Notre Dame Law Review

Volume 82 | Issue 2

Article 2

12-1-2006

Setting Software Defaults: Perspectives from Law, Computer Science and Behavioral Economics

Jay P. Kesan

Rajiv C. Shah

Follow this and additional works at: http://scholarship.law.nd.edu/ndlr

Recommended Citation

Jay P. Kesan & Rajiv C. Shah, *Setting Software Defaults: Perspectives from Law, Computer Science and Behavioral Economics*, 82 Notre Dame L. Rev. 583 (2013).
Available at: http://scholarship.law.nd.edu/ndlr/vol82/iss2/2

This Article is brought to you for free and open access by NDLScholarship. It has been accepted for inclusion in Notre Dame Law Review by an authorized administrator of NDLScholarship. For more information, please contact lawdr@nd.edu.

SETTING SOFTWARE DEFAULTS: PERSPECTIVES FROM LAW, COMPUTER SCIENCE AND BEHAVIORAL ECONOMICS

Jay P. Kesan[*]
Rajiv C. Shah[†]

© 2006 Jay P. Kesan & Rajiv C. Shah. Individuals and nonprofit institutions may reproduce and distribute copies of this Article in any format, at or below cost, for educational purposes, so long as each copy identifies the author, provides a citation to the *Notre Dame Law Review*, and includes this provision and copyright notice.

[*] Professor, College of Law and the Department of Electrical & Computer Engineering, University of Illinois at Urbana-Champaign.

[†] Adjunct Assistant Professor, Department of Communication, University of Illinois at Chicago.

The authors would like to thank Matthew Kramer, Betsy Palathinkal, and Shyama Sridharan for their research assistance. The authors would also like to thank Greg Vetter and Allen Friedman for their useful comments and suggestions.

This material is based upon work supported by the National Science Foundation under Grant No. IIS-0429217. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

INTRODUCTION

An infusion pump at a hospital lost its battery charge and was plugged into a wall outlet to ensure continued operation. But when plugged in, the infusion rate switched from 71 mL/hr to 500 mL/hr![1] Such an increase could easily cause fatal overdose in a patient. To prevent this defect, the pump software was revised to include a default set at zero for rate and volume settings as well as the inclusion of a "check settings" alarm.[2]

People from around the world were able to peer into the girl's locker room at Livingstone Middle School.[3] The school had installed Axis cameras as a security measure. What they didn't do was change the default password on the cameras. Because the default password, "pass," is well known, anyone could view the images. This could have been prevented if every camera had a unique password or forced each user to change the password during setup. Instead, the manufacturer knowingly opted to do nothing.[4]

[1] Adverse Event Report, U.S. Food & Drug Admin., Abbott Laboratories Lifecare Infusion Plum XL Pump Infusion Pump (Oct. 1, 1999), http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfMAUDE/Detail.CFM?MDRFOI__ID=251892. There are numerous examples like this in the FDA's Manufacturer and User Facility Device Experience Database, http://www.fda.gov/cdrh/maude.html.

[2] Adverse Event Report, *supra* note 1.

[3] Patrick Di Justo, *On the Net, Unseen Eyes*, N.Y. TIMES, Feb. 24, 2005, at G1 (writing about a lawsuit filed by students who had visited Livingston Middle School).

[4] *Id.*

Over two-thirds of the people who use computers were concerned with cyber-security in 2000.[5] Two of the four bestselling software titles in 2003 were system utilities and security products.[6] You would expect that the informed and motivated individuals who bought these products would have secure computer systems. However, in-home studies of computers have found considerable security deficiencies. The most recent study conducted in December 2005 found that eighty-one percent of home computers lacked core security protections, such as recently updated anti-virus software, properly configured firewall and/or spyware protection.[7] The explanation for this discrepancy between people's security concerns and their computer's common security defects is best explained by users' inability to properly configure security software despite their best efforts.

In each of these three examples, default settings play a crucial role in how people use computers. Default settings are pre-selected options chosen by the manufacturer or the software developer. The software adopts these default settings unless the user affirmatively chooses an alternative option. Defaults push users toward certain choices. This Article examines the role of software defaults and provides recommendations for how defaults should be set. Our hope is that proper guidance will ensure that manufacturers and developers set defaults properly, so as to avoid the kind of problems encountered with the infusion pump or the security camera, while also making it easier for users to properly configure their computers to vindicate their security or privacy preferences.

This Article takes off from the recognition by scholars that software has the ability to affect fundamental social concerns, such as

[5] Press Release, Info. Tech. Ass'n of Am., New Nationwide Poll Shows Two-Thirds of Americans Worry About Cybercrime; Online Criminals Seen as Less Likely To Be Caught (June 19, 2000), http://www.itaa.org/infosec/release.cfm?ID=285.

[6] Press Release, NPD Techworld, NPD Group Reports Overall Decrease in PC Software Sales for 2003: Demand for Tax and Security Software Helps Negate Dwindling Sales in Education and Games (Feb. 5, 2004), http://www.npdtechworld.com/techServlet?nextpage=PR_body_it.html&content_id=720.

This trend has not changed. Three of the five top-selling PC software products were security related, and more than half of the top twenty PC software products were security related in September 2005. Press Release, NPD Techworld, Top-Selling PC Software: September 2005 (Oct. 19, 2005), http://www.npdtechworld.com/techServlet?nextpage=PR_body_it.html&content_id=2238 [hereinafter NPD, Top-Selling].

[7] AM. ONLINE & NAT'L CYBER SEC. ALLIANCE, AOL/NCSA ONLINE SAFETY STUDY 2 (2005), *available at* http://www.staysafeonline.info/pdf/safety_study_2005.pdf [hereinafter ONLINE SAFETY STUDY].

privacy and free speech.[8] Scholars and software developers equally recognize that it is possible to proactively design software to address issues such as crime,[9] competition,[10] free speech,[11] privacy,[12] fair use in copyright,[13] and democratic discourse.[14] This approach relies on the ability of policymakers to manipulate (or create an environment to manipulate) software settings. In other words, software possesses characteristics that can be relied upon to govern. We have highlighted several of these governance characteristics of software,[15] which are analogous to "knobs and levers" that policymakers can manipulate to favor specific values or preferences. Just as policymakers influence behavior by manipulating incentives and penalties through subsidies and fines, they can also influence user behavior by manipulating the

[8] *See* STUART BIEGEL, BEYOND OUR CONTROL? 187–211 (2001) (discussing software-based regulation); LAWRENCE LESSIG, CODE AND OTHER LAWS OF CYBERSPACE 24–29 (1999) (describing the role of architecture); Michael J. Madison, *Law as Design: Objects, Concepts, and Digital Things*, 56 CASE W. RES. L. REV. 381, 414–19, 425–30, 440–47, 463–75 (providing a sophisticated account of the role of materiality as it relates to software regulation); Joel R. Reidenberg, *Lex Informatica: The Formulation of Information Policy Rules Through Technology*, 76 TEX. L. REV. 553, 557–58 (1998); *see also* Sandra Braman, *The Long View*, *in* COMMUNICATION RESEARCHERS AND POLICY-MAKING 11 (Sandra Braman ed., 2003) (urging communications scholars to study how technology affects fundamental societal issues).

[9] *See, e.g.*, Neal Kumar Katyal, *Criminal Law in Cyberspace*, 149 U. PA. L. REV. 1003, 1102–06 (2001).

[10] *See, e.g.*, Mark N. Cooper, *Anticompetitive Problems of Closed Communications Facilities*, *in* OPEN ARCHITECTURE AS COMMUNICATIONS POLICY 155, 161 (Mark N. Cooper ed., 2004), *available at* http://cyberlaw.stanford.edu/blogs/cooper/archives/openarchitecture.pdf.

[11] *See, e.g.*, Lawrence Lessig & Paul Resnick, *Zoning Speech on the Internet: A Legal and Technical Model*, 98 MICH. L. REV. 395, 399 (1999); Jonathan Weinberg, *Rating the Net*, 19 HASTINGS COMM. & ENT. L.J. 453, 456–59 (1997).

[12] *See, e.g.*, William McGeveran, *Programmed Privacy Promises: P3P and Web Privacy Law*, 76 N.Y.U. L. REV. 1812, 1826–27 (2001) (arguing for the Preferences for Privacy Project (P3P) as a solution to privacy problems).

[13] *See, e.g.*, TARLETON GILLESPIE, WIRED SHUT (forthcoming Spring 2007) (analyzing the role of digital rights management software); Dan L. Burk & Julie E. Cohen, *Fair Use Infrastructure for Rights Management Systems*, 15 HARV. J.L. & TECH 41, 47–54 (2001) (providing an example of an architectural solution to allow fair use in digital-based intellectual property).

[14] *See, e.g.*, ANTHONY G. WILHELM, DEMOCRACY IN THE DIGITAL AGE 44–47 (2000) (discussing how to design a democratic future); Cathy Bryan et al., *Electronic Democracy and the Civic Networking Movement in Context*, *in* CYBERDEMOCRACY 1, 6–8 (Roza Tsagarousianou et al. eds., 1998) (providing a number of examples for using electronic resources for stimulating democratic discussion and growth).

[15] Rajiv C. Shah & Jay P. Kesan, *Manipulating the Governance Characteristics of Code*, INFO, Aug. 2003, at 5–8.

design of software.[16] This Article continues this line of inquiry by focusing on the role that default settings play in software development and use.

Default settings appear in a variety of contexts; for example, in *Preferred Placement*,[17] several authors explore how default settings for privacy,[18] portals,[19] and search engines[20] affect how people use the Web. As an example, consider that the most valuable part of Netscape was not its software, but its default setting for its home page. Because a large number of users (estimated at forty percent) never changed this default setting, Netscape's home page had enormous popularity.[21] Analysts touted the importance of this default home page (a top ten Website at the time) when AOL purchased Netscape for about four billion dollars.[22] The economic significance of this default setting highlights the power of defaults. Defaults play an important role in virtually every important decision users make online. These decisions have ramifications in areas such as privacy and security and involve software in diverse products such as Web browsers, operating systems, and wireless access points.

Default settings are not a creation of the Internet. Legal scholars and behavioral economists have long studied the role of default settings, albeit not software defaults.[23] Research by behavioral economists has studied the deference to defaults in decisions regarding

[16] *See* Dan L. Burk, *Legal and Technical Standards in Digital Rights Management Technology*, 74 FORDHAM L. REV. 537, 546–47 (2005) (discussing the use of design-based software regulation).

[17] PREFERRED PLACEMENT (Richard Rogers, ed., 2000).

[18] Greg Elmer, *The politics of Profiling*, *in* PREFERRED PLACEMENT, *supra* note 17, at 65, 69–72 (discussing privacy concerns raised by internet browsers' activation of cookies without informing internet users).

[19] Richard Rogers and Ian Morris, *Operating the Internet with Socio-Epistemological Logics*, *in* PREFERRED PLACEMENT, *supra* note 17, at 145, 149–55 (discussing corporations' agreements with web portals-such as America Online in an effort to gain increased exposure by having links to their content placed on Websites to which users are automatically directed).

[20] Lucas Introna and Helen Nissenbaum, *The Public Good Vision of the Internet and the Politics of Search Engines*, *in* PREFERRED PLACEMENT, *supra* note 17, at 25, 31–37 (discussing the process by which search engines rank Websites in their search results and the increased traffic experienced by Websites with a high ranking).

[21] LORRIE FAITH CRANOR & REBECCA N. WRIGHT, INFLUENCING SOFTWARE USAGE 6 (1998), http://xxx.lanl.gov/PS_cache/cs/pdf/9809/9809018.pdf (citing the 40% estimate in their discussion of software defaults).

[22] Douglas Herbert, *Netscape in Talks with AOL*, CNNMONEY.COM, Nov. 23, 1998, http://money.cnn.com/1998/11/23/deals/netscape/.

[23] *See infra* Part I.A.

organ donation and investment saving plans.[24] Their work explains the systematic differences that occur between opt-in and opt-out default plans. Their explanations for the power of defaults focus on bounded rationality, cognitive limitations, and the legitimating effect.[25] These biases are also important for understanding how software defaults operate.

Legal scholarship is another arena that provides a useful analogy for understanding software defaults. For example, the Uniform Commercial Code contains a variety of default rules, such as the implied warranty of merchantability, which apply absent contrary agreement by the parties.[26] Legal scholars have wrestled with questions about what rules should be default rules versus mandatory rules. Contract scholars have focused on the role of consent. Consent is relevant to defaults, since policymakers need to consider whether the parties have freely consented to these defaults or whether they were coerced into accepting the default settings.

At first brush, default settings in software appear to be solely a concern for computer scientists. Computer scientists within Human-Computer Interaction (HCI) have written about how software defaults should be set.[27] However, their approach is almost entirely technical. It focuses on enhancing the performance of software and the efficiency of users. While HCI considers the limitations of users, it lacks a framework for setting defaults for humanistic or societal issues, such as privacy.

Ultimately, we rely on the combination of the three approaches of computer science, behavioral economics, and legal scholarship to provide key insights into understanding how defaults operate. This understanding leads us to focus on how society can harness default settings in software to enhance societal welfare. Sunstein and Thaler have coined the term "libertarian paternalism" to refer to the use of default settings as a method of social regulation.[28] To enable the proactive use of defaults, we offer a general rule for setting defaults in software as well as identifying several circumstances when policymakers should intervene and change default settings. To illustrate this process we have developed several flowcharts that highlight the decisionmaking process. This normative analysis regarding software settings is unique. Many scholars have recognized the power of software,

[24] *See infra* notes 35–37 and accompanying text.

[25] *See infra* Part III.B.

[26] U.C.C. § 2-314 (2006).

[27] CRANOR & WRIGHT, *supra* note 21, at 6–7.

[28] Cass R. Sunstein & Richard H. Thaler, *Libertarian Paternalism Is Not an Oxymoron*, 70 U. CHI. L. REV. 1159, 1171 (2003).

however there is little scholarship that focuses on how software settings should be determined by employing a generalized framework for analysis.

The Article is organized as follows. Part I reviews empirical data on the effectiveness of defaults. This research substantiates the importance and power of defaults. Part II considers a variety of previously mentioned theoretical approaches for understanding default settings. The second part ends by illustrating the limitations of these four approaches by applying them to three controversial uses of software defaults in the areas of competition, privacy, and security. Part III focuses on how defaults should be set. Part of this normative discussion urges that defaults are currently set incorrectly for two technologies (Internet cookies and wireless security encryption) that affect security and privacy. Finally, Part IV discusses how government could influence default settings in software. We do not attempt to catalog all the possible actions by government, but instead show that government is not powerless in dealing with defaults.

Our efforts are aimed at explaining how defaults operate in software and how policymakers should set software defaults. We use the term "policymaker" throughout this Article as a catch-all definition for a wide range of individuals including software developers, executives, policy activists, and scholars who are concerned with the implications of software regulation. After all, there are many parties that are interested in and capable of modifying software.

I. The Power Of Defaults

This Part reviews research on the power of defaults to influence behavior in a variety of contexts. While it is possible for people to change a default setting, there are many situations where they defer to the default setting. This Part shows the impact of their deference to the default setting, not only on the individual, but also on norms and our culture.

