We may eventually move toward full regulation by a specialized government agency that sets prices and supervises product development and quality (discussed under B below);

- We may enforce the existing antitrust laws with the goal of preserving existing competition as a "process of rivalry" (discussed under C below).

A. Abstention

An optimist might have hope for the dynamic nature of the software market. Monopolies might not persist, but end as soon as a competitor has reaped the (rightful) rewards for his technological innovation.[220] Such "temporary" or "serial" monopolies are considered beneficial to the market.[221] The "leapfrogging" innovation in the industry in the 1980s and early 1990s seems to support that hope. It is, however, noteworthy that the market for operating systems software for personal computers as "classic" standard software was always dominated by one manufacturer, initially IBM and later Microsoft. "Free" competition never existed. In addition, it is questionable whether the relatively short experience with standard software offers meaningful insights into the *modus operandi* of the market.

The concentration in the standard software market might also be merely incidental to the increasing saturation of the market. If software becomes increasingly financially unattractive, market power would cease, as the costs of avoiding regulatory scrutiny would exceed the possible profits gained by abuse of monopoly

---

[220] E.g. John E. Lopatka, "*Unites States v. IBM*: A Monument to Arrogance", 68 Antitrust L.J. 145 (2000-2001), at 149.

[221] Fisher, McGowan & Greenwood, *IBM and the U.S. Data Processing Industry: An Economic History* (1983), at 163; Stanley & Margolis, *supra* note 130, at 10, 137.

power. Going even further, challenging market concentration *per se* seems to unreasonably punish the most efficient competitor for his or her (desirable) efficiency.[222]

This pro-concentration bias is short-sighted, however. The efficiency argument is only sound as long as the barriers to market entry are low and significant competitive pressure holds prices down. This is, in fact, not the case: The natural monopoly structure on the supply side and the network externalities on the demand side of the software industry all but eliminate this kind of pressure on dominant players in the software market. Instead, the dominant manufacturers in the software industry tend to expand the scope of their monopolies to after-markets.

Thus, plain application of the efficiency criterion may force small software manufacturers out of the market, not because of bad products, but because of high prices due to few economies of scale. While failure is inherent in doing business (and shouldn't be a reason for mourning per se), short-term and long-term efficiency often conflict in the standard software market. On the short run, efficiency gains on the product site will undeniably be beneficial to both manufacturers (economics of scale) and consumers (network externalities). A policy that is solely predicated on such short-run efficiency will, however, threaten innovation in the long-run: Since development of complex software is very costly, incumbents will stop improving such products or heavily increase prices unless there are sufficient viable competitors. Both types of conduct are detrimental to society.

Finally, relying exclusively on "efficiency" disregards that competition law is ultimately meant to benefit society. Externalities provided by innovative businesses justify a lenient approach towards concentration as *societal* benefits exceed investors' *private* benefits.[223] As a competitor's market share grows, it usually becomes less innovative.[224] However, society usually benefits from lower product prices due to more efficient production and economies of scale. This holds true as long as competition forces the firm to pass (an

---

[222]   Bork, *The Antitrust Paradox*, *supra* note 134, at *.

[223]   See Jeffrey G. MacIntosh, "Legal and Institutional Barriers to Financing Innovative Enterprise in Canada" (monograph prepared for the Government and Competitiveness Project, School of Policy Studies, Queen's University, Discussion Paper 94-10, 1994) [unpublished] at 8; see also Paul A. Gompers & Josh Lerner, *The Money of Invention: How Venture Capital Creates New Wealth* (Cambridge, MA: Harvard Business School Press, 2001); OECD, *The New Economy: Beyond the Hype*, Final Report on the OECD Growth Project: Executive Summary (Paris: OECD, 2001) at 12 *et seq.*, online <http://www.oecd.org/dataoecd/2/26/2380634.pdf>.

[224]   Ibid.

appropriate part of)[225] its potentially higher profit margin along to consumers. Without competition, however, no incentive to pass along efficiency gains to consumers remains. In the absence of regulatory interference, consumers will be charged monopolistic prices, and societal interests will be harmed.

These problems would be less dire if they were limited to a specific market. As antitrust law is currently unable to effectively counter pro-concentration practices such as certain pricing strategies, tying, refusals to grant licenses, and mergers, concentration does, however, take place both *horizontally*, as well as *vertically*: The hostile takeover of PeopleSoft by Oracle exemplifies the capture of an application provider by a server software manufacturer, designed to prepare Oracle for the "war" against Microsoft in years to come. Similarly, Microsoft and SAP held merger negotiations in late 2003.[226] As large manufacturers expand their market power to additional markets, their power *over the industry* grows. It is therefore only a matter of time until all levels of software (including operating systems, middleware, and applications) merge in the hands of a few incumbents. A multiple-market oligopoly increases the risks for society – and hence the need for regulation as well. Superseding even temporary market power of such software powerhouses will become increasingly more difficult.

Consequently, without intervention by regulators, consumers will eventually be stripped of all instruments that have warranted innovation and / or low prices up to this point in time. Further regulatory action is therefore necessary.