Subpart A reviews several academic studies in the context of 401(k) plans, organ donation, and opt-in versus opt-out checkboxes. Subpart B then turns its attention to the power of defaults in software. Our discussion of software provides examples of how defaults affect competition, privacy, and security. These examples illustrate the power of defaults in computer software to influence behavior and are referenced throughout our later discussions on understanding defaults and how best to set them. Subpart C illustrates the wide-ranging effects of defaults in software with an example of a file-sharing software. Finally, subpart D considers how defaults affect society's norms and the creation of culture.

A. *Research on the Power of Defaults*

This subpart reviews three studies that reveal the power of defaults in influencing behavior. In the first study, Madrian and Shea examine the saving behavior of individuals enrolled in a 401(k) savings plan.[29] Initially, the human resources policy default was set so that employees were not automatically enrolled in a 401(k) savings plan.[30] The employer later changed this setting, so that the new default setting automatically enrolled employees. In both circumstances, employees were free to join or leave the program.[31] Contributions ranged from 1% to 15% by the employee with the employer matching 50% of employee contributions up to 6% of employee compensation.[32] The only material difference was the change in the default setting and a default value of 3% employee contribution in the automatic savings plan.[33] This switch in default settings resulted in an increase in participation in the 401(k) savings plan from 37% to 86%![34] Clearly, the default was significant.

A second example that illustrates the power of defaults is organ donation defaults. Countries have two general approaches to organ donation—either a person is presumed to have consented to organ donation or a person must explicitly consent to donation.[35] Johnson and Goldstein analyzed the role of default settings by looking at cadaveric donations in several countries.[36] They found that the default had a strong effect on donations. When donation is the default, there is a 16% increase in donation.[37] Their work shows the power of defaults to influence behavior and how default settings can save lives in certain circumstances (in this case by increasing organ donations).

Bellman, Johnson, and Lohse examined the role of default settings in online checkboxes for opting-in or opting-out of certain prac-

[29] Brigitte C. Madrian & Dennis F. Shea, *The Power of Suggestion: Inertia in 401(k) Participation and Savings Behavior*, 116 Q.J. ECON. 1149, 1151 (2001).

[30] *Id.* at 1151.

[31] *Id.* at 1152.

[32] *Id.*

[33] *Id.*

[34] *Id.* at 1160.

[35] Eric J. Johnson & Daniel Goldstein, *Do Defaults Save Lives?*, 302 SCI. 1338, 1338 (2003).

[36] *Id.*

[37] *Id.* at 1339.

tices.[38] These checkboxes are typically used for privacy settings, junk e-mail settings, and a variety of other simple questions in online forms.[39] In this experiment, participants were asked in an online form whether or not to be notified about future surveys. Participants had to choose between "yes" and "no." When the default was set to "no," only 60% of the participants agreed to be notified later.[40] But when the default was set to "yes," 89% of the participants agreed to be notified later.[41] This difference is quite pronounced and shows how people may defer to a default.

B. *The Role of Defaults in Software*

A default in software is analogous to the defaults described above. A definition for a software default is a pre-selected option adopted by the software when no alternative is specified by the user. Defaults only refer to functions that can be changed by the user. A setting that the user is unable to change is a fixed aspect of the system ("wired in") and is therefore not a default. Developers often use "wired-in" settings for aspects of software that users do not need to modify.[42] The degree to which software can be modified can be seen along the continuum in Figure 1.[43]

FIGURE 1. CONTINUUM OF SETTINGS

Fixed settings————Default Settings————Fully Customizable
"wired-in"   "pushing the user"   "free choice"

The malleability of software means that developers can add, remove, or change default settings. A typical program has tens (and up to hundreds) of defaults that are set by the developer. These defaults may also change over time as developers revise their software. These defaults may be default values, which refer to strings, numbers, or bits that are held in a particular field for input screens or forms. Other defaults include default settings, which are values, options, and choices that are stored and referenced by an application. Finally, de-

[38] Steve Bellman et al., *To Opt-In or Opt-Out? It Depends on the Question*, COMM. ACM, Feb. 2001, at 25, 25.

[39] *Id.*

[40] *Id.* at 26.

[41] *Id.*

[42] *See* Burk, *supra* note 16, at 546–51 (discussing the use of embedded rules in software).

[43] As Greg Vetter has pointed out to us, our analysis is user-centric. From a developer's perspective, there are additional layers of modifiable settings that may appear to the user as wired in.

fault actions are courses of actions that are presented to a user interactively. These defaults often come in the form of alert or confirmation boxes. In this Article, we use the term default or default settings to refer to all three meanings of defaults in software.

The first example for illustrating the power of defaults in software concerns desktop icons on Microsoft Windows operating systems. The issue of which desktop icons to include in a computer's operating system was prominent in the mid-1990s when Microsoft was attempting to catchup to Netscape's Web-browsing software use. Microsoft's internal research found that "consumers tend strongly to use whatever browsing software is placed most readily at their disposal, and that once they have acquired, found, and used one browser product, most are reluctant—and indeed have little reason—to expend the effort to switch to another."[44] In effect, Microsoft recognized that the initial default for Web browsers is crucial for attracting and retaining consumers.

This led to a policy where Microsoft threatened to terminate the Windows license from computer manufacturers that removed Microsoft's chosen default icons, such as Internet Explorer, from the Windows desktop.[45] In one instance, Microsoft threatened Compaq after Compaq entered into a marketing agreement with AOL. Compaq had agreed to place AOL's icon and no other online service icons, such as Internet Explorer, on the desktop of PCs.[46] Microsoft then threatened to terminate Compaq's licenses for Windows 95 if its icons were not restored.[47] At the time, Compaq was the highest-volume original equipment manufacturer (OEM) partner that Microsoft had.[48] Nevertheless, Compaq acquiesced and restored the Internet Explorer icon as a default desktop setting.[49]

Clearly default settings were important for Microsoft and AOL. While we do not know what the value of the setting was to Microsoft or Compaq, we have an idea of how valuable it was to AOL. A few years later, AOL was still pushing manufacturers to add default icons and pop-up ads promoting AOL. AOL was offering manufacturers thirty-

[44] United States v. Microsoft Corp., 84 F. Supp. 2d 9, 47 (D.D.C. 1999).

[45] *Id.* at 59.

[46] *Id.*

[47] *Id.* at 60.

[48] *Id.*

[49] Compaq's behavior led Microsoft to clarify in its contracts with manufacturers that it prohibited changes to the default icons, folders, or "Start" menu entries. *Id.* at 61.

five dollars for each customer that signed up with AOL.[50] To keep this in perspective, Compaq was paying Microsoft about twenty-five dollars for each copy of Windows 95.[51] These numbers suggest that default icons carried significant economic power and are why Microsoft was ready to terminate business with one of its largest customers when it threatened to remove Microsoft's browser from the desktop. While Compaq was intimidated and conceded, Microsoft has continued to battle with competitors such as RealNetworks[52] and Kodak[53] over default settings.[54]

A second example illustrating the power of defaults is the use of cookies technology found in Web browsers. Cookies allow Websites to maintain information on their visitors, which raises privacy concerns.[55] Websites place cookies, small pieces of information, on a visitor's computer. This allows Websites to identify and maintain information on visitors by checking and updating the cookie information. Users can manage the use of cookies through their Web browsers. The default on all Web browsers is set to accept cookies. If consumers want to limit privacy intrusions from cookies, they need to

[50] Alec Klein, *AOL to Offer Bounty for Space on New PCs*, WASH. POST, July 26, 2001, at A1.

[51] Graham Lea, *MS Pricing for Win95: Compaq $25, IBM $46*, THE REGISTER, Jun. 14, 1999, http://www.theregister.com/1999/06/14/ms_pricing_for_win95_ compaq.

[52] RealNetworks filed a billion-dollar lawsuit partly over the fact that Microsoft prohibited providing a desktop icon for RealNetworks. RealNetworks also argued that PC manufacturers were not allowed to make any player other than Windows Media Player the default player. Even if a user chose RealNetworks media player as the default player, Windows XP favored its own media player in certain situations. Evan Hansen & David Becker, *Real Hits Microsoft with $1 Billion Antitrust Suit*, CNET NEWS.COM, Dec. 18, 2003, http://news.com.com/Real+its+Microsoft+with+1+billion+antitrust+suit/2100-1025_3-5129316.html; *Microsoft, RealNetworks Battle*, CNNMONEY.COM, May 2, 2002, http://money.cnn.com/2002/05/02/technology/microsoft/; Andrew Orlowski, *Why Real Sued Microsoft*, THE REGISTER, December 20, 2003, http://www.theregister.co.uk/2003/12/20/why_real_sued_microsoft/.

[53] Kodak considered antitrust action against Microsoft when its software could not be easily made the default option for photo software. Microsoft's motivation was clear—it was planning to charge a fee for images that were sent through Windows to its partners. John R. Wilke & James Bandler, *Shutter Bug: New Digital Camera Deals Kodak a Lesson in Microsoft's Ways*, WALL ST. J., July 2, 2001, at A1.

[54] The issue over pre-installed software on the Windows operating system re-emerged recently with news that Google and Dell are working together to pre-install Google's software onto computers. The reports suggested that in exchange Google is planning to pay Dell one billion dollars over the next three years. Robert A. Guth & Kevin J. Delaney, *Default Lines: Pressuring Microsoft, PC Makers Team Up with Its Software Rivals*, WALL ST. J., Feb. 7, 2006, at A1.

[55] *See* Shah & Kesan, *supra* note 15, at 5 (providing background on the cookies technology).

change the default setting themselves without any interactive prompting.

To understand the implications of the default setting to accept cookies, let us begin by recognizing that Internet users are concerned about online privacy. A Pew Internet & American Life Project study from August 2000 found that 84% of Internet users in the United States were concerned about businesses and strangers getting their personal data online.[56] However, 56% did not know about cookies.[57] More notably, 10% said they took steps to block cookies from their PCs.[58] However, a study by Web Side Story found the cookie rejection rate was less than 1%.[59] These data show that while people were concerned about their online privacy, they were unaware of the most significant technology that affects online privacy. While a small proportion of these people claimed to have changed the default setting, the data actually show that a very small percentage, less than 1%, actually change the default setting. In sum, despite the overwhelming concern for privacy, almost everyone deferred to the default setting and accepted cookies.

A final example on the power of defaults is the use of security settings in Wi-Fi access points (APs). These APs are a common consumer technology for creating wireless networks inside homes and businesses. Shah and Sandvig analyzed the data from hundreds of thousands of APs to understand how people configure their APs.[60] They found defaults programmed into APs to be powerful as half of all users never changed any default setting on their APs.[61]

One particular default setting the study examined was the use of encryption in APs. Encryption is widely recommended as a necessary step for properly configuring an access point. The majority of access points turn off encryption by default, resulting in only about 28% of access points using encryption.[62] However, Microsoft's access points

[56] SUSANNAH FOX ET AL., TRUST AND PRIVACY ONLINE 4 (2000), *available at* http://www.pewinternet.org/pdfs/PIP_Trust_Privacy_Report.pdf (surveying users on online privacy issues).

[57] *Id.* at 3.

[58] *Id.*

[59] Dick Kelsey, *Almost No One Rejects Cookies*, NEWSBYTES NEWS NETWORK, Apr. 3, 2001, http://www.findarticles.com/p/articles/mi_m0NEW/is_2001_April_3/ai_72736309 (discussing a study that measured cookie rejection rate).

[60] RAJIV SHAH & CHRISTIAN SANDVIG, SOFTWARE DEFAULTS AS DE FACTO REGULATION 7–8 (2005), *available at* http://web.si.umich.edu/tprc/papers/2005/427/TPRC%20Wireless%20Defaults.pdf.

[61] *Id.* at 16.

[62] *Id.* at 11.

turn on encryption by default if users follow the CD setup process.[63] As a result, 58% of Microsoft's access points are using encryption.[64] 2Wire also turns on encryption by default in their access points leading to 96% of their access points using encryption.[65] These data show an enormous shift in encryption from 28% to 96% by merely changing the default value.[66]

C. Defaults in Software Affect a Variety of Issues

Default settings in software affect a wide variety of fundamental social policy issues. To illustrate this, we examine the defaults in a popular file-sharing program known as Limewire.[67] Limewire contains several default settings that promote file sharing. Although the main purpose of the program is file sharing, there are several default settings that affect a variety of fundamental societal concerns.

The first default setting in Limewire sets the upload bandwidth default to 100%. This setting promotes using all of the computer's available bandwidth for file sharing. Another default setting sets the program to automatically connect to the network when the application starts up. This ensures that file sharing starts immediately. A third default setting treats users with fast computers and Internet connections as an "ultrapeer." An "ultrapeer" helps other users download faster, but demands a greater load on the user's computer. All three of these default settings are used to promote file sharing. However, these are not the only defaults in Limewire.

Limewire uses default settings for filtering search results by specific words, adult content, or file types. This setting affects free speech, essentially censoring certain Websites from its users. Other default settings define the community of file sharers. Limewire has a default setting to share files only with people who are sharing files. Users can set the minimum number of files an uploader has to share. This feature defines the community's boundaries. It can exclude "freeloaders" or people sharing only a few files. Limewire sets the default to one file and, thus, effectively allows everyone (including "freeloaders") to share files. Finally, there is a default affecting social communication determining whether the chat feature is on or off.

[63] *Id.* at 12.

[64] *Id.* at 11.

[65] *Id.* at 12.

[66] *Id.* at 11.

[67] This subpart is based on our study of the Limewire file sharing program. The observations are based on Limewire Basic Client version 2.1.3.

Limewire's use of defaults demonstrates how defaults can affect a wide variety of issues. As a matter of policy, defaults are good for a number of reasons. First, defaults provide users with agency. Users have a choice in the matter: They can go with the default option or choose another setting. Second, a default setting guides the user by providing a recommendation. However, there may be situations where users do not need or should not have options. We discuss these situations in more detail later, but the key point is sometimes we do not want to give a user choices.

D. *Cultural Context of Software Defaults*

Defaults are important not only in affecting a person's actions, but also in shaping norms and creating culture.[68] This occurs in two general ways. First, defaults can serve to reinforce and amplify existing norms. A simple example is that people know they should save money. However, they often neglect to save on a day-to-day basis. This led Thaler and Benartzi to craft a savings program that takes advantage of people's deference to defaults.[69]

Second, new communication technologies often incorporate defaults (sometimes unintentionally) that have cultural ramifications. For example, consider the defaults in Wi-Fi technology that limit security. While these defaults limit security, they aid the creation of a larger cultural movement toward the sharing of wireless networks and the development of community wireless networking. As Sandvig notes, the "mushrooming of free APs . . . was the result not of a conscious altruism, it was the triumph of unreflective accidents."[70] The accident here is that when a user takes an AP out of its packaging and starts using it, it becomes open and free to others by default and not by the conscious action of its owner.

There is a subtle but profound concern that default settings will not be seen as defaults but accepted as unchangeable. After all, if people don't know about defaults, they will assume that any alternative settings are impossible or unreasonable. This influence on peo-

[68] *See, e.g.*, Matt Ratto, *Embedded Technical Expression: Code and the Leveraging of Functionality*, 21 INFO. SOC'Y 205, 207–11 (2005) (discussing how software embeds expression in several ways while also expressing appropriate methods for doing tasks).

[69] Richard H. Thaler & Shlomo Benartzi, *Save More Tomorrow: Using Behavioral Economics to Increase Employee Saving*, 112 J. POL. ECON. S164, S170–71 (2004) (proposing the Save More Tomorrow savings plan that increases the contribution rate in conjunction with raises, therefore relying on people's inertia to lead them to save at higher rates).

[70] Christian Sandvig, *An Initial Assessment of Cooperative Action in Wi-Fi Networking*, 28 TELECOMM. POL'Y 579, 591 (2004) (discussing the growth of the Wi-Fi networking).

ple's perception of their control over software configuration is a core concern with software regulation. This concern arises with the use of filtering software. Everyday users will not notice Websites that are blocked out, such as Websites presenting information on breast cancer or AIDS.[71] Instead, they will just assume there is no information on that topic or that the topic is unimportant. This can have a striking effect on a person's view and use of culture. This effect is the result of software creating an artificial and unknowable barrier.[72] We discuss this issue further in Part III, focusing on how best to set defaults.

II. UNDERSTANDING DEFAULTS

Once defaults are recognized as powerful in influencing people's behavior, the next issue is to explain why people are swayed by default settings. In this Part, we offer four different perspectives based on extant scholarship for understanding or theorizing the effect of defaults on people's behavior and choices. Additionally, we offer another perspective from our investigations into software defaults. Subpart A focuses on work within computer science in the field of Human-Computer Interaction (HCI). Subpart B examines the work of behavioral economists. Subpart C considers the work of legal scholars, largely those focusing on defaults in contract law. Subpart D offers a perspective on technology defaults from a health communication approach. Finally, subpart E considers the role of technical sophistication for explaining why people may defer to default settings.

A. *Human-Computer Interaction (HCI) Theory*

Scholars within the Human-Computer Interaction (HCI) subfield of computer science have developed theories and conducted research on how people use computers. The most direct work on defaults has been done by Cranor and Wright.[73] As an example, Cranor's group gave careful thought to the default settings in their design of the

[71] VICTORIA RIDEOUT ET AL., KAISER FAMILY FOUND., SEE NO EVIL: HOW INTERNET FILTERS AFFECT THE SEARCH FOR ONLINE HEALTH INFORMATION 6–10 (2002), *available at* http://www.kaisernetwork.org/health_cast/uploaded_files/Internet_Filtering_exec_summ.pdf (finding that software filters affect the ability of people to find health information online).

[72] Lee Tien, *Architectural Regulation and the Evolution of Social Norms*, 7 YALE J.L. & TECH. 1, 18 (2004) (discussing whether software is an appropriate regulatory tool).

[73] CRANOR & WRIGHT, *supra* note 21, at 6–7 (discussing the role of defaults and wired-in settings for software designers).

AT&T Privacy Bird, which is a Web browser plug-in that notifies users about a Website's privacy policy.[74] While there is little research by computer scientists directly on defaults, defaults have been considered in the context of system design and user customization. This subpart reviews this research and then applies it to several examples of software defaults in order to determine their usefulness for establishing public policy regarding software defaults.

The user customization research focuses on how users tailor software to their needs. This work is relevant because when users customize software they are usually changing default settings. The principle findings are that people are more likely to customize a software program as their experience with computers and time with the software program increases.[75] The research has shown that while users often change some software features, they often limit themselves to changing the minimum necessary to use the software.[76] Mackay recognizes this as "users 'satisfice' rather than optimize."[77] While theoretically users could carefully evaluate every possible option to customize, they do not act that way. Instead, users view customization as time consuming and troublesome and, therefore, avoid customizing software.

The principles of system design illustrate how software developers set defaults. As a starting point, it is useful to review the general principles for user interfaces. One set of common sense guidelines comes from researchers at IBM. They believe the interface should: 1) be similar to known tasks; 2) protect the user from making mistakes; 3) be easy to learn; 4) be easy to use; 5) be easy to remember; and 6) provide fast paths for experienced users.[78] Once we understand these guidelines, we can see why researchers like Dix believe that *defaults*

[74] Lorrie Faith Cranor et al., *User Interfaces for Privacy Agents*, 13 ACM TRANSACTIONS ON COMPUTER-HUM. INTERACTION 135, 143–57 (2006) (providing a case study on developing software that addresses privacy concerns).

[75] Mary Beth Rosson, *Effects of Experience on Learning, Using, and Evaluating a Text Editor*, 26 HUM. FACTORS 463, 473–74 (1984).

[76] *See* Stanley R. Page et al., *User Customization of a Word Processor*, *in* HUMAN FACTORS IN COMPUTING SYSTEMS 340, 344–45 (Michael Tauber ed., 1996).

[77] Wendy E. Mackay, *Triggers and Barriers to Customizing Software*, *in* PROCEEDINGS OF THE SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS 153, 159 (Scott P. Robertson et al. eds., 1991), *available at* http://insitu.lri.fr/~mackay/pdffiles/CHI91.Triggers.pdf.

[78] Edward J. See & Douglas C. Woestendiek, *Effective User Interfaces: Some Common Sense Guidelines*, *in* PROCEEDINGS OF THE 5TH ANNUAL INTERNATIONAL CONFERENCE ON SYSTEMS DOCUMENTATION 87, 88 (Virginia DeBuys ed., 1986), *available at* http://doi.acm.org/10.1145/318723.318738 (follow "PDF" hyperlink) (discussing guidelines for developing a user interface).

"can assist the user by passive recall . . . . It also reduces the number of physical actions necessary to input a value. Thus, providing default values is a kind of error prevention mechanism."[79] Similarly, Preece writes that "the default value is usually the most frequently used or safest option, indicated by a thickened border around a button, or some similar visual device."[80] Furthermore, consider industry guidelines on defaults, such as the Apple Human Interface Guidelines. It states:

> The default button should be the button that represents the action that the user is most likely to perform *if* that action isn't potentially dangerous. . . .
>
> Don't use a default button if the most likely action is dangerous—for example, if it causes a loss of user data. When there is no default button, pressing Return or Enter has no effect; the user must explicitly click a button. This guideline protects users from accidentally damaging their work by pressing Return or Enter. You can consider using a safe default button, such as Cancel.[81]

There are two core principles in all three approaches described above (Dix, Preece, and Apple) for setting defaults. The first principle is that the default should be set to a value appropriate for novice users. An application of this is seen in Cranor's work on the Privacy Bird software when it considers novice users by recognizing that changing defaults can be time consuming and confusing, because users risk "messing up" their software.[82] The second principle is that the default should be set to a value that will improve efficiency. Efficiency could be a sensible value, a value least likely to cause errors, or "[w]hat do people enter or choose most often."[83]

[79] ALAN DIX ET AL., HUMAN-COMPUTER INTERACTION 173 (2d ed. Prentice Hall Eur. 1998) (1994) (discussing the role of defaults).

[80] JENNY PREECE ET AL., HUMAN-COMPUTER INTERACTION 298 (1994); *see also* SUSAN L. FOWLER & VICTOR R. STANWICK, THE GUI STYLE GUIDE 19 (1995) (encouraging use of defaults as a time saving device in data-entry programs in which a certain result is overwhelmingly more common than others). In the context of privacy, Beckwith argues that since users trust computer systems to be benign, the defaults should be set conservatively. The defaults should also be understandable and well defined so that users can depend on them. Richard Beckwith, *Designing for Ubiquity: The Perception of Privacy*, PERVASIVE COMPUTING, Apr.-June 2003, at 40, 46 (2003).

[81] APPLE COMPUTER, INC., APPLE HUMAN INTERFACE GUIDELINES 214 (2006), *available at* http://developer.apple.com/documentation/UserExperience/Conceptual/OSXHIGuidelines/OSXHIGuidelines.pdf.

[82] Cranor et al., *supra* note 74, at 54.

[83] SUSAN FOWLER & VICTOR STANWICK, WEB APPLICATION DESIGN HANDBOOK 79 (2004).

Now that we have determined the two core principles (consider novice users and efficiency) for computer scientists, the next step is applying them to our examples. The first example concerns default icons on the desktop of Windows operating systems. HCI suggests that default icons should be set up for the most common programs and for programs and features most used by novices. Because a Web browser is an important feature, it would make sense to include an icon for one. The question becomes whether icons for two competing browsers would confuse novices or increase efficiency by allowing users to select the browser they need. This is a difficult determination and requires user testing to determine the better outcome. Note that the HCI approach does not address the issue of competition.

The second example concerns the privacy risks of enabling cookies. The principle of protecting novices suggests that cookies should be blocked until people are adequately informed of the risk they pose to information security. However, blocking cookies from the outset would drastically impair the Web experience for most novices. From an efficiency standpoint, it is important to determine the important role cookies play and ask why they are ubiquitous; in other words, do they make using the Web more efficient for users? Once again, conflicting principles provide little guidance for setting the default.

In the third example of wireless security, if the principle is protecting novices, then the default should be set to encryption. However, from the efficiency standpoint the issue is more complicated because most users don't use encryption. But, it is likely that most experienced and knowledgeable users would use encryption. Until we know why people do not choose encryption, either from informed or uninformed decisionmaking, we cannot determine which default would be more efficient. The lack of specificity for what is efficient leads to problems in setting this default based on HCI principles of efficiency.

From a policy perspective, both existing rationales (consider novice users and consider efficiency) for setting defaults are far too vague. First, what is a novice user? Is it their knowledge, experience, education, or ability to use a computer? It is not clear what defines a novice user. Moreover, why should we protect novice users? Second, efficiency is an ambiguous concept. Is the default setting most efficient for the software developers, expert users, or novices? Or is it the setting that provides the most utility? Efficiency also assumes that it is possible to determine and calculate the costs and benefits of a default setting. However, many default settings impact fuzzy values, such as privacy, or externalities, such as security, which are difficult to calculate. While these rationales are undoubtedly useful to developers, they provide an insufficient basis for setting defaults from a policy perspective.

The difference in rationales can be explained by the differences in the goals being pursued by developers and policymakers. Computer scientists typically focus on the performance of software. To this end, they break down software into small pieces and optimize each piece, keeping their goals technically oriented rather than focusing on larger, complicated social values. From a policy perspective, however, the goal is not only ensuring that the software works, but also ensuring that it comports with our societal norms.

B. *Behavioral Economics*

Behavioral economists have analyzed how defaults should be set, largely in the context of law and social policy.[84] For example, Madrian's research on a 401(k) plan discussed earlier is one of several studies that have shown the power of defaults on decisionmaking in everyday life.[85] Default settings are interesting to behavioral economists, because they appear to conflict with a key theorem in behavioral economics. The Coase theorem holds that a default rule does not matter if there are no transaction costs.[86] The default rule does not matter because the parties will bargain to a common result that is efficient for both parties. However, there are numerous empirical studies showing a bias toward deferring to defaults, a bias which is counter to what the Coase theorem would suggest, leading behavioral economists to explore what is missing from the Coase theorem. In this subpart, we discuss three explanations from behavioral economists for why people defer to defaults: bounded rationality, cognitive biases, and the legitimating effect. We then apply them to several examples of software defaults to examine their usefulness.

The first explanation involves the concept of bounded rationality. People do not change defaults when they are uninformed that another choice exists. If a person does not know about the possibility of changing an option or the ramifications of each choice, then a default setting is equivalent to a fixed setting. An example of this is how people defer to defaults for cookies, because they are either uninformed or misinformed about the cookies function. The Pew study in 2000

[84] *See generally* Ian Ayres & Robert Gertner, *Filling the Gaps in Incomplete Contracts: An Economic Theory of Default Rules*, 99 YALE L.J. 87 (1989) (discussing defaults in contract law); Cass R. Sunstein, *Switching the Default Rule*, 77 N.Y.U. L. REV. 106 (2002) (discussing defaults in the context of employment law).

[85] Madrian & Shea, *supra* note 29, at 1158–61.

[86] R.H. Coase, *The Problem of Social Cost*, 3 J.L. & ECON. 1, 15 (1960).

found that 84% of Internet users were concerned with privacy, but 56% did not know about cookies.[87] Several years later, people are still uninformed about cookies. A 2005 survey found that 42% of respondents agreed with patently false statements such as, "Internet cookies make my computer susceptible to viruses" and "Internet cookies make my computer unsafe for personal information."[88] Another 30% admitted that they know nothing about Internet cookies. Hence, users defer to the default setting that enables cookies.[89] We cannot expect users to change default settings for issues about which they are uninformed.

A second explanation from behavioral economists is that cognitive biases may impede people from changing defaults. These cognitive biases include the status quo bias, the omission bias, and the endowment effect. The status quo bias leads people to favor the status quo over a change. Samuelson and Zeckhauser describe the status quo bias as favoring inertia over action or as having an anchoring effect.[90] To explain, individuals place greater value on the current state and, thus, believe they will lose more if they make a change. The status quo bias is further explained by the omission bias. The emphasis here is not on the current state, but on the fact that people often judge actions to be worse than omissions.[91] The omission bias suggests that individuals prefer to be hurt because some action was not taken rather than equally hurt because some action was taken. In the realm of software, the omission bias suggests people will avoid changing a setting because they fear it might "break" the computer more than they fear "breaking" the computer by not taking any action.

The status quo and omission biases provide reasonable explanations for why people defer to defaults. To illustrate the differences between these explanations, consider a security setting for a firewall in a computer operating system. When a firewall is turned on, it provides the user with increased protection. Either bias could come into play in determining whether a user turns on the firewall when the

[87] *See supra* notes 56–57 and accompanying text.

[88] Press Release, BURST Media, BURST Media Reports Consumer View of Cookies: "Don't Understand Them, Can Be Good, but, Should Be Deleted" (June 2, 2005), http://www.burstmedia.com/release/pressreleases/pr_06_02_05.htm (presenting the results of a survey on the knowledge and perception of Internet cookies).

[89] *Id.*

[90] William Samuelson & Richard Zeckhauser, *Status Quo Bias in Decision Making*, 1 J. RISK & UNCERTAINTY 7, 8–10, 37–38 (1988) (examining the role of status quo effect with several experiments).

[91] Ilana Ritov & Jonathan Baron, *Status-Quo and Omission Biases*, 5 J. RISK & UNCERTAINTY 49, 50 (1992).

default is set for the firewall to be off. For example, a user knows that the firewall will protect her computer from certain hackers but may be nervous about enabling the firewall, because she is afraid it may "break" the computer. The status-quo bias suggests that the current state (a working computer) is a safe state and that leaving that state could result in a loss. Furthermore, the user is choosing to accept a possible harm due to omission versus a possible harm due to commission (turning on the firewall could lead the computer to malfunction). As such, the omission bias comes into play.

Another cognitive bias is known as the endowment effect. The endowment effect refers to how people place more value on settings when the default initially favors them than when the default is set to favor another party.[92] Empirical research has shown the endowment effect to occur when people demand much more money to give up something than they would be willing to pay to acquire it.[93] The endowment effect suggests that the initial default setting affects how defaults are valued by users. These valuations may make it very difficult for a later switch from one default setting to another one. This effect means that policymakers need to carefully consider the initial default setting.