B. Direct (Ex Ante) Regulation

But, *how* should regulators substitute for the lack of competition? Other industries that exhibit characteristics of natural monopolies and network externalities are subject to direct regulation of output, product development and price.[227] Given that computers stand in almost every private and public household, and that possibly more people have software driven tools than shares in their deposit, is it time to establish the Federal or European Software Agency?

---

[225] Some incentives for innovation must remain!

[226] Microsoft provides server software and operating systems, and also some business software; SAP offers the most successful business software.

[227] See *

Previous experiences with direct ex ante regulation, however, provide little confidence in the efficiency of such an agency.[228] Ex ante regulation requires unpleasant tasks such as determining "appropriate returns" for a monopolistic manufacturer or setting proper requirements with respect to product innovation. Traditional bureaucracy is unfit to deal with the fast-moving subject of software development. The bureaucratic time-lag experienced in other regulated industries might prove fatal in the software industry. Furthermore, there is little use in employing hundreds of bureaucrats, defending a storm of lobbyists, and letting the industry finally pay the costs for direct supervision. We therefore deem direct regulation an undesirable option.

C. Indirect Regulation

The long-term prospect of the standard software market seems bleak indeed: In the (expected) absence of legislative changes the current (short-sighted) practice of antitrust law will lead to either costly and inefficient ex ante regulation or uncontrolled monopoly pricing in the long run.

A flexible approach seems to be the only way to avoid the time-lag characterizing direct regulation of output and prices in other industries. While all of the developments suggested here come at high costs to society, we maintain that implicit regulation through sector-specific adjustments to antitrust and intellectual property law represents the least expensive option. In particular, by enabling **both** public and private enforcement, antitrust law avoids the problems of both entirely private enforcement systems[229] and primarily state-controlled enforcement systems. [230] Additionally, the existing antitrust law is a **highly flexible instrument**, with respect to both scope and available sanctions. Since a relevant market share is required to trigger the antitrust law provisions and antitrust laws contain certain minimum thresholds that prevent antitrust scrutiny with respect to small firms, the costs will be burdened only on major firms, whereas direct regulation would impose a burden on small and medium enterprises as well.

Given that an indirect regulation regime is already in place, what changes are necessary to warrant competition in the market on the long run? Since scalability in the software market is virtually unlimited,

---

[228]   The settlement between Microsoft and the U.S. Department of Justice was limited to a five year period instead of ten years, as the government was scared of it becoming "highly regulatory", see United States v. Microsoft Corp., 231 F. Supp. 2d 144, 194 (D.D.C. 2002).

[229]   As in Intellectual Property law, see *supra* III.A.5.

[230]   As in direct regulation by a government agency, see above IV.B. with further references.

multiple manufacturers for a specific product class always seem to be **facially "inefficient"**. However, as was shown above, emphasizing this "inefficiency" erroneously focuses only on short-term benefits, as antitrust law is designed to prevent the dangers of dependency on a single manufacturer. The best means to that end lie in the protection of future rivalry even at the price of inefficiencies in the present. Thus, we have to (re-)consider shifting the general efficiency-enhancing rationale of antitrust towards preserving existing competition as a "process of rivalry", at least to a certain extent.

Antitrust law is not only designed to protect pricing, but also to ensure that consumers have a choice among different products, which implies the existence of several manufacturers and innovation. Under this rationale, smaller competitors are generally worthy of protection: Their enterprises are often relatively more innovative than large market participants as they only survive in the market by compensating for less efficient production with more innovative products. Diversification is a generally accepted economic principle – diversification of risks is a sound economic strategy. Almost no firm relies on one supplier for essential supplies or on a single stock-broker for large stock purchases and sales. Diversification of the software market is similarly desirable. In times in which there are probably more computers than cars on the streets of developed countries, there is an inherent need for a plethora of skilled, well-organized, large scale software developers. This is the only means by which we can avoid the pressure that the winner of the race may otherwise impose on society, and mitigate the risk that the winner's imperfections impose on all of us. One of the most significant problems is that societies lack alternatives. In particular, there is no means to financially assess or insure the damages of dependent companies or consumers caused by lack of diversification in the software market. Even if the risk was enumerable, it would require an extremely liquid capital market and would possibly cause a chain-reaction collapse when the software market fails.

Insofar "price efficiency" in the traditional understanding of the Chicago School **cannot provide justification for anti-competitive acts in the standard software industry**.[231]

On the other hand, without the protection of the **"efficiency"-defense,** the legal system would run the risk of "shooting the winner", with potential harm accruing to both innovation and society. There would be little incentive to innovate if regulators would inevitably destroy a successful market participants' hard-

---

[231] See Jon Polenberg, tfosorciM and croMiftos: Why High-Technology Antitrust Inquiry Is Backwards and Inside-Out, 57 U. Miami L. Rev. 1275 (2003), arguing that the likely intent of antitrust laws was to protect society from any anti-competitive behavior and finding that "high-technology companies are not Robin Hoods, acting with grace of courage, but modern footpads, committing their robberies by stealth".

earned position in the market by absolutely emphasizing the process of rivalry. Such a system would be detrimental to consumer interests not only in the short-term, but also in the long-run. An optimum solution must ensure competition as a process of rivalry (in order to achieve long-run efficiency), while still protecting efficiency gains that further the interest of society in the short-term.

V. Balanced Remedies

The aforementioned considerations reveal that balancing is necessary and inevitable in order to determine sound regulatory principles. However, it remains to be examined how, in detail, such balancing should take place. For good reasons, regulators around the world consider efficiency arguments in competition law. For the reasons pointed out above, however, the criterion of (price)-efficiency is not helpful in assessing conduct in the standard software market. Since the traditional "efficiency defense" does more harm than good in the standard software market, inefficiency should not be considered a pre-condition for regulatory interference with possibly pro-concentration forces (on the "primary level"). Instead, "efficiency" should merely be employed on a "secondary level" as a **limitation on the remedies imposed by courts and regulators**.

While this approach lowers the threshold for regulatory interference, in general, it does not expose the industry to over-regulation, as efficiency considerations will still limit the measures available to regulators. Under this premise, the potential remedies are more limited, but can be imposed under less rigid preconditions, as compared to other more competitive markets. Thereby, a sound regulatory approach for the standard software market exhibits characteristics of the modern approach [232] to traditional "regulated industries" (under the legal basis of antitrust law).

While the basic principles of regulation are insofar clear, limiting the remedies by relying on efficiency grounds remains treacherous ground. These difficulties explain why even the authors of this paper are unable to agree on a common result with respect to *all* practical implications of their approach. We will therefore first provide our shared perspective on regulation before we explain our different approaches with respect to the particular regulatory measures.

---

[232]   See, e.g., Edward Iacobucci, Michael Trebilcock& Ralph Winter, The Canadian Experience with Deregulation (Queen's Univ. Law & Economics Research Paper No.2005-03, March 2005), online: http://ssrn.com/abstract=704841.

A. The Common Ground

We agree that determining a "reasonable" price and thereby controlling a monopolist's pricing does not effectively balance the process of rivalry and efficiency under an indirect regulation regime. These measures are typical of an undesirable system of direct regulation which would not be eliminated by re-labeling the system as "indirect regulation". Governments should also not be required to subsidize inefficient, smaller competitors – e.g. by requiring public employees to use Linux or MacOS.[233] Pushing for a secondary platform without any additional benefits would generally reduce the beneficial network externalities of standard software markets. It would also promote artificial competition which could not survive on its own as the government would be hard-pressed to pick the "best" competitors to support.

Instead, regulators should strive for a **greater level of transparency and interoperability**, because interoperability reinforces incentives to innovate and benefits from already established standards. Since standards are effective due to network externalities (which are a demand-side, hence a customer-related effect), firms have no justification for privatizing customer behavior. To that end, governments should promote **open standards** and require mandatory **disclosure of proprietary interfaces** of dominant market participants to eliminate incumbents' taking advantage of network externalities. This also implies a reduction of legal incentives to use trade secret protection through the creation of incentives to disclose those secrets (for example, through granting longer rights to claim royalties on certain intellectual property rights or the periodical protection from antitrust scrutiny). The opening of interfaces does not unreasonably harm a dominant market participant. The valuable innovation does not lie in the interfaces, but in the technology that can be accessed through them. Insofar even an open interface would require competitors to re-implement the underlying functionality. This would enable third parties to develop complementary products, whereas creating replacements would usually be subject to significant costs and efforts as the original product would remain with its inventor.[234]

While one might argue that there is little incentive to buy a "copy" if the "original" is available for the same price or less it must be noted that data-compatibility does not necessitate feature-parity or an identical

---

[233]   See David S. Evans, Bernard J. Reddy, Government Preferences for Promoting Open-Source Software: A Solution in Search of a Problem, 9 Mich. Telecomm. Tech. L. Rev. 313 (2003).

[234]   For example even though most Windows-programming-interfaces (APIs) are documented, the Open-Source-Reimplementation WINE (http://www.winehq.com/site/about) is still struggling with the huge amount of functionality after 12 years. Similarly even though the user interface of Windows is widely visible the Open Source community has only recently begun providing applications with a comparable appearance.

user-interface. For example, many (non-Microsoft) Office programs now share the OASIS file format, even though only some support a basic feature-set and each has a completely different user-interface. Further, we suggest that any modification of the standard must be published well in advance to allow dependent competitors to adjust their products to the changes. Violations of disclosure requirements must be severely sanctioned in order to deter incumbents (which otherwise have incentives to cheat) and to avoid (other) information asymmetry problems that could potentially negatively affect consumer choice.

Adjudicators should also be open to a consideration of **"must-carry" rules**[235] to provide competitors with access to standard software as a distribution medium for their innovations. "Must-carry" rules require firms that are distributing one (typically less attractive) product together with another (typically, but not exclusively) attractive product[236] to offer competing products at identical conditions as the unattractive product in the existing bundle (which generally requires that the bundled competing product is available for free) – i.e. imposing an equal access "must-carry" regime. We believe those rules to be beneficial, because including additional third party products in dominant products does not limit the benefits that a common platform provides to developers of complementary products. Instead, it actually expands the available shared features, allowing for even more flexibility. Nowadays neither the distribution media (especially when using DVDs) nor the user's local storage (i.e. hard disk) poses a significant restriction, as limited capacity for data storage formerly did. Insofar "must-carry" rules allow competitors to make use of a dominant software program's distribution mechanism. At least in the media player market, bundling of multiple manufacturers' products with an operating system should be easy as only three main competitors distribute their streaming media players for free.[237] When imposing such a "must-carry" rule, regulators must ensure that no product is "preferred" on the end-user's machine, which may (cumulatively) require the making of all such products non-removable, the hiding of direct access buttons, and the imposition of random rotation of the applications that are used for a specific task (playing movies, surfing the Internet, exchanging email, etc.). Again, violations must be punished severely, in extreme cases also with personal sanctions against management.