The third explanation that behavioral economists have recognized to explain default preference is the legitimating effect.[94] This effect arises because people believe defaults convey information on how people should act. Defaults are assumed to be reasonable, ordinary, and sensible practices. As a result, people can be resistant to changing a default setting. This assumption about defaults is not surprising. For example, because of product liability law, manufacturers have a duty to warn of dangerous products[95] and a duty to "design out" dangers in a product.[96] Consequently, when people use software,

[92] *See, e.g.*, Daniel Kahneman et al., *Anomalies: The Endowment Effect, Loss Aversion, and Status Quo Bias*, 5 J. ECON. PERSP. 193, 194–97 (1991) (providing a good background on the endowment effect).

[93] Russell Korobkin, *The Endowment Effect and Legal Analysis*, 97 NW. U. L. REV. 1227, 1228, 1232–35 (2003) (reviewing empirical evidence of the endowment effect and showing how the effect broadly affects the law).

[94] Sunstein, *supra* note 84, at 116 ("[T]he significant effect from the default rule is probably a product of its informational signal. . . . [T]he initially proposed plan carries a certain legitimacy, perhaps because it seems to have resulted from some conscious thought about what makes the most sense for the most people.").

[95] *See* M. Stuart Madden, *The Duty to Warn in Products Liability: Contours and Criticism*, 11 J. PROD. LIAB. 103, 104 (1988) (discussing the duty to warn by manufacturers).

[96] *See* RESTATEMENT (THIRD) OF TORTS § 2 cmt. d (1998) (noting that manufacturers have a duty to design out dangers on a reasonable basis).

they assume that defaults are reasonable and sensible; otherwise, another choice would have been selected.

The approach of behavioral economists has focused on reasons why people comport with defaults. This is a different approach from the one within HCI, which focused on how we should set defaults. Applying the behavioral economists' insights, we gain a better understanding of why people defer to defaults. However, behavioral economists do not provide a simple answer for how best to set defaults. They realize there are different standards for judging defaults, such as efficiency, distribution, and welfare.[97] Instead, as we point out in the prescriptive subpart, their most important contribution is explaining how information flow between developers and users leads users to defer to defaults, thereby increasing the power of defaults.

Let us test the behavioral economists' explanations with our three examples of desktop icons, cookies, and wireless security. In the first example regarding the choice of default desktop icons, the endowment effect and legitimating effect can explain the companies' conflict over setting the default icons. According to the endowment effect, as the initial default setting favored Microsoft's browser, users are going to demand much more to give up the default Microsoft icon than they would be willing to pay to set it if the default did not favor Microsoft. The legitimating effect would lead people to favor one browser over another. If there is only one icon on the desktop, people are going to assume that it is the sensible or reasonable browser to use. This is recognized by the browser companies and explains why they care so much about the initial default icons.

In the second example involving enabling or disabling cookies, behavioral economists would point out the issue of bounded rationality in determining user choices. As discussed earlier, since people do not know about cookies, they cannot be expected to change the default settings.[98] Moreover, as the default is set to accept cookies, the legitimating effect explains why people would accept cookies rather than not, because, according to this effect, people trust or defer to the pre-determined selection. In the third example involving encryption for wireless security, all three cognitive biases come into play. Most people do not understand wireless security, and cognitive biases such as the omission bias and the status quo bias suggest that people will be reluctant to change the default to avoid change or potentially damaging their computers through their actions. Furthermore, because the access points come with no encryption enabled, people are likely to

[97] Sunstein, *supra* note 84, at 123–28.

[98] *See supra* text accompanying notes 56–59.

assume that this is a reasonable setting, and there is no reason to change the default setting, thus demonstrating the legitimating bias. These last two examples involving cookies and encryption show how defaults affect our actions and influence our preferences and norms. After all, the initial settings here will likely lead people to believe that cookies are desirable and that no encryption is desirable. It is in this way that defaults can subtly, but profoundly, affect the production and transmission of culture.

C. Legal Scholarship

Having discussed the explanations provided by computer scientists and behavioral economists to account for default values, we now turn to legal scholarship. Legal scholars have long been interested in defaults, because default settings are found throughout the law in contracts,[99] labor and employment law,[100] and inheritance law.[101] Contract law scholars have focused especially on the role of defaults. This subpart considers two key issues concerning defaults as understood from the perspective of contract law. The first issue concerns what are the default laws, as opposed to mandatory laws, that people cannot waive. The second issue focuses on the role of consent when people enter into contracts and how courts enforce these contracts. After covering these two issues, we apply their insights to our examples of software defaults involved in desktop icons, cookies, and wireless security.

Contract law scholars rely on a concept of default rules, which is similar to the concept of defaults in software. For example, consider Barnett's discussion about the default rule approach in the context of contract law and how he employs the analogy of software defaults:

> The default rule approach analogizes the way that contract law fills gaps in the expressed consent of contracting parties to the way that word-processing programs set our margins for us in the absence of our expressly setting them for ourselves. A word-processing program that required us to set every variable needed to write a page of text would be more trouble than it was worth. Instead, all word-processing programs provide default settings for such variables as margins, type fonts, and line spacing and leave it to the user

[99] Alan Schwartz & Robert E. Scott, *Contract Theory and the Limits of Contract Law*, 113 YALE L.J. 541, 594–609 (2003) (discussing the role of defaults in contract law).

[100] *See generally* Sunstein, *supra* note 84 (discussing the default rule in the context of employment law).

[101] *See generally* Adam J. Hirsch, *Default Rules in Inheritance Law: A Problem in Search of Its Context*, 73 FORDHAM L. REV. 1031, 1078–94 (2004) (discussing the default rule in the context of inheritance law).

to change any of these default settings to better suit his or her purposes.[102]

For Barnett, the default rule approach refers to how certain obligations and responsibilities are placed on the parties in the absence of manifested assent to the contrary.[103] If a party wishes to change a rule, he or she must specify so in the contract. This approach in contract law is analogous to how software defaults place certain obligations or limitations on the users, unless the users change the defaults.

Legal scholars have also recognized that there are some rules that parties cannot change by contract. These are known as immutable rules.[104] For example, the warranty of merchantability is a default rule that parties can waive, while the duty to act in good faith cannot be waived.[105] The difference between default rules and immutable rules is shown in an example by Ware:

> [T]he tort law giving me the right not to be punched in the nose is a default rule because I can make an enforceable contract to enter a boxing match. . . . In contrast, the law giving a consumer the right to buy safe goods is mandatory because it applies no matter what the contract terms say.[106]

The concept of immutable rules by legal scholars is analogous to how rules may be wired into software. The commonality here is that consumers or users cannot change or modify these immutable or wired-in rules.

An area of considerable controversy regarding immutable rules is intellectual property law. Radin has shown how contractual agreements and technology are creating new legal regimes, which overshadow the existing legal infrastructure of the state.[107] An example is whether "fair use" is an immutable rule or a default rule that parties can bargain away. Another related concern over immutable rules is the use of arbitration agreements. Ware argues that because arbitrators may not apply law correctly and courts are reluctant to change the results of arbitration, arbitration allows parties to sidestep mandatory

[102] Randy E. Barnett, *The Sound of Silence: Default Rules and Contractual Consent*, 78 VA. L. REV. 821, 824 (1992).

[103] *Id.* at 825.

[104] Ayres & Gertner, *supra* note 84, at 87.

[105] *Id.*

[106] Stephen J. Ware, *Default Rules from Mandatory Rules: Privatizing Law Through Arbitration*, 83 MINN. L. REV. 703, 710 (1999).

[107] Margaret Jane Radin, *Regulation by Contract, Regulation by Machine*, 160 J. INSTITUTIONAL & THEORETICAL ECON. 142, 142–51 (2004) (discussing immutable rules in contracts).

rules.[108] In effect, by using arbitration, it is possible to turn a mandatory rule into a default rule. This ambiguity between what defines default rules and mandatory rules in the law leads Radin to urge scholars and policymakers to firmly establish society's mandatory rules.[109]

A second issue of concern for contract scholars is the consensual model of contract. Much of contract law is based on the assumption that consumers have consented to default terms through a bargaining process and a meeting of the minds. However, the reality is that most consumer contracts do not function like this.[110] This has led contract scholars to examine a number of different forms of contracts and identify their flaws. Their research is relevant to defaults, because the types of agreements they study are closer in form to the default settings that consumers "consent" to in software.

Adhesion contracts are standard form contracts that are presented on a "take-it-or-leave-it" basis.[111] In this situation the consumer may be subject to terms in the contract over which he or she has little control. The modern approach has been for courts to refuse enforcement of adhesion contracts.[112] The celebrated case of *Williams v. Walker-Thomas Furniture*[113] concerned the enforcement of a standard form contract. Judge Wright wrote that courts have the power to refuse enforcement of contracts found to be unconscionable.[114] His opinion also points out the key issues for determining whether a contract is unconscionable, because it is an adhesion contract:

> Ordinarily, one who signs an agreement without full knowledge of its terms might be held to assume the risk that he has entered a one-sided bargain. But when a party of little bargaining power, and hence little real choice, signs a commercially unreasonable contract with little or no knowledge of its terms, it is hardly likely that his consent, or even an objective manifestation of his consent, was ever

[108] Ware, *supra* note 106, at 711.

[109] Radin, *supra* note 107, at 142–43.

[110] *See* John E. Murray, *The Standardized Agreement Phenomena in the Restatement (Second) of Contracts*, 67 CORNELL L. REV. 735, 739–41 (1982) (discussing the issues with integrating standard form contract law into the Restatement (Second) of Contracts); W. David Slawson, *Standard Form Contracts and Democratic Control of Lawmaking Power*, 84 HARV. L. REV. 529, 539–44 (1971) (developing legal principles for standard form contracts).

[111] *See* Todd D. Rakoff, *Contracts of Adhesion: An Essay in Reconstruction*, 96 HARV. L. REV. 1173, 1177 (1983) (providing a good definition of standard form contracts).

[112] *See id.* at 1195–96.

[113] 350 F.2d 445 (D.C. Cir. 1965).

[114] *Id.* at 449–50.

> given to all the terms. In such a case the usual rule that the terms of the agreement are not to be questioned should be abandoned and the court should consider whether the terms of the contract are so unfair that enforcement should be withheld.[115]

The issue of adhesion contracts is directly applicable to software. There are agreements that users routinely enter into when they open a box of software or click on an End User License Agreement from software they have downloaded. These agreements are known as shrink-wrap or click-wrap agreements.[116] In these transactions, there is no negotiation on the terms between the parties; consumers are presented with software on a "take-it-or-leave-it" basis. The situation is analogous to what Judge Wright discussed in *Williams*.[117] The parties have little bargaining power, and it is an open question whether they have truly consented to the terms. For example, many everyday contracts (and some licenses for software) contain pre-dispute arbitration clauses. Consumers do not bargain for these clauses, but these terms are put forth in standard form contracts on a "take-it-or-leave-it" basis.[118] This has led to debate over whether consumers should be subject to all the terms. Some scholars argue that the terms should be unenforceable, because consumers have not assented to them.[119] However, Judge Easterbrook in an influential decision held that shrinkwrap agreements are enforceable in certain circumstances.[120]

Contract scholars have argued that the solution to adhesion contracts is that the courts "should consider whether the terms of the agreement are so unfair that enforcement should be withheld."[121] This means courts can choose either to refuse to enforce a contract or

[115] *Id.*

[116] Kevin W. Grierson, Annotation, *Enforceability of "Clickwrap" or "Shrinkwrap" Agreements Common in Computer Software, Hardware, and Internet Transactions*, 106 A.L.R. 5TH 309, 317 nn.1–2 (2003).

[117] 350 F.2d at 449.

[118] Richard M. Alderman, *Pre-Dispute Mandatory Arbitration in Consumer Contracts: A Call for Reform*, 38 HOUS. L. REV. 1237, 1246–49 (2001); *see also* Margaret Moses, *Privatized "Justice,"* 36 LOY. U. CHI. L.J. 535, 536–38 (2005) (focusing on how the Supreme Court has influenced the use of arbitration provisions); David S. Schwartz, *Enforcing Small Print to Protect Big Business: Employee and Consumer Rights Claims in an Age of Compelled Arbitration*, 1997 WIS. L. REV. 33, 81 (analyzing the Supreme Court's broad endorsement of mandatory pre-dispute arbitration agreements).

[119] *See, e.g.*, Batya Goodman, Note, *Honey, I Shrink-Wrapped the Consumer: The Shrink-Wrap Agreement as an Adhesion Contract*, 21 CARDOZO L. REV. 319, 354–59 (1999) (arguing that adhesion contract principles should apply to shrinkwrap agreements); *see also* Mark A. Lemley, *Shrinkwraps in Cyberspace*, 35 JURIMETRICS J. 311, 317–18 (1995) (predicting many of the legal issues with shrinkwrap agreements).

[120] ProCD, Inc. v. Zeidenberg, 86 F.3d 1447, 1449 (7th Cir. 1996).

[121] Rakoff, *supra* note 111, at 1192 (quoting *Williams*, 350 F.2d at 450).

to rewrite the terms of the contract. However, when we consider defaults in software, enforcement is automatic and nonreviewable.[122] There is little in common between how contracts are enforced and how software is enforced. This reflects a serious distinction between law and software and will be discussed later in a discussion on how policymakers should set defaults.[123]

Now we will apply the work of legal scholars from above to our three software default examples. In the first example involving default desktop icons, the issue is what party (Compaq or Microsoft) should set the default terms. At first glance it might appear that Compaq has significant bargaining power because of its size and expertise compared to other computer hardware producers. However, Compaq was reliant on a monopoly software producer, and there is justifiable concern over whether there was a true bargaining process. As we have seen, Microsoft's behavior later led to government antitrust investigations into whether Microsoft was behaving unfairly.[124] Nonetheless, in this case, Compaq backed down in order to satisfy Microsoft's demand to restore its Internet Explorer browser icon as a default desktop setting.[125] Compaq's only remedy would have been a judicial remedy, which was uncertain, costly, time consuming, and would have hindered its relationship with a crucial supplier. This points to a crucial problem with default settings in software—there is no enforcement process for users who take issue with software settings. It is not readily apparent what a party can do if he or she is subject to "unfair" default terms. While one can refuse to use the software, this option is often an unreasonable course of action because of the lack of comparable substitutes.

While the first example of desktop icons focuses on defaults and producers, the second example (cookies) and third example (wireless security) are all situations where consumers accepted default settings without truly consenting. It could be argued that most consumers would not have consented to these settings if they were apprised of the privacy and security risks. Nevertheless, they had to take these default settings on a "take-it-or-leave-it" basis. This raises several questions: the first is whether this is analogous to a classic adhesion contract. The key difference here is that consumers are free to change the default settings. In an adhesion contract, consumers cannot

[122] James Grimmelmann, *Regulation by Software*, 114 YALE L.J. 1719, 1738–41 (2005) (noting how software rules cannot be ignored).

[123] *See infra* Part III.