---

[235] See Randal C. Picker, Unbundling Scope-of-Permission Goods: When Should We Invest in Reducing Entry Barriers?, 72 U. Chi. L. Rev. 189 (2005) but also compare his earlier view advocating "mandatory versioning", Randal C. Picker, Pursuing a Remedy in Microsoft: The Declining Need for Centralized Coordination in a Networked World, 158 J Institutional & Theoretical Econ 113, 115-16 (2002).

[236] This is not limited to dominant products as the current flood of "Spyware" bundled with otherwise free utility-programs (such as sound-recorders or files-haring software) shows.

[237] Apple's Quicktime, Microsoft's Media Player and Real's Realplayer.

Regulators should also seek to prevent **mergers** in the standard software market as it is likely that these mergers are almost never associated with "true" efficiency gains on the "production level" of the standard software markets – a fact yet widely missed in the current discussion. The "production level" in software consists of the software teams which are typically many small units.[238] The potential synergies of vertical integration of these software developer units are very low: Each software tool has to expose sufficient functionality to create a "platform" that is useable and supported by third parties as no single firm can create customary solutions for all potential users. Even in a single large company, different divisions generally depend on these disclosed interfaces and share little if any resources in day-to-day work.[239] The advantages of a larger development team are usually irrelevant as the number of (usually highly specialized) developers employed in a single firm is usually countered by the costs of coordination, thereby in effect creating multiple small development units under a common cloak. The Open Source community and the successful development of Unix by free developer groups exemplify this point. This does not exclude that there are efficiency gains if mergers happen. Besides the fact that these synergies are, generally speaking, very hard to measure and even harder to prove or disprove, competing manufacturers may always account for mergers by demonstrating efficiency gains on the administrative, distribution, or financial level of the firm (rather than the productive level).

Further, since mergers reduce the number of suppliers in any market, the results are especially severe in the software market, where mergers primarily take place between companies that have a strong position in the market. Since the natural monopoly nature of the market makes new entry very hard, it is unlikely that a new firm will take the position of one of the existing companies. When two firms supplying comparable products (like Adobe and Macromedia) merge, only one "code base" will continue to be developed. Any know-how that is embedded in the discontinued (and undisclosed) source-code will vanish. The existing users of the cancelled product will often not be able to get improvements for their programs and are required to switch to a different product (and re-learn the user-interface, convert their documents, etc.). While this risk can be eliminated by licensing the cancelled product as Open Source (or even committing it to the Public Domain), this does, generally speaking, not happen.[240] The investment into the old, non-continued codebase must insofar be considered a complete social loss as no re-use is possible. Thus, a stricter merger control

---

238   *.

239   *.

240   As mergers are often intended to reduce competition which would persist if one of the desired products would continue to be developed.

approach than the current methods is justified. This approach includes assessing the likely fate of the different platforms in question and requiring the merging firms to commit the abandoned product to the Public Domain if few serious competitors remain in the market.

If, from the product quality perspective, bundling different developing units under the ceiling of a firm is not necessary, adjudicators should also not hesitate to split up firms that, as a "bundle" of development units, threaten competition. Consequently, unlike traditional economic theory, we do not strongly oppose **breaking up large software companies**. These structural measures are likely to correct the conflicts of interest caused by one company serving different market levels. These risks are especially high in the standard software market where competition works on several levels (operating system, development tools and end-user productivity software). In those markets, a vertically integrated company might be tempted to abuse its power in one sector to expand into the other.

In addition, we would generally require **closer scrutiny** of the actions of market leaders in order to strengthen the process of rivalry in the software market. In order to effectively enforce antitrust laws, international cooperation is required.[241] In addition, special task forces or committees must be able to permanently access and review the relevant source code and address complaints by competitors.[242]

Furthermore, on a highly abstract level, governments already regulate the software market by defining the **scope of intellectual property rights**. In the future, those limitations of intellectual property might adjust in order to better coordinate the conflicting rules with antitrust law, as exemplified by the "misuse" doctrines in copyright and patent law. For example, the judiciary and the legislature should consider granting smaller competitors further leeway in reverse-engineering a dominant firm's products (thereby effectively extending the generally accepted "misuse-doctrines" of copyright and patent-law into the realm of trade secrets).[243] In addition, compulsory licensing on the basis of the essential facilities doctrine should be expanded and the barriers to achieve such licenses reduced.

---

[241] For example, it is highly unlikely that "Windows N" as required by the European Commission will ever be available in the United States. While European authorities relied on the U.S. settlement regarding Microsoft's actions in the middleware-market, it is unlikely that there will be antitrust suits in the U.S. dealing with Media Player and middleware.

[242] Like the technical committee created under the settlement between Microsoft and the Department of Justice consisting of one member appointed by Microsoft, one member appointed by the government and one jointly selected member, see Settlement, IV.B.8.c or the Monitoring Trustee appointed by the European Commission, see Commission Decision of 24.03.2004 Relating to a Proceeding Under Article 82 of the EC Treaty (Case COMP/C-3/37.792 Microsoft) at page 1045.

[243] Kelber, *supra* note 18, at 163-164.

Beyond these general principles, the authors disagree with respect to the preconditions to which the aforementioned measures should be subjected, and with respect to the question of whether there should be exceptions to our principle. The latter question particularly regards the beneficial character of certain other measures, which harm efficiency (at least) on the short-run, for the market structure, in general.

B. Michael Beurskens' Effect-Oriented Approach

As pointed out above, regulators must neither slow down an incumbent's drive for innovation nor punish him or her for superior product development or design. Further, "price efficiency" as such is not helpful in assessing market results, while other types of efficiency, such as "innovation efficiency" and "product security efficiency" are hard to measure.[244] Consequently, I am very reluctant to regulate beyond such clear abuses (which do not always coincide with market power in the software market[245]). The case against Microsoft in the United States was easy insofar as it evidenced clearly unjustifiable **abusive behavior** (threats and misrepresentations). Beyond this misconduct, however, Microsoft's aggressive strategy has generally benefited consumers: Microsoft introduced technologically advanced products; its entry into new markets (such as word processing, graphical user interfaces or spreadsheets) has usually caused prices in this market to decrease.[246]

With respect to the preconditions of applying the aforementioned measures, I, in contrast to my co-author, generally doubt that **market power** is a relevant criterion for regulating standard software markets. While the usefulness of market share analysis has been subject to concerns in (general) antitrust cases, it is particularly questionable in standard software markets, since defining the relevant "markets" is very complex: While Microsoft is an important player in the operating system market, only a fraction of its customers run the latest incarnation of its operating system. If Microsoft added a feature in an upcoming version in order to eliminate competitors in a secondary market, it would initially only reach a very small group of enthusiastic users willing to upgrade. As software is extremely durable, many users still (successfully) employ Windows 95 (a 10 year old operating-system) or Office 97 (an 8 year old version of Microsoft's productivity software) on their aging computers and have little incentive to upgrade. In addition, the use of software is heavily

---

[244]   If such issues can be determined, at all.

[245]   Microsoft's misrepresentations regarding Java-compatbility were not founded on market power, their position with regards to Java development systems and virtual machines was dubious at last.

[246]   See Liebowitz & Margolis, *Winners, Losers & Microsoft*, *supra* note 14.

influenced by illegal copying. The latest statistics suggest that more than a third of all software programs are not purchased.[247] More powerful technical copy-protection, on the other hand, might eventually push large user groups to cheaper or free software or at least keep them from upgrading.

Microsoft's main market for new copies is therefore limited to corporate licenses (which are limited in time) and sales to Original Equipment manufacturers (which are linked to specific hardware that will eventually break and be replaced). It is almost impossible, however, to use those markets to leverage monopoly power as long as Microsoft is itself bound by network externalities. The burden of compatibility with non-updated software (since users must be able to share and exchange data) makes fundamental changes between one product generation and the other impossible. Even in cases where Microsoft designed those standards, their first-mover advantage is slowly diminishing as time goes by. Eventually, competition in the form of Open Source development shared among many firms and individual developers is extremely hard to counter in the long run. Rather than tying regulation to market shares, regulation should therefore focus on the **effects** that it seeks to prevent. I currently observe only three undesirable results in the software market: Intransparency, unnecessary consumer cost differentials by means of leveraging, and loss of existing innovation or innovative potential due to mergers.

Since hiding interfaces furthers the potential capture of network externalities by a single firm, the central element to maintain competition in the standard software market lies in increasing **transparency of the interfaces**. In contrast to my co-author, however, I maintain that mandatory disclosure rules should be imposed on *all* manufacturers, regardless of their market shares. If disclosure is beneficial to society (as evidenced by patent law), there is no justification for making the imposition of the disclosure requirements subject to a certain market share threshold. I nevertheless do not argue in favor of a general public domain by eliminating any legal protection of software. Instead, incentives to innovate should be created by strengthening intellectual property rights in order to eliminate the need for self-help. Strong legal protection will result in (efficiently bargained for) agreements that allow all manufacturers to benefit from network externalities. Such transparency requirements are not entirely new – registration of copyrights has been a long-standing tradition in the United States, and patent law universally requires full disclosure of the invention. Requiring software manufacturers to disclose their source code in order to gain legal protection (against software piracy, i.e. copying of the binary version of the software) seems not overly detrimental. Legal

---

247   Second Annual BSA and IDC Global Software Piracy Study (May 2005), online < http://www.bsa.org/globalstudy/ >.

protection should however not be extended to interfaces as such, which neither constitute an invention (required for patent protection) nor show any level of originality (required for copyright). Of course, transparency as such is not sufficient, as stronger competitors (due to innovative new products or an existing strong market position) might deny licenses to third parties. Insofar an effective and speedy **compulsory licensing** system is also necessary. This may be achieved by employing a central settlement authority or specific court proceedings. The requirements for compulsory licensing should be low – a showing of failed good-faith negotiation should be sufficient, and there should be no need to show the importance of licensing (as this should be recognized in determining the reasonable licensing fee). It can be assumed that such a system will encourage the participants to bargain for an agreement, as both parties must fear unfavorable terms if they are determined by someone else.