[124] United States v. Microsoft Corp., 84 F. Supp. 2d 9, 60 (D.D.C. 1999).

[125] *Id.*

change the terms. Second, the main remedy against adhesion contracts is not applicable to software defaults. Consumers cannot look to the courts to require manufacturers to change a setting because the consumers did not properly consent. While courts hold contract terms unenforceable, they would be justifiably hesitant to require changes to default settings that consumers could readily modify themselves.

Legal scholarship provides useful insights into the legitimacy of software defaults. We rely on these insights to discuss how to set default settings. After all, policymakers need to understand what defaults are acceptable and what settings cannot be default settings. While research on adhesion contracts does not transfer to software, it does provide a useful template for understanding whether people consented to a transaction in other contexts. In a later section on how policymakers should set defaults, we point out that this contractual notion of consent is a useful step in evaluating whether users were informed or not about default settings.

D. Health Communication

Communication scholars studying risky behavior prefer yet another approach for addressing software defaults than those used by computer scientists, behavioral economists, or legal scholars. Although LaRose works within health communications, he is trying to transfer insights from his field to the field of software.[126] He argues that online policy issues are "too much of a moving target to ever be assured by technical means alone."[127] LaRose instead advocates educating consumers to protect themselves.[128] His work is rooted in heath communications, which focuses on changing individuals' risky behavior.[129] Using health communication research as his basis, LaRose suggests an approach for improving online security by increasing

[126] Robert LaRose et al., Understanding Online Safety Behavior: A Multivariate Model (May 31, 2005), http://www.msu.edu/~isafety/papers/ICApanelmult21.htm.

[127] *Id.*

[128] *Id.*

[129] He builds upon protection motivation theory (PMT), which had its origins in "health communication messages targeting risky behavior." *Id.* However, LaRose notes that this model has been applied to a variety of risk management issues, from crime control to exercise participation and environment protection. PMT suggests that protective behavior "is motivated by perceptions of the threat, efficacy, and consequences associated with taking protective measures and maintaining maladaptive behavior." *Id.*

self-efficacy through means such as "verbal persuasion, anxiety reduction, and progressive mastery."[130]

While we recognize a role for education and training in addressing software specifications, we believe LaRose overstates its usefulness. Software often hides subtle but important settings from its users. We simply cannot expect people to devote their resources and capacity to become the ubergeeks that modern software requires. For example, we cannot expect the uninitiated users who rely on Web browsers and wireless technologies to investigate all the possible risks of these everyday technologies. Instead, these users "satisfice" (to use Mackay's term)[131] and, therefore, defer to the settings that are given to them. While policymakers should support educating users, it is also necessary to recognize the elephant in the room—the difficulty of mastering software. Until software comports with our established norms and is easy to use, people are not going to be capable of addressing fundamental online policy concerns alone.

E. *The Missing Piece of Technical Ability*

One understudied reason why people do not change defaults is their lack of technical sophistication. In these cases, people know they ought to change the default, but cannot figure out how to do so. A crucial factor affecting their technical inadequacy is the usability of software. Usability is a broad field that cuts across computer science, psychology, and design. Two examples that highlight this problem are security and pop-up advertising.

People are very concerned about security. As the introduction noted,[132] software sales show security software is one of the most popular items purchased.[133] However, these same well-informed and motivated individuals, who bought security software, have computer systems with significant security problems. Indeed, 81% of home computers lack core security protections, such as recently updated antivirus software, properly configured firewall and/or spyware protection.[134] The best explanation for this discrepancy is that people lack the technical sophistication to properly configure their computers.

Another similar example that illustrates how a lack of technological sophistication affects people's propensity to rely on defaults is the inability of people to avoid pop-up ads. Surveys show that 77% of

[130] *Id.*

[131] Mackay, *supra* note 77, at 159.

[132] *See supra* notes 5–6 and accompanying text.

[133] *See* NPD, Top-Selling, *supra* note 6.

[134] ONLINE SAFETY STUDY, *supra* note 7, at 2.

Americans find that telemarketing calls are "always annoying"[135] and 78% of Americans consider pop-up ads "very annoying."[136] In response to these annoyances, it is estimated that over 60% of households have signed up for the FTC's Do Not Call Registry.[137] In contrast, only about 25% of people have installed blocking software for pop-up ads.[138] This discrepancy between people's proactive approach to deterring telephone marketing and their acceptance of Internet marketing pop-ups is best explained by the technical difficulty of finding, installing, and configuring pop-up ad blockers as compared with signing up for the FTC's Do Not Call Registry, which requires people to complete a simple form or call a toll-free number.[139]

These two examples illustrate that deference to software defaults is explained by a number of factors besides those discussed in the fields of computer science, economics, law and communications, one of which is usability. It is not enough for people to understand they that need to change a default; they also need to understand how to change it. This requires some technical capacity on their part as well as a usable software interface.

III. Setting Defaults

Knowing how powerfully default settings can affect people's behavior and choices leads to questions about how best to set defaults. This Part focuses on how policymakers ought to set defaults. The very notion that policymakers should be engaged with influencing the design of software has been criticized. Tien begins his article by noting the very different genealogy of law and architectural regulation in the

[135] David Krane, *National Do Not Call Registry Popular, but Public Perception of Impact on Calls Unrealistic*, Harris Interactive, Sept. 4, 2003, http://www.harrisinteractive.com/harris_poll/index.asp?PID=400.

[136] Margaret Kane, *Pop-ups, the Ads We Love to Hate*, CNET News.com, Jan. 14, 2003, http://news.com.com/2100-1023-980563.html (noting results from a GartnerG2 survey).

[137] FCC, Trends in Telephone Service 1–2 (2003), *available at* http://www.fcc.gov/Bureaus/Common_Carrier/Reports/FCC-State_Link/IAD/trend803.pdf (noting that 104 million households had telephone service as of November 2002); FTC, Annual Report to Congress for FY 2003 and 2004 Pursuant to the Do Not Call Implementation Act on Implementation of the Do Not Call Registry 1 (2005), *available at* http://www.ftc.gov/reports/donotcall/051004dncfy0304.pdf [hereinafter Do Not Call Registry] (noting that more than sixty-four million phone numbers were entered into the registry).

[138] Stefanie Olsen, *New IE May Burst Pop-up Bubble*, CNET News.com, Nov. 24, 2003, http://news.com.com/2100-1024_3-5110805.html (discussing estimates of the use of pop-up blocking software).

[139] Do Not Call Registry, *supra* note 137, at 2–3.

context of software.[140] This leads him to argue that software operates surreptitiously as compared with law, which is based around public deliberation and an enforcement process.[141] He is concerned that the surreptitious nature of software leads people to unquestioningly view software features as part of the background and not as something that is intended to control us.[142] An example of this surreptitious nature is with software filtering, which may lead us to "forget" about certain types of content.[143] This leads Tien to express extreme reluctance about relying on software as a method of regulation.[144]

We recognize Tien's concerns, but his concerns are much weaker in the case of defaults. Policymakers are typically not creating default settings, but instead are trying to tune existing default settings to maximize social welfare. In some cases, if policymakers do not intervene and switch a default setting then people will be worse off. Also the process of policy intervention into defaults will undoubtedly highlight the role of software and its malleability. This should dispel many of the concerns that Tien has raised.

This next subpart begins by considering the threshold question of whether there should even be a default setting in software for particular functions. The argument here is largely based upon the work of legal scholars, who have analyzed whether a law should be immutable or a default. The second subpart then focuses on how defaults should be set. In providing guidance, we rely on key insights from several disciplines on understanding how defaults operate. As a starting point, we rely on behavioral economists' analysis of defaults with the understanding that behavioral economists have explored how defaults should be set for a variety of public policy issues. However, in discussing how defaults should be set, we also rely on the observations of computer scientists on the role of user customization and the goal of efficiency. Finally, legal analysis of the role of consent, as well as our emphasis on a user's technical sophistication, is also integrated into our recommendations. We also include flowcharts as well as a subpart applying these flowcharts to the cookies and wireless security example.

[140] Tien, *supra* note 72, at 4–12.

[141] *Id.* at 11–12.

[142] *Id.* at 14.

[143] *See supra* text accompanying notes 71–72.

[144] Tien, *supra* note 72, at 14.

A. *Default or Wired-In*

A threshold issue when setting software defaults is whether there should be a default setting or a wired-in setting. A wired-in setting is in effect a mandatory rule. As a starting point, consider the earlier analysis by legal scholars on the conflicts between default rules and mandatory rules in law.[145] Within law, there are a set of rights that are clearly nonwaivable, for example, in the areas of legal enforcement or redress of grievances, human rights, and politically weak or vulnerable rights.[146] Practical examples are safety regulations and the right to family leave.[147] The question then becomes, are there similar limitations on wired-in settings, and how can policymakers identify these settings? We explore this issue by first considering public policy limitations on wired-in settings and then move on to a pragmatic evaluation for identifying wired-in settings.

Software is malleable and can be manipulated in such a way as to limit traditional legal regimes. The classic example is the use of Digital Rights Management software, which may limit the ability of a user to copy content or even destroys content after a certain period of time.[148] The twist is that instead of using terms in a contract, a developer can incorporate the terms into the software. This ability to use a wired-in setting or a technological protection measure (TPM)[149] is a way of substituting contract terms with technology, thereby forcing the user to adhere to the developers' preferences. Other examples of how developers use TPMs to replace contract terms could affect distribution of the software or its content (e.g., limiting the number of computers it can operate on) or replacing restrictions on personal versus commercial use with numerical limits (e.g., limiting consumer version of photo editing software to 1000 photos). In these cases, technology settings are replacing contract terms.

[145] *See supra* text accompanying notes 104–09 (discussing immutable rules in the law).

[146] Margaret Jane Radin, Machine Rule: The Latest Challenge to Law 22 (Jan. 31, 2005) (on file with author), *available at* http://www.aals.org/2005midyear/contracts/RadinmaterialsMachineRule.pdf (arguing generally that the law should be extended or interpreted to make rights in these categories harder to waive or non-waivable).

[147] Occupational Safety and Health Act of 1970, 29 U.S.C. §§ 651–678 & 42 U.S.C. § 3142-1 (2000) (stating the nonwaivable right to certain safety regulations); Family and Medical Leave Act of 1993, Pub. L. No. 103-3, 107 Stat. 6 (codified in scattered sections of 5 and 29 U.S.C.) (stating the nonwaivable right to family leave).

[148] Ariel Katz, *The Potential Demise of Another Natural Monopoly: New Technologies and the Administration of Performing Rights*, 2 J. COMPETITION, L. & ECON. 245, 248–51 (2006) (discussing digital rights management in the context of music downloads).

[149] Radin, *supra* note 146, at 2.

The issue then becomes are there any limitations to wired-in settings? Radin suggests that we think of wired-in settings as technological self-help. She writes, "Using TPM's is more like landowners building high fences and less like using trespass law. Indeed, using TPM's is more like landowners building fences beyond their official property lines, and deploying automatic spring guns to defend the captured territory."[150] As Radin's example illustrates, while self-help plays a role in determining how producers develop their technology, the state places limitations on technological self-help. Without these limitations, too much self-help would lead to a Hobbesian "war of all against all."[151] Consequently, as a starting point policymakers need to identify in stark terms the mandatory or immutable rules that society requires for wired-in settings and default settings.[152] If developers know what can and cannot be a default term, they will likely respect this guidance and develop their software accordingly. This would prevent conflicts between public policy and software.

When developers rely on wired-in settings, Radin offers two recommendations on their usage. First, it is necessary to give users notice and information about how the wired-in setting operates.[153] Second, there should be a judicial remedy for wired-in settings.[154] Radin suggests that users be allowed to challenge the setting and seek a judicial declaration invalidating it.[155] This would provide a way for users to challenge a wired-in setting on the grounds of public policy.

Once policymakers have decided a potential wired-in setting is legitimate, the next question is whether it is practical.[156] Sunstein provides us with four factors policymakers should consider when choosing between a default setting and a wired-in setting. The first is whether users have informed preferences.[157] If they know little about the setting, they are not likely to change it, and vice versa. It makes sense to include a wired-in setting over a default setting when people know little about the setting. The second issue is whether the mapping of defaults in software to user preferences is transparent.[158] In the case of software, this requires an easy-to-use interface that allows

[150] *Id.* at 27.

[151] *Id.* at 29.

[152] Radin, *supra* note 107, at 144.

[153] Radin, *supra* note 146, at 33.

[154] *Id.*

[155] *Id.*

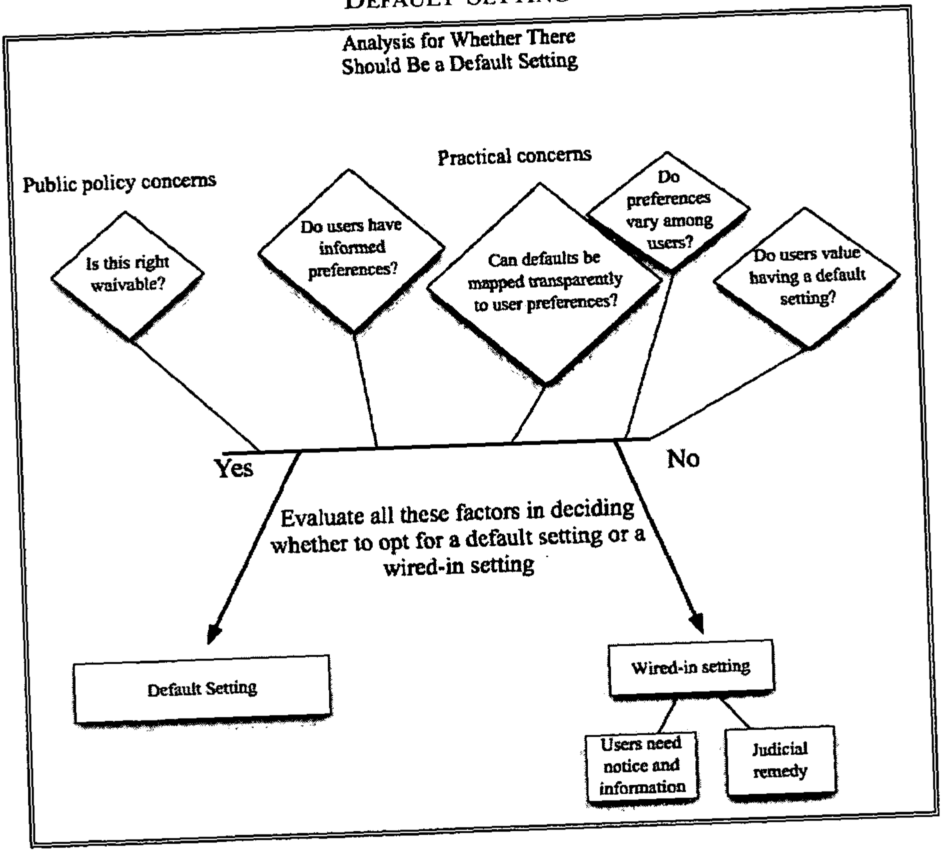
[156] *See* Burk, *supra* note 16, at 19–20 (examining the costs of DRM technologies that are wired-in).

[157] Sunstein & Thaler, *supra* note 28, at 1197.