Second, regulators must face the danger that software manufacturers can leverage their position in one market very easily into another and thereby focus consumer choice on **non-competitive** criteria **beyond** price and quality. In particular, Microsoft was able to strengthen the market share of its (arguably technologically superior[248]) web browser by bundling it with its most popular product (the Windows operating system). Due to the fact that users had one product readily available, but were required to spend additional time to download and install competing products, they stayed with Internet Explorer and refrained from using Netscape (even though Netscape Navigator was distributed for free much like Realplayer and Quicktime today). The risk of leveraging, however, is effectively remedied by the **"must-carry-rules"** mentioned above. Insofar, the advantage of bundling (leading to lower consumer costs) is, in essence, eliminated. From a policy perspective, "unbundling" (i.e. stopping a "dominant" market participant from integrating new features into his product) does not offer any benefits. Specifically, preventing market leaders from **implementing more and better features** in subsequent revisions of their products will benefit consumer welfare neither in the short-run nor in the long-run.[249] Instead, such measures impose high costs on regulators, courts and

---

[248]   While most computer magazine reviews preferred Internet Explorer 4.0 to Netscape's Communicator, this enthusiasm was not shared by the courts, see e.g. United States v. Microsoft Corp., 87 F. Supp. 2d 30, 40 (D.D.C., 2000): "*Internet Explorer is not demonstrably the current "best of breed" Web browser, nor is it likely to be so at any time in the immediate future*". However the courts at least recognize that Microsoft invested a huge amount of money and personnel to create a superior product.

[249]   For example in 2001 the FCC conditioned the AOL/Time Warner merger, among other things, on the adoption of a server-to-server interoperability standard for their (then market leading) instant messenger, to allow competing service providers to detect the presence of AOL-messenger users on line, *see* the Conditioned Approval of AOL - Time Warner Merger, FCC 01-11 (January 11, 2001) <http://www.fcc.gov/transaction/aol-tw-decision.html>. AOL was not allowed to improve their Messenger until they met this requirement. However, instead of offering transparency, AOL stopped improving their product until it lost its dominant position (to the detriment of its users). Finally, the order was lifted in 2003.

consumers (that cannot be justified by long-term pro-competitive effects). Further, they create divergent and insufficient platforms, imposing high deployment[250] and development[251] costs on any third party trying to provide applications on a software platform. This aspect also eliminates the highly desirable beneficial re-use of existing, standardized frameworks). Due to the abstraction of software, "unbundling" of products has a similar effect as a ban on innovation – as improvements or new features can often not be distinguished from bundled products.

Finally, elimination of competitors (if not competition) by **mergers** should be prevented, for the two reasons pointed out above, which was a lack of efficiency gains, and discontinuance of the "code-base". However, a merger does not exhibit the same negative effects (or may even have beneficial effects), if the future development of the product after the merger is warranted. Thus, I differ directly competing entities on the same product level ("horizontal mergers", e.g. one developer of word-processors buys another developer of word-processors) from mergers across different product levels ("cross market mergers", e.g. an operating system developer buys an application developer).

From a regulatory perspective, **horizontal mergers** in the standard software industry are almost never desirable. Such mergers will usually "substantially lessen competition" as explained above, unless any discontinued / cancelled product is re-licensed as Open Source. Alternatively, its source code (not only the binary) can be committed to the Public Domain. A strict approach towards these mergers is in line with the existing criteria under U.S. and European antitrust law (even though a stricter application by regulators is desirable). Limitations on mergers will not impose an undue burden on the market participants, however, as they can usually acquire the necessary *knowledge* without acquiring *a competitor* (and completely eliminating its product(s)) through the software labor market: The software labor market is very volatile, as there are only a limited number of highly specialized developers. In the absence of patents (or in the presence of extensive cross-licensing) such a "genius-developer" can lead a team to (re-)implement any innovation present at another firm – without violating copyright or trade-secrets. By hiring such developers, a corporation can ensure its leading position in a certain market.[252] Traditionally, antitrust authorities have refrained from

---

[250]   All former platform elements must be "bundled" with the applications, a fact that increases the program size and often eliminates distribution over the Internet.

[251]   Features not included in the platform must either be bought separately or re-developed in house.

[252]   E.g. Anders Heilsberg, designer of Borland's flagship products "Turbo Pascal" and "Delphi" moved to Microsoft to design "Visual J++" and the ".Net-framework" for them. Bill Gates is often quoted for his desire to have all "the best of the best" working for him.

interfering with the labor market. They did so for good reasons: Requiring a genius developer to work for a certain firm or even to prevent a larger firm from hiring him would severely limit the developer's bargaining power (as he cannot threaten his current employer to change firms) and thereby affect innocent third parties. Similarly, an employers desire to determine the qualities of its employees and to create a workable team is one of the core elements of his free business judgment. Regulation of the labor market as part of the regulation of the software market is therefore undesirable (and also hardly feasible, in a liberal economy).