[158] *Id.* at 1198.

users to configure the software according to their preferences. The third issue focuses on how much preferences vary across individuals.[159] If there is little or no variation in society, it hardly makes sense to create a default setting as opposed to a wired-in setting. The final issue is whether users value having a default setting.[160] This can be determined by examining marketing materials, software reviews, and comments from users. If there is little concern over the default setting, it becomes reasonable for designers to opt for a wired-in setting. A summary of these issues can be found in Figure 2, which maps out the issues that policymakers should consider.

FIGURE 2. ANALYSIS FOR WHETHER THERE SHOULD BE A DEFAULT SETTING


Analysis for Whether There Should Be a Default Setting
Practical concerns
Public policy concerns
Is this right waivable?
Do users have informed preferences?
Can defaults be mapped transparently to user preferences?
Do preferences vary among users?
Do users value having a default setting?
Yes
No
Evaluate all these factors in deciding whether to opt for a default setting or a wired-in setting
Default Setting
Wired-in setting
Users need notice and information
Judicial remedy

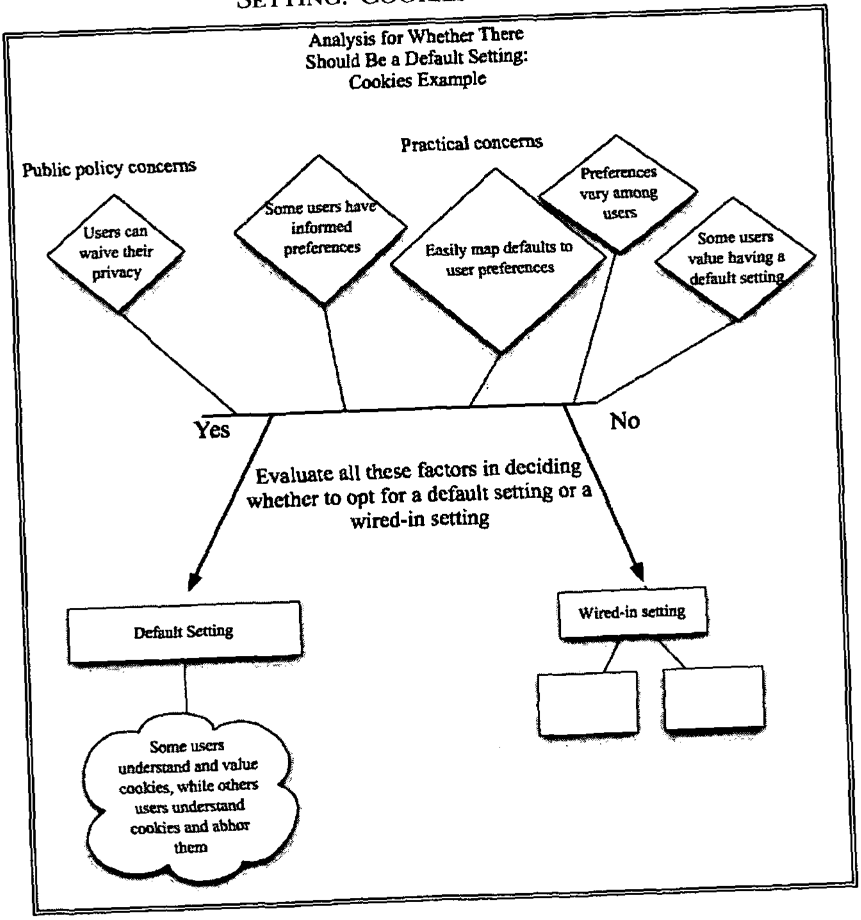
As an example, we can apply this to the cookies example. Should cookies be wired in beyond the control of users or should we allow

[159] *Id.*

[160] *Id.* at 1198–99.

users the option of a default setting? As a starting point, the first issue is whether this is a waivable right. Clearly, if people want to reduce their privacy, it is their right. The next issue then considers a whole host of practical concerns. First, it is clear that some users have informed preferences. There is a small but significant number of users that advocate choice and control when it comes to privacy settings. Second, the control over the cookies setting can easily be mapped to user preferences. All that is needed is a simple checkbox that allows users to decide whether they want to permit cookies. Third, there is a variation of preferences among users, with some users seeking to protect their privacy at all costs and other users willing to exchange their privacy readily for conveniences. Finally, there are a considerable number of users that value having a choice, even though they may choose not to exercise it. These issues are summarized in Figure 3.

FIGURE 3. ANALYSIS FOR WHETHER THERE SHOULD BE A DEFAULT SETTING: COOKIES EXAMPLE


Analysis for Whether There Should Be a Default Setting: Cookies Example
Practical concerns
Public policy concerns
Users can waive their privacy
Some users have informed preferences
Easily map defaults to user preferences
Preferences vary among users
Some users value having a default setting
Yes
No
Evaluate all these factors in deciding whether to opt for a default setting or a wired-in setting
Default Setting
Wired-in setting
Some users understand and value cookies, while others users understand cookies and abhor them

B. A Framework for Setting Defaults

This subpart focuses on how policymakers should set default settings. The first section provides a general rule for setting defaults. The next three sections are exceptions to this general rule. The final section provides a number of methods for adjusting the power of a default setting.

1. Defaults as the "Would Have Wanted Standard"

Behavioral economists have analyzed how defaults should be set.[161] Much of this analysis has focused on defaults associated with law and social policy, specifically contracts, but this reasoning can be extended to software. As we discussed earlier, behavioral economists' starting point is the Coase theorem, which holds that a default rule does not matter if there are no transaction costs.[162] This is because the parties will bargain to a common result that is efficient. According to this analysis, regulators do not need to be concerned with defaults in software, assuming there are no transaction costs. Yet there are always transaction costs in setting defaults. The general approach of legal scholars in contract law is that defaults should be set to minimize transaction costs. Posner argues that default rules should "economize on transaction costs by supplying standard contract terms that the parties would otherwise have to adopt by express agreement."[163] The idea here is that the default settings should be what the parties would have bargained for if the costs of negotiating were sufficiently low. This approach is known as the "would have wanted" standard and is the general approach for setting defaults in contract law.[164]

The "would have wanted" standard is a good starting point for setting defaults in software. Let the parties decide what they want software to accomplish, and then let the developers decide what options to build into software. In following this approach, developers would likely follow the common sense principles of HCI in protecting novices and enhancing efficiency.[165] The underlying assumption in assessing the default is that both parties are negotiating over the default.

The "would have wanted" standard does not mean that there are no limitations for setting defaults. To the contrary, as we point out in

[161] *See generally* Ayres & Gertner, *supra* note 84 (discussing default rules in the context of economic efficiency).

[162] Coase, *supra* note 86, at 15.

[163] RICHARD A. POSNER, ECONOMIC ANALYSIS OF LAW 413 (6th ed. 2003)

[164] Ayres & Gertner, *supra* note 84, at 89–90 (1989).

[165] *See* s*upra* notes 78–83 and accompanying text.

the next few sections there are several situations where the "would have wanted" standard is not the best basis for setting defaults. In these cases, policymakers may need to intervene. Besides this intervention, policymakers need to be proactive. As behavioral economists have shown, the initial default setting has a disproportionate effect on users because of the status quo bias, omission bias, the endowment effect, and the legitimating effect.[166] This means that policymakers need to ensure that the initial default settings are correct. If they are not, it will be a much more difficult job for policymakers to later switch the default setting to another one.

The next three sections focus on limitations to the "would have wanted" standard. Before discussing them, we need to note a necessary requirement for government intervention in software settings. A default setting should only be actionable if it materially affects a fundamental societal concern. While it is not in society's interest for government to select the default font for a word processor, it is in society's interest to make sure fundamental societal values are protected. To illustrate this, consider the examples we have used throughout this Article involving desktop icons, cookies, and wireless security. All three of these examples affect fundamental societal concerns of competition, privacy, and security, respectively.

2. Problem of Information

There are situations when you would expect certain users to change the default. If they are not changing it, then it is necessary to examine their deference. For example, if defaults relating to accessibility are not widely changed among users, this should not raise a red flag, unless disabled users are not changing these default settings. If the disabled are not changing them, then there could be an informational problem that is leading them to defer to the default setting. At this point, policymakers must evaluate whether there is a problem of information.

In considering whether parties are fully informed, policymakers need to examine several factors. These factors were identified in our earlier discussion of understanding defaults and include bounded rationality,[167] cognitive biases,[168] the legitimating effect,[169] and technical sophistication.[170] All of these factors should be used by

[166] *See supra* Part II.B.

[167] *See supra* text accompanying notes 87–89.

[168] *See supra* text accompanying notes 90–93.

[169] *See supra* text accompanying notes 94–96.

[170] *See supra* text accompanying notes 134–39.

policymakers to assess whether users are fully informed. After all, factors such as the omission bias or endowment effect may influence people to defer to default settings. An analytical starting point for determining whether users are informed is the work of legal scholars. Their analysis of consent in contracts should be useful to policymakers in determining whether users are truly informed about defaults.[171] As an example, consider Judge Wright's analysis of consent in a standard form contract.[172]

If users are not fully informed and capable of changing the default settings, then the default should be what the parties "would have NOT wanted." The idea here is that this setting will force the developers to communicate and share information in order to have users change the setting to what they "would have wanted." In contract law, this is known as a penalty default and is used to encourage disclosure between the parties.[173] A classic example of a penalty default is that courts assume a default value of zero for the quantity of a contract.[174] The value of zero is clearly not what the parties would have wanted, because they were bargaining for an exchange of goods. However, this penalty default serves to penalize the parties if they do not explicitly change the default.

Penalty defaults are best used in situations where parties are not equally informed.[175] In the case of software, this can mean users who are uninformed, misinformed, or lacking technical sophistication. In everyday practice, this suggests that socially significant defaults should be set to protect the less-informed party. This setting forces software developers to inform and communicate with users when they want users to perform advanced actions that may have adverse consequences on their computers if not set properly. In addition, it encourages developers to ensure that defaults can be changed with a minimal degree of technical sophistication. As an example, some manufacturers of wireless points already use penalty defaults. Most (but not all) wireless access points are disabled by default. Users must go through a setup process or to a configuration menu to enable the access point. While this default setting is not what a consumer would have wanted, this penalty setting allows manufacturers to help the user properly configure the access point through a setup process.

[171] *See supra* text accompanying notes 110–22.

[172] *See supra* text accompanying notes 113–15.

[173] Ayres & Gertner, *supra* note 84, at 95–107 (discussing the use of penalty defaults).

[174] *Id.* at 95–96.

[175] *Id.* at 98–100.

Another example where a penalty default is appropriate is the setting for cookies in Web browsers. As we pointed our earlier, cookies are not well understood by most people. A penalty default would require the default be set to reject cookies. If Web browsers and Websites want people to use cookies, then they would have to explain to users what cookies are and how to turn them on. By changing this default, policymakers can use the information-forcing function of penalty defaults to improve the state of online privacy. We believe that if Web browsers were forced to do this, they would quickly develop an interface that would inform users about cookies and highlight the benefits of using them. This would ensure that people understood the privacy risks of cookies. Penalty defaults are not appropriate in all circumstances, such as for settings that people readily understand. For example, if most people understand the concept of filters and are capable of using software-filtering technology, then a penalty default is unwarranted. In this case, policymakers should follow the "would have wanted" standard for setting defaults.

3. Externalities

A second reason for settings defaults at what the parties "would have *not* wanted" is to account for externalities. Settings in software can often affect third parties in a myriad of ways that are analogous to increasing the risk to an innocent passerby or through pollution. In these situations, policymakers should consider the overall welfare of users and intervene to ensure a default value is set to reduce externalities. However, if the problem is grave enough, it may be necessary to change the setting from a default value to a wired-in setting. In effect, this recommendation echoes HCI guidance by setting the default to what is most efficient for society.[176]

An example of where software defaults create high externalities is wireless security. Most manufacturers would prefer not to enable all wireless security functions, mainly because it leads to reduced functionality and increased support costs. Most users know very little about wireless security issues and cannot adequately bargain for their inclusion. This inaction costs everyone when wireless security is compromised. These costs could be reduced if security features, such as encryption, were enabled by default.

The core finding for wireless security can be applied to security in software. Default settings for all software should be generally set to enable security. Unfortunately, developers are still selling products

[176] *See supra* text accompanying notes 78–83.

that have defaults set to insecure values. The most egregious examples are internet-enabled products that rely on default passwords, such as the Axis camera used at Livingstone Middle School as discussed in the introduction.[177] Policymakers should force these developers to change their default password function to improve security and societal welfare.

4. Compliance with the Law

There are occasional circumstances when policymakers need to set defaults to comply with laws, regulations, or established legal principles. While these circumstances often involve issues with externalities or lack of information for users, they do not necessarily have these issues. They may be protecting values we hold as immutable.[178] For example, government may mandate default settings under the guise of paternalism. The Children's Online Privacy Protection Act sets a default rule that websites cannot collect information from children.[179] Websites can switch from this default setting only if they have obtained parental consent.[180] This example illustrates how policymakers may need to defer to existing laws in setting defaults.

The first example of software defaults we discussed involved Microsoft and Compaq sparring over the default icons on the desktop. How should a policymaker set the default in this situation? This question is a difficult one that the courts considered during Microsoft's antitrust trial. The district court and court of appeals held that Microsoft's restrictions on default icons were anticompetitive because they raised the cost for manufacturers to add additional software and therefore protected Microsoft's monopoly.[181] At this point forward, policymakers now have guidance for how these software defaults should be set. It is more difficult to argue retrospectively that policymakers in 1995 should have intervened and set these defaults. Nevertheless, this example shows how policymakers may need to set defaults to comport with existing law and policy.

[177] *See supra* notes 3–4 and accompanying text.

[178] *See supra* text accompanying notes 104–09.

[179] Children's Online Privacy Protection Act, 15 U.S.C. §§ 6501–6506 (2000). For general background, see EPIC's Children's Online Privacy Protection Act (COPPA) website at http://www.epic.org/privacy/kids/ (last visited Oct. 6, 2006).

[180] 15 U.S.C. §§ 6502(b) (2000).

[181] United States v. Microsoft Corp., 87 F. Supp. 2d 30, 39 (D.D.C. 2000), *aff'd in part, rev'd in part*, 253 F.3d 34, 64–67 (D.C. Cir. 2001); *see* David McGowan, *Between Logic and Experience: Error Costs and* United States v. Microsoft Corp., 20 BERKELEY TECH. L.J. 1185, 1231–36 (2005) (reviewing the issue of default icons in the Microsoft antitrust trial).

5. Applications of the Framework

This section illustrates the decision-making process for switching an existing default setting. A flowchart for the process can be found in Figure 4. To illustrate this process, we provide flowcharts for two examples: wireless security and cookies.