In contrast, **cross market mergers** are less harmful to competition (if the above transparency requirements are in place). In fact, the consolidated marketing power of larger firms might make a formerly non-competitive product by a smaller firm an interesting choice for consumers of the larger firm who formerly depended on an independent supplier.[253] There is little risk for competition when a single software company supplies multiple markets as long as each of its products can be individually inexpensively substituted for by a competing product (which "must-carry" rules seek to achieve): As pointed out above, the advantages of a larger development team are usually irrelevant. In the absence of an information advantage with respect to interface information (due to secrecy or delayed disclosure), a company has little incentive to for expansion into new markets. In case of an abuse of market-power (i.e. when using secret interfaces or intentionally making products incompatible) the traditional rules of antitrust or torts apply, as such clear violations are not problematic to assess (as seen in the Microsoft case).

Consequently, I believe that mandatory disclosure with compulsory licensing, equal access must-carry rules, as well as strict merger-controls combined with mandatory break-ups will counter the three undesirable effects currently observed in the standard software market, and balance efficiency and the process of rivalry.

C. Dirk Zetzsche's Pendulum - Model

Regulators' choice between efficiency-enhancing measures and protection of rivalry is exemplified by, metaphorically speaking, a pendulum that swings forth and back between the extreme points "efficiency" and "process of rivalry": If the risk is significant that no competitor remains, efficiency is a lesser concern as compared to the situation when it is unclear which standard will prevail in the market.

---

[253] For example, Microsoft's acquisition of Frontpage from a small firm granted the formerly unknown product enormous inroads into a market which was formerly dominated by Adobe.

With respect to the criterion under which regulators should subject conduct to antitrust scrutiny, I hold that regulators should still adhere to the safe-harbour rules developed in antitrust law. Even though I admit that agencies sometimes face difficulties in market delineation, the problem will decrease to the same extent as regulators' software expertise increases. At the same time, the safe-harbour regime creates certainty for small manufacturers (who must focus all their attention on innovation to keep up with the big players) and provides incentives to innovate to new entrants. New entrants can reap their development costs more easily if they are not required to disclose their interface information and enable competition on their protocols before their product has gained a significant share in the market.

With respect to the justification for certain regulatory measures, I, in contrast to Michael Beurskens, believe that in certain, but very limited circumstances, regulators should be able to protect the relics of the process of rivalry, even if this reduces the incentive to innovate, and even if this causes harm to consumer welfare, due to lesser product integration. As a second look reveals, only the latter aspect is, in fact, a serious concern. It is simply not logical to assume that new entrants do seriously care about measures that are going to be imposed on them once they have reached a very high market share. Since live as a new entrant is uncomfortable, management and investors have other things to be concerned of, such as product developing, financing and securing survival and/or growth. The same is true if an incumbent firm has significant market power and regulatory interference merely prevents it from gaining *more* market power, as it is compelled to innovate in order to defend its current market share. Conversely, the latter issue, harm to consumer welfare, should be carefully assessed, as pointed out above more in detail with regard to the principle that efficiency limits the scale of potential remedies available to regulators. In contrast to my co-author, however, I maintain that in extreme cases there should be an *exception* to the principle: On an *ultima ratio* basis, even harmful regulatory measures should be legal, if these measures *retain relics of competition.*

Bearing these considerations in mind, a three step model is most reflective of the concerns that we have raised with respect to the balance between efficiency and process of rivalry. The **first level of regulatory measures** must be imposed when the firm reaches the **threshold for dominant firms under antitrust law**, which – in the absence of exceptions - is commonly associated with a market share of approximately 30%. A firm that reaches the 30% threshold has not yet prompted network externalities that push aside consumer demand for all other products, but there is a significant risk that we will see these effects set in and that the product will become a standard while its important interface information is proprietary to one manufacturer. Thus, if a firm reaches this market share, regulators need to impose the disclosure and transparency requirements mentioned above. Below this market share, firms can cash in on any intellectual property that

they own without being bothered by antitrust concerns. Besides these measures, regulators should refrain from imposing any measures and let market forces work out which product and / or standard is the best. Obviously, as under the concept presented herein, efficiency is not a justification for mergers, since a dominant firm cannot gain further market power through mergers that affect the market of the same product. Mergers with respect to unrelated markets, however, are still possible.

If the firm nonetheless reaches further ground in the market (which is then likely due to its superior products), the economies of scale in production and network externalities will increase to the same extent as the pricing and innovation-furthering function of product markets is diminished. Then, at a certain point, the dominant firm's market power may prevent consumers from inexpensively accessing competitors' product, and thus indirectly harm consumer choice. In this case, a "must-carry" requirement will give consumers the choice of alternative products at the same cost as the incumbent's product. I deem this second level to be reached at **a market share threshold close to 50%**, though I admit that this threshold is chosen somewhat arbitrarily and needs to be confirmed by econometric methods. Analytical arguments, however, support the magnitude. If the threshold is significantly lower, several products benefit from the economic characteristics of software, but none does so to an extent that threatens competition. If the threshold is higher, chances are that the dominant firm already internalised significant network externalities.

Finally, if the dominant firm reaches **a market share of 90%**, the focus will change from enforcing efficiency to protecting the process of rivalry, as such. In this case, the key concern is to prevent the extinction of competition in neighbor- and after-markets. In order to achieve this goal, under the *ultima ratio* rationale pointed out above, it is justified to require the dominant firm to limit the scope of its incumbent product(s), or – if necessary in order to let some competitors in other markets survive - even require it to **de-bundle certain functions** from its incumbent product and price it separately. Given the economics of software, de-bundling can only achieve the wanted results if the pricing-spread between the dominant product and the de-bundled functions equals the expected costs of *the most efficient competitor* for manufacturing + development + distribution + the capital invested. Requiring the dominant firm to price the product at its *own costs* is unlikely to open the market for rivals. Further, given that software firms are internally structured in many small and mostly independent programming units, it seems to be a mere technical question of whether we achieve de-bundling through de-incorporating (splitting up the firm) or through contractual means. The latter is easier to justify legally, while the former is less expensive to monitor, as cheating across several independent business

entities is risky to all parties involved.[254] Ideally, capital market forces would force the former following the latter, since shareholders can diversify their risk at lower costs than management can do.

If all the aforementioned measures do not slow down the dominant firm's expansion into other markets, regulators should be enabled to limit the dominant firm to its own innovative potential, which could be achieved through totally prohibiting the dominant firm from (1) entering into cross-licensing agreements with other firms; (2) hiring expert-software developers from competitors; and (3) outright acquiring a competitor's product. Once these measures are in place, capital markets / shareholders would press the dominant firm's management for higher dividends, due to a lack of use for cash at the firm level, besides research & development (which is good). Interaction of tax laws (higher taxes for retained distributions), capital market pressure and indirect product market regulation may then possibly achieve overall beneficial results.

While the *ultima ratio* measures would impede the (efficient?) transfer of technologies from rivals to the dominant firm (but neither from one rival to another, nor from the dominant firm to rivals), they help avoid the situation that bureaucracies are required to set input and output levels – a state which Michael Beurskens and I both deem to be an even less efficient state than keeping some smaller / "inefficient" competitors alive. Thus, rather than protecting rivals as such, my approach aims at preserving relics of the pricing and innovation-furthering function of product markets. Under this concept, currently inefficient competitors function as cells for the future growth of new competition in the market. This is deemed to be a more "efficient" regulatory approach than dealing with the alternative - a plethora of state-administered monopolies.

---

[254] If B is the benefit from cheating and in order to cheat you need N people to collude, and P is the cost (penalty) for each colluding party, it would be rational to collude only if $B/N > P$; as N grows the utility from collusion decreases.

VI. A Preliminary Result

At first glance, there seems to be little need for a change in regulatory policy: The "social welfare" criterion guiding the construction of §§ 1 - 2 of the *Sherman Act* and Articles 81 - 82 of the EC Treaty provide sufficient discretion for the peculiarities of the standard software market. Nonetheless, both systems reach quite different results.

The European Union has started enforcing rivalry by means of direct regulation of the standard software market. In particular, the European Commission seeks to promote competition as a process of rivalry beyond plain "efficiency": "Competition benefits consumers in terms of choice of goods and services, availability of innovative products and price. Without competitors, there is no competition. End of story."[255] It thereby emphasizes that Article 82 covers both *direct and indirect effects* on the competitive structure. This interpretation allowed the Commission to overrule Microsoft's argument that its monopolizing behavior actually benefits consumers.[256] The pending appeal of the antitrust measures before the Court of First Instance and probably the European Court of Justice will reveal whether this rivalry-oriented approach will eventually prevail.

In contrast, **U.S.**-**Courts** have thus far assessed software under the rules for competitive industries. Under the traditional distinction[257] between "regular competitive industries" and "natural monopoly industries" (i.e. "regulated industries", which cover those industries in which market forces typically do not work), the DoJ and the courts treat the market for standard software under the former rules while – from an economic point of view - it should be subject to the latter.

The results presented herein cannot be more than preliminary as the future legal developments are still unpredictable. Further, neither the goals nor the means of the current efforts are clear: Determining efficiency is problematic as such; when efficiency competes with the goal of free rivalry results are completely unclear. While lawyers and economists have debated these issues for many years, a closer focus on the peculiarities of the standard software market might be of assistance in developing better methods in balancing both goals. At

---

[255]   European Commission, Questions & Answers re the CFI-Order (22 Dec 2004), online: http://europa.eu.int/rapid/pressReleasesAction.do?reference=MEMO/04/305&format=HTML&aged=0&language=EN &guiLanguage=en .

[256]   EC, Microsoft-decision, *supra* note 6., at [704.], citing *Hoffmann-LaRoche*, *supra* note 116, at [125.].

[257]   This is, for example, evidenced by the relatively strict judicature on the "essential facilities doctrine." See *supra* notes 172 et seq.

least, it should be clear that, with respect to the market for standard software, the time for ignoring competition as a "process of rivalry" in the name of "efficiency" is definitely over.