FIGURE 4. DECISION PROCESS FOR SWITCHING AN EXISTING DEFAULT SETTING

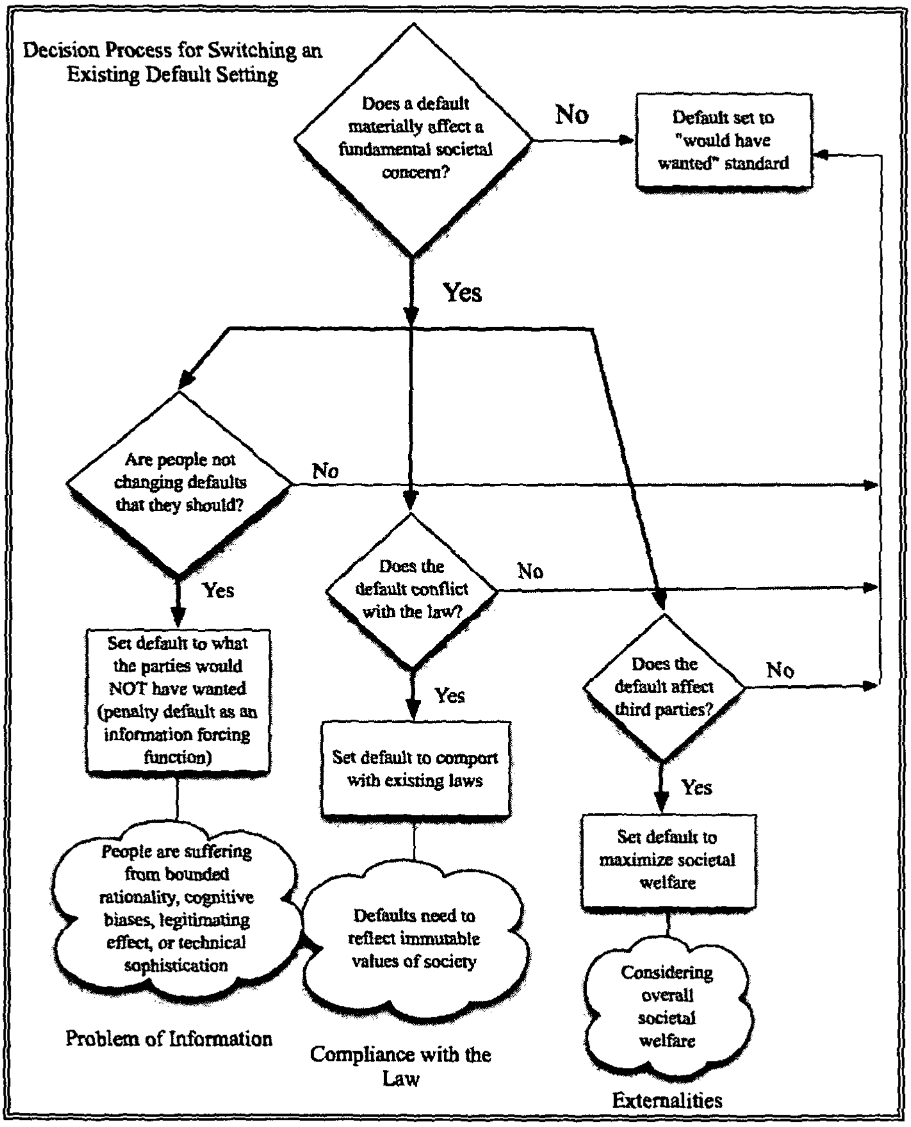
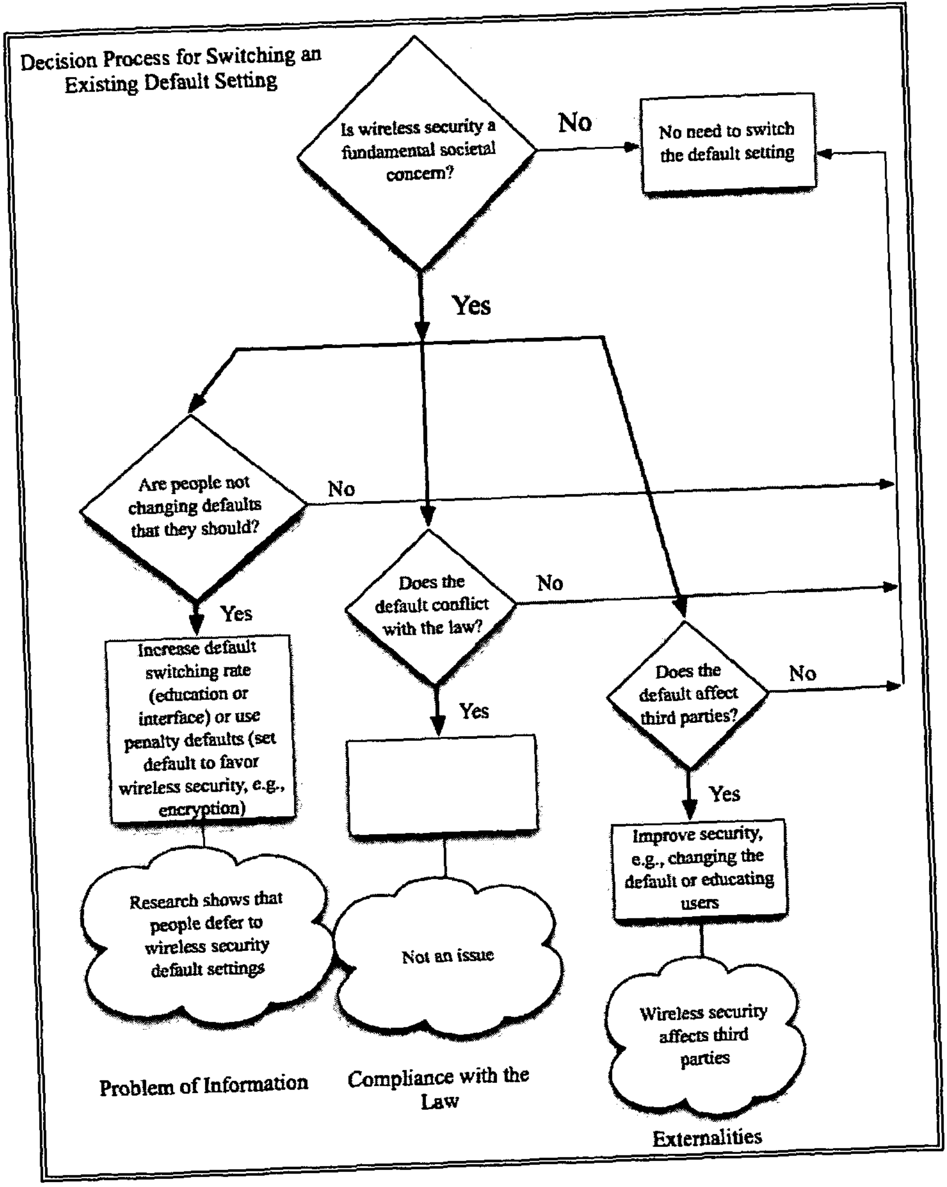

Decision Process for Switching an Existing Default Setting
Does a default materially affect a fundamental societal concern?
No
Default set to "would have wanted" standard
Yes
Are people not changing defaults that they should?
No
Yes
Set default to what the parties would NOT have wanted (penalty default as an information forcing function)
People are suffering from bounded rationality, cognitive biases, legitimating effect, or technical sophistication
Problem of Information
Does the default conflict with the law?
No
Yes
Set default to comport with existing laws
Defaults need to reflect immutable values of society
Compliance with the Law
Does the default affect third parties?
No
Yes
Set default to maximize societal welfare
Considering overall societal welfare
Externalities

FIGURE 5. WIRELESS SECURITY EXAMPLE


Decision Process for Switching an Existing Default Setting
Is wireless security a fundamental societal concern?
No
No need to switch the default setting
Yes
Are people not changing defaults that they should?
No
Yes
Increase default switching rate (education or interface) or use penalty defaults (set default to favor wireless security, e.g., encryption)
Research shows that people defer to wireless security default settings
Problem of Information
Does the default conflict with the law?
No
Yes
Not an issue
Compliance with the Law
Does the default affect third parties?
No
Yes
Improve security, e.g., changing the default or educating users
Wireless security affects third parties
Externalities

In Figure 5, the decision process is applied to the wireless security example. The first decision for wireless security is whether it affects a fundamental societal concern. Security is universally considered a significant issue. The next step is to consider whether any of the three criteria apply for intervening and switching a default setting. The first is whether people are not changing defaults that they should. Research shows that people are deferring to existing default settings without truly consenting. This suggests that policymakers should intervene and switch the defaults settings, in effect utilizing penalty defaults. As we discussed above, penalty defaults provide an information-forcing function that can improve the state of security.[182] Another alternative is to increase default switching through education or an improved user interface. The second criterion does not apply here, because there is no issue regarding compliance with the law. The third issue of externalities is important. As we discussed in the section on externalities, wireless security imposes costs on third parties.[183] Consequently, policymakers need to either change the default setting or ensure that more people will switch the default, e.g., by educating users.

In Figure 6, the decision process is applied to the cookies example. The first issue is the threshold issue and asks whether cookies affect a fundamental societal concern. The issue of online privacy is fundamental and is manifested in debates over cookies in the policy community. The first criterion for switching an existing default setting is whether people are not changing defaults that they should. As we discussed earlier in the problem of information section, cookies are not well understood by people.[184] Consequently, one way of remedying this is by using penalty defaults that would entail switching the default to "off" for cookies. The second criterion is compliance with the law. This issue depends upon the user. Federal agencies have restrictions on using cookies.[185] While there have been lawsuits against the use of cookies, courts have not found cookies illegal or deceptive.[186] For now, this criterion does not push for changing default settings for cookies. Finally, the last criterion does not apply, because the collection of cookies data does not affect third parties.

[182] *See supra* Part III.B.2.

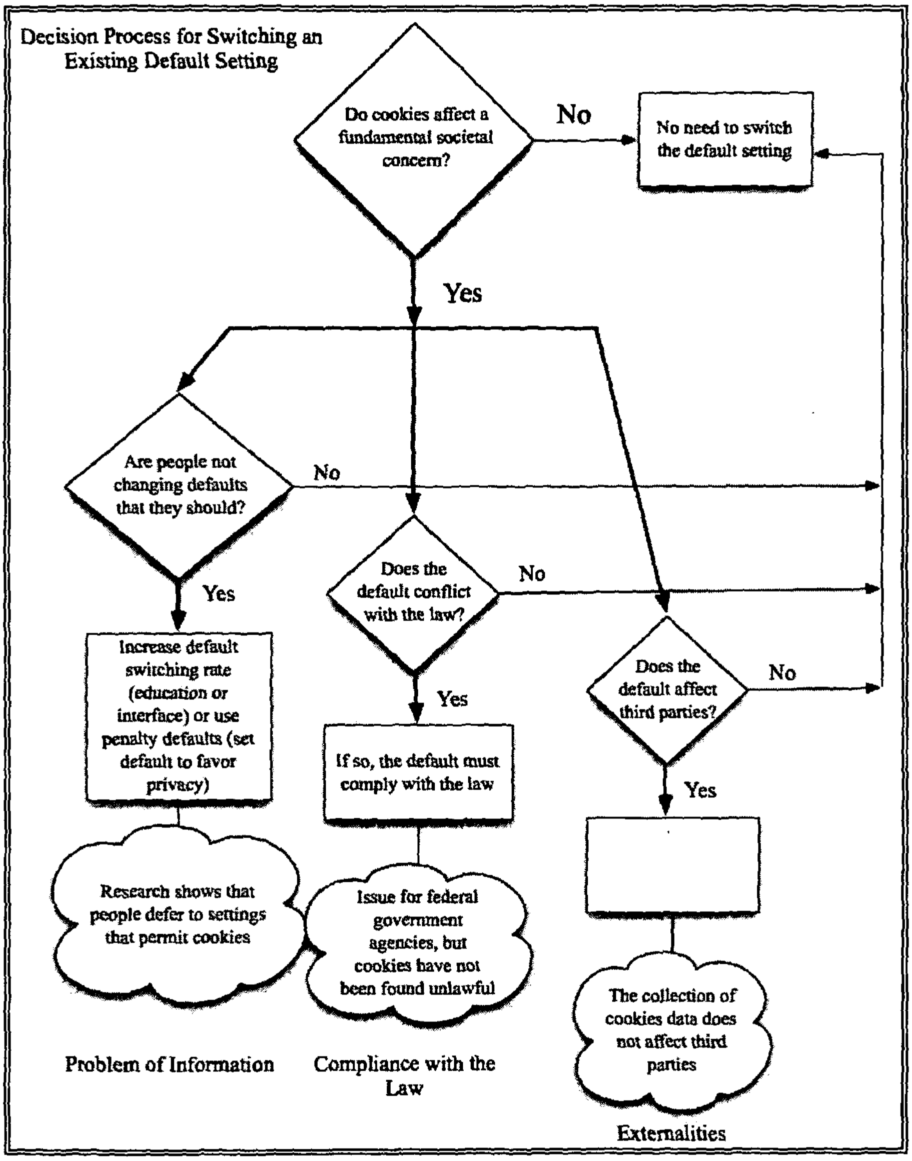
[183] *See supra* Part III.B.3.

[184] *See supra* Part III.B.2.

[185] Memorandum from Jacob J. Lew, Director, Office of Mgmt. and Budget, to Heads of Executive Departments and Agencies, Privacy Policies and Data Collection on Federal Web Sites 2 (June 22, 2000), *available at* http://www.whitehouse.gov/OMB/memoranda/m00-13.html.

[186] Dan Richman, *Online Privacy Gets Safeguard*, SEATTLE POST-INTELLIGENCER, Aug. 27, 2002, at E1.

FIGURE 6. COOKIES EXAMPLE


Decision Process for Switching an Existing Default Setting
Do cookies affect a fundamental societal concern?
No
No need to switch the default setting
Yes
Are people not changing defaults that they should?
No
Yes
Increase default switching rate (education or interface) or use penalty defaults (set default to favor privacy)
Research shows that people defer to settings that permit cookies
Problem of Information
Does the default conflict with the law?
No
Yes
If so, the default must comply with the law
Issue for federal government agencies, but cookies have not been found unlawful
Compliance with the Law
Does the default affect third parties?
No
Yes
The collection of cookies data does not affect third parties
Externalities

6. Adjusting the Power of a Default

In general, more default settings are better, because they allow users to reconfigure and use their software as they see fit. However, there are limitations to this rule that are recognized within HCI's user customization research.[187] First, the more defaults that are present,

[187] *See supra* text accompanying notes 75–77.

the more likely users will be confused and intimidated by the number of choices. Second, there are practical limits to how many default settings designers can present in a useful manner without overloading the user interface. As designers add more functions that users can modify, the software will reach a point of diminishing returns where users are overwhelmed and confused. In effect, this places a practical limit on how many default options should be available to users.

The power of a default setting can be modified in two ways. The first is through changes in the user interface. For example, increasing (or reducing) the prominence of a default setting in the user interface can affect its use. Second, procedural constraints can make it more costly to change a default setting. These procedural constraints could ensure users are acting voluntarily and are fully informed before they change a default setting. A simple example is an extra prompt that asks users whether they are really sure they want to change the default setting. A more extensive example is changing the settings for an air bag. To install an air bag on-off switch, the consumer must send a request form to NHTSA and then bring the NHTSA authorization letter to a dealership to have a switch installed.[188] These procedural constraints attempt to address the problem of bounded rationality and bounded self-control. While a wide range of possible procedural constraints exist, they all serve to raise the cost of switching the default setting.

If modifications to the user interface and procedural constraints are not enough, then the situation may require a wired-in setting versus a default setting.[189] There are a variety of reasons, including safety and various externalities (e.g., radio interference, network congestion, or security), why users should not be able to change a setting. In these situations, a policymaker may seek a wired-in setting; however, this is a serious decision, because it limits the user's control.

IV. Shaping Defaults Through Government Intervention

Unlike in contract law, there appears to be very little role for the judicial system or government in enforcing defaults. This does not mean that the judicial system or government is powerless over defaults. Instead, there are a number of actions government can take to influence default settings in software. In general, there are two approaches for government intervention into defaults settings. This Part begins by discussing government forcing developers to offer a default setting versus government mandating a certain default setting. The

[188] Make Inoperative Exemptions, 49 C.F.R. § 595.5(b)(1) (2005).

[189] *See supra* Part III.A.

rest of this Part focuses on methods the government can use to affect default settings, such as regulation.

The first method government could use is mandating developers to incorporate certain features into software. These features could be wired-in or default settings, but the emphasis here is changing the software to include these features. A simple example in automobile manufacturing is how the government mandated seat belts in automobiles.[190] The government is not focused on the default setting for seat belt use; instead it just wants to ensure that occupants have a choice.

The second method available to the government is for it to favor a certain default setting. In some cases, the government has to pick a default option, because there has to be a choice if an individual does not make any decision. A good example here is government's policy on organ donation. Government has to choose a default position, either a person is presumed to have consented to organ donation or a person must explicitly consent to donation.[191] In other cases, the government chooses a default value to advance societal welfare. For example, the warranty of merchantability is a default rule that parties can waive.[192]

A. *Technology Forcing Regulation*

The typical approach for government to promote social welfare is to rely on technology forcing regulation to ensure certain features are incorporated into communication technologies.[193] For example, the government mandated closed-captioning technology into televisions to aid people who are deaf.[194] Similarly, the government mandated the incorporation of the V-chip to assist parents in blocking inappropriate television content.[195] In both these examples, the govern-

[190] *See* ROBERT W. CRANDALL ET AL., REGULATING THE AUTOMOBILE 155–56 (1986) (discussing government-mandated safety improvement for automobiles).

[191] Johnson & Goldstein, *supra* note 35, at 1338.

[192] Ayres & Gertner, *supra* note 84, at 87.

[193] *See* Jay P. Kesan & Rajiv C. Shah, *Shaping Code*, 18 HARV. J.L. & TECH. 319, 363–70 (2005) (providing an overview of technology forcing regulation for software).

[194] The incorporation of closed captioning technology was similar to the incorporation of the ultrahigh frequency (UHF) tuner. Before government regulation, consumers were forced to buy an expensive stand-alone decoder. *See generally* Sy DuBow, *The Television Decoder Circuitry Act—TV for All*, 64 TEMP. L. REV. 609, 610–11, 615–16 (1991) (providing a history of legislative process to require manufacturers to incorporate closed captioning).

[195] The V-chip was a relatively simple technology based on the modification of the closed captioning technology. *See* Kristen S. Burns, Legislative Update, *Protecting the Child: The V-Chip Provisions of the Telecommunications Act of 1996*, 7 DEPAUL-LCA J. ART

ment's goal is to ensure users have an option. They are not requiring manufacturers to set a certain default setting.

In other instances, technology forcing regulation can also require certain default settings. The anti-spam legislation known as CAN-SPAM had a default setting of opt-out for commercial electronic mail messages.[196] A sender has to provide a mechanism in each message to allow recipients to refuse additional messages. This policy is different from the one adopted by the European Union, which requires an opt-in process. In the European Union a recipient must have given prior consent before they can be sent an email message.[197] Similarly, the United States government's National Do Not Call Registry provides people with a choice to receive telemarketing calls.[198] The default is that people will accept telemarketing calls. If they do not wish to receive these calls, they need to register their phone number with the registry.[199]

Another example of technology forcing regulation affecting default settings is the Children's Internet Protection Act (CIPA).[200] The Supreme Court decision on CIPA focused on the disabling of filters for adult access.[201] The ability to disable the filters was an important element to ensure the law was not overly restrictive. The general consensus by librarians is that to comply with the law, they need to set up computers where the filter is on by default, but adult patrons can disable the filter.[202]

& ENT. L. & POL'Y 143, 145–46 (1996); Lisa D. Cornacchia, Note, *The V-Chip: A Little Thing but a Big Deal*, 25 SETON HALL LEGIS. J. 385, 391–92 (2001).

[196] Controlling the Assault of Non-Solicited Pornography and Marketing Act of 2003 (CAN-SPAM Act) § 2, 15 U.S.C. §§ 7701–7713 (Supp. III 2003).

[197] *See* FRANÇOISE BECKER, CAN-SPAM AND THE EU DIRECTIVE 1 (2003), *available at* http://www.lsoft.com/news/optin2003/canspamvseu.pdf (providing an overview of the differences between the U.S. and European approach towards unsolicited "junk" email).

[198] "Do-Not Call" Provisions of Telemarketing Sales Rule, 64 Fed. Reg. 66,124, 66,124 to 66,126 (Nov. 24, 1999) (codified at 16 C.F.R. §310.4 (2006)) (announcement of public forum).

[199] *Id.*

[200] Children's Internet Protection Act (CIPA), Pub. L. No. 106-554, §§ 1701–1741, 114 Stat. 2763, 2763A-335 to 2763A-352 (2000) (codified at 20 U.S.C. § 9134 (2000 & Supp. III 2003); 47 U.S.C. § 254 (2000 & Supp. III 2003)).

[201] United States v. Am. Library Ass'n, 539 U.S. 194, 203–09 (2003) (noting the ability of library patrons to have software filtering disabled).

[202] *See* Robert Bocher & Mary Minow, *CIPA: Key Issues for Decision Makers*, WEBJUNCTION, Aug. 31, 2003, http://webjunction.org/do/DisplayContent?id=990 (offering a summary of issues associated with CIPA, including filter disabling, the use of computers by patrons, and definitions of key legal terms); Thomas M. Susman, *Questions and Answers on Filter Disabling Under CIPA*, AM. LIBRARY ASS'N, Dec. 3, 2003, http://

B. Other Means for Shaping Software

The government has several means at its disposal to influence default settings besides regulation. The first is a market-based approach, which uses market incentives as either a stick or a carrot.[203] In the stick approach, the government relies on its tax policy to penalize certain software settings.[204] An exemplar of how the government uses tax policy to penalize certain sales is the gas-guzzler tax, which penalizes the sale of inefficient automobiles.[205] A similar policy could be used to penalize software that does not meet a certain criterion, such as basic security or accessibility features. This would encourage developers to develop software differently. The problem with this approach is enforcement. Many software programs are not sold, such as open source software, or are bought from other countries. A better approach may be for the government to rely on tax expenditures.

Tax expenditures operate by reducing a firm's tax burden to create an incentive for developing certain software.[206] For example, government could give a tax break to software developers whose software is highly secure or incorporates accessibility features. Enforcement is much easier in this case, because firms have an incentive to prove to the government that they are complying with the requirements of the tax expenditure. This carrot approach is likely to be much more successful at pushing developers to include certain features or defaults in software.

A second approach the government can use to influence default settings is the implementation of information-forcing measures. This strategy could include requiring software developers to disclose information about their products to the public.[207] Software developers could be forced to disclose certain security or privacy features to consumers. This would increase consumer awareness that there are cer-

www.ala.org/ala/washoff/WOissues/civilliberties/cipaweb/adviceresources/scenarios.htm (providing examples of this policy of filtering by default).

[203] *See* Kesan & Shah, *supra* note 193, at 342–51 (discussing market-based approaches for shaping software).

[204] *Id.* at 343–46. *See generally* Eric M. Zolt, *Deterrence Via Taxation: A Critical Analysis of Tax Penalty Provisions,* 37 UCLA L. REV. 343 (1989) (discussing the use of tax penalties).

[205] Gas Guzzler Tax, 26 U.S.C.A. § 4064 (West 2000 & Supp. 2006).

[206] *See* Kesan & Shah, *supra* note 193, at 380–84 (discussing the use of tax expenditures for shaping software). *See generally,* STANLEY S. SURREY & PAUL R. MCDANIEL, TAX EXPENDITURES (1985) (providing the authoritative work on tax expenditures).

[207] *See* Kesan & Shah, *supra* note 193, at 361–63 (discussing the role of disclosure for shaping software); *see also* STEPHEN BREYER, REGULATION AND ITS REFORM 161–64 (1982) (discussing disclosure as a means of regulation).

tain settings incorporated into the software. An example of disclosure requirements is within the Children's Online Privacy Protection Act,[208] which sets a default rule that Websites cannot collect information from children.[209] Websites can switch from this default setting, only if they have obtained parental consent. Instead of forcing disclosure, the government could spend its resources educating people about settings in software. For example, the FCC set up a consumer education initiative for digital television,[210] and the SEC has launched educational campaigns to warn investors of scam Websites.[211]

A third approach relies on government's procurement power to favor software with preferred default settings.[212] For example, government has set procurement requirements favoring energy efficient computers.[213] The same set of requirements could be set for software in areas such as security, privacy, or accessibility. Similarly, the government could favor certain default rules by ensuring the government purchases technology with those default rules. This method strives to stimulate demand for a certain set of technologies.[214] The government could create a market for technologies that are secure by default. For example, it would only purchase technology that does not use default passwords.

[208] Children's Online Privacy Protection Act, 15 U.S.C. §§ 6501–6506 (2000).

[209] *Id.* § 6502(b). Similarly, the FCC has a rule that "prohibit[s] interactivity during children's programming that connects viewers to commercial matter unless parents 'opt in' to such services." Children's Television Obligations of Digital Television Broadcasters, 19 F.C.C.R. 22,945, 22,968 (2004).

[210] *See Digital Television (DTV) Tomorrow's TV Today!*, http://www.dtv.gov/ (last visited Oct. 6, 2006) (providing the public with information about digital television).

[211] Press Release, SEC, Regulators Launch Fake Scam Websites To Warn Investors About Fraud (Jan. 30, 2002), http://www.sec.gov/news/headlines/scamsites.htm.

[212] *See* Kesan & Shah, *supra* note 193, at 371–79 (discussing procurement as an effective method by government to influence software). *See generally* C. Edquist & L. Hommen, *Public Technology Procurement and Innovation Theory*, *in* PUBLIC TECHNOLOGY PROCUREMENT AND INNOVATION 5, 20–27 (Charles Edquist et al. eds., 2000) (discussing the opportunity that governments have to pursue policy goals by making specific demands when procuring new technology).

[213] Exec. Order No. 11,912, 41 Fed. Reg. 15,825, 15,825 to 15,826 (Apr. 13, 1976) (calling for several measures to improve energy efficiency of equipment government purchases).

[214] *See generally* Jennifer McCadney, *The Green Society? Leveraging the Government's Buying Powers to Create Markets for Recycled Products*, 29 PUB. CONT. L.J. 135 (1999) (discussing the government's procurement power in the context of environmental concerns).

CONCLUSION

Defaults in software are powerful, because for a variety of reasons, people defer to them. This has implications for specific societal issues, such as wireless security, but it may also affect our social norms and culture. After all, the notion of open and free Wi-Fi is in part attributable to the default value of no encryption. Consequently, defaults are important not only for policymakers, but also for those seeking to understand the impact of technology upon culture.

This Article provides several examples of how defaults can influence behavior. Defaults are powerful not only because so many people rely on them rather than choose an alternative, but also because there is little understanding of software defaults. We considered how the disciplines of computer science, behavioral economics, legal scholarship, and communications theorize defaults. While we found limitations in all these disciplinary approaches, we also found useful insights for understanding why people defer to software defaults. To illustrate these insights, we applied all four approaches to several concrete examples dealing with issues of competition, privacy, and security.

This led us to provide recommendations for how defaults should be set. As a threshold matter, we set forth a methodology for deciding whether we should use a default setting or a wired-in setting in a particular situation. We argue, in general, that policymakers should not intervene in default settings and that developers should rely on the "would have wanted" standard.[215] This standard ensures that the wishes of both parties are met in the design of defaults. However, there are three circumstances where policymakers may need to intervene and challenge the settings agreed to by users and developers. These are all highlighted on the flowcharts for the decisionmaking process. The first circumstance typically arises when users lack the knowledge and ability to change an important default setting. In these cases, policymakers ought to use penalty defaults to shift the burden of the default to the developer. This penalty default setting serves as an information-forcing function to educate users while users are changing the default settings.

One scenario for the government to implement a penalty default is one involving privacy issues. Setting a penalty default to protect a user's information forces developers to notify and educate users before they have to share their personal information. While this approach is paternalistic, it still provides users with the freedom to

[215] *See supra* Part III.B.1.

choose as they wish. We suggest that in these rare situations when there is a fundamental societal concern at stake and people are uninformed, misinformed, or not technically sophisticated enough to change the default, then, as a matter of public policy, people should be protected. If people want to give up that protection, then we should support well-informed individuals to make that decision. However, the default should be set to protect individuals.

The second circumstance where policymakers need to intervene involves default settings that cause harm to third parties. These externalities may need to be addressed by changing a default value. A good example of this is system security. While it is in the interest of users and developers to make systems very open to other users, this can have a negative externality because of costs from network congestion and spam. In this situation, policymakers have an interest in ensuring a default is either set to reduce externalities or to insist that the default be replaced with a "wired-in" setting to limit externalities.

The final circumstance in which policymakers need to intervene is when a default setting does not comport with existing law and policy. In these situations, it is necessary for policymakers to ensure the default setting is changed. Examples of this are defaults relating to competition and antitrust. Policymakers may need to ensure that a monopolist does not use defaults in an anticompetitive fashion.

Besides these recommendations, we also noted a number of other considerations policymakers need to take into account. First, biases such as the endowment effect and the legitimating effect can make changing the initial default very costly. This means policymakers need to carefully consider the initial default setting. Second, a concerted effort needs to be undertaken to identify the defaults software can and cannot have. Arguably, there are some values that software developers cannot allow users to waive.

The final part of the Article focused on steps government can take in shaping defaults. This part was not meant as an exhaustive list of measures government can take, but as a way to show that government is not powerless in dealing with software defaults. Government has a long history of regulating software and influencing software defaults. Besides regulation, government has a variety of other approaches available. These approaches include fiscal measures, such as its power of taxation and procurement power, as well as trying to ensure that users are informed about software defaults.

This Article's normative analysis regarding software settings is unique. While many scholars have recognized the power of software, our approach is unique in terms of arguing from a generalized framework how default settings in software should be determined. We believe that as scholars further investigate and understand the impact of software on social welfare, they will conduct normative analyses for other software characteristics, such as standards, modularity, and the like. Indeed, today policymakers have little guidance for analyzing other governance characteristics of software, such as transparency and standards. Our hope is that this Article provides a step toward influencing software to enhance social welfare.