Software and Patent Scope: A Report from the Middle Innings

Robert P. Merges*

*In the 1980s and early 1990s, it was commonly said that patents would severely damage the software industry. I review some of these early predictions and hold them up to the light of actual experience. However judged—by overall industry revenues, by product innovation, or by vibrancy of new firm entry—the industry today appears quite robust. I conclude that the early predictions were wrong. This helps explain why we are experiencing what might be called the "normalization" of software patents. Now, the frontier legal issues pertaining to software no longer center on whether it should be patentable in the first place. Post-*State Street Bank*, the interesting questions now concern the details and contours of patent protection for software inventions. As with other technologies, the breadth or scope of software patents is a crucial issue. One of the several doctrines that collectively determine a patent's scope is the "written description" requirement in patent law. I briefly review the rise of this doctrine after 1995, arguing that in many cases the new doctrine is redundant: traditional principles of enablement are often a better ground for decision. One exception is the line of cases involving "misappropriation by claim amendment," but even here a modest extension of enablement principles would achieve a fair result without the cumbersome apparatus of written description. I then look in detail at the recent* LizardTech *case, which applied the written description requirement to a software patent. This serves as an interesting case study in how software firms are acquiring and using patents in their competitive strategies. The overall theme of the Article is normalization: the legal system is integrating software into the fabric of patent law, and software firms are integrating patents into the competitive fabric of the industry. Proper application of enablement principles will help insure reasonable scope for software patents and thus assist this process of normalization.*

---

* Wilson Sonsini Goodrich & Rosati Professor of Law and Technology, U.C. Berkeley (Boalt Hall) School of Law. I would like to thank my research assistants Ranganath Sudarshan and Jeffery Kuhn for truly outstanding help in preparing this Article. I also thank the staff of the Texas Law Review for their professionalism and diligence in running the conference at which I presented this Article and in the editing process.

I. Introduction

I started studying and teaching[1] patent law in 1988. At that time, software patents were just beginning to seep out from the back currents and ease into the mainstream. I will not bore the reader with a case-by-case account of this long process. Suffice it to say: with the persistence of rabbits in a field or termites in a fallen tree, software patents just kept showing up. Finally, perhaps exhausted with the business of drawing lines around software inventions in a futile attempt to keep them out of the domain of patents, the Federal Circuit threw in the towel in 1998. And, as so many predicted, *après State Street Bank*,[2] *le déluge*. Software applications, and then of course issued patents, increased dramatically. Today they are commonplace.

This brief Article surveys the post-deluge scene in the software industry. Though there are many—and I mean *many*—interesting predictions and provocative hypotheses in the vast literature on the topic, I will limit myself to two observations about how patents have affected the industry, which I will make in Part II. For good measure, or just to stir things up, I will add a conjecture about the future. Observation number one is straightforward: the software industry is thriving rather than dying. Whatever else patents have done or not done to this industry, they have not killed it. Observation two explains why: legal issues are of secondary importance. The major driving forces in the industry, technological change and widespread capital availability, simply swamp whatever marginal effects law might be exerting. This leads to the conjecture: my educated guess is that patents over business methods will follow roughly the same trajectory. With perhaps a few carve outs for particularly problematic subject matter (such as patents on tax strategies), business method patents seem destined to become a regular feature of the commercial landscape in the coming years. The same problems of line drawing we saw with software are already appearing in this area. If those problems prove as intractable—and they might well—we will have to learn to live with business methods too.

Viewed from the perspective of legal doctrine, the trends I survey in Part II add up to one simple conclusion: history has largely answered the § 101 question, viz., is software patentable subject matter, and should it be?[3] There is no doubt now that it is, and the normative question is dropping away in importance.[4] Thus, the interesting questions now concern the contours and

1. Actually, as a beloved older colleague used to remind me, I merely "met my classes regularly." Only sporadically, then and now, do I actually teach anyone anything.

2. State St. Bank & Trust Co. v. Signature Fin. Group, Inc., 149 F.3d 1368 (Fed. Cir. 1998).

3. 35 U.S.C. § 101 (2000).

4. There are two qualifications. First, this pertains to the United States, but not necessarily other countries. *See* Matthew Broesma, *Europe's Software Patent War Ignites Again*, CNET NEWS.COM, Sept. 21, 2006, http://news.com.com/2100-1012-6118063.html (describing the latest in a long round of controversies over software patents in Europe). Second, there have been a few recent rumblings signaling that perhaps the § 101 issue is not quite dead even in the United States. *See* Transcript of Oral Argument at 13, Microsoft Corp. v. AT&T Corp., 127 S. Ct. 1746 (2007)

details of the software protection doctrine. As always with patent law, many of the important questions come down to issues of *scope*: how broad will software patents be? How many potential software programs will be covered by a given patent? Already, the case law is moving beyond § 101 to address this fundamental question of patent scope. And of the doctrines that collectively determine scope, none is more *au courant* than the law of disclosure—in particular, the knotty and troublesome "written description" doctrine.

A recent Federal Circuit software patent case, *LizardTech, Inc. v. Earth Resource Mapping, Inc.*,[5] turned decisively on application of this doctrine. *LizardTech* thus provides an excellent vehicle for discussing current developments in the law of software patent scope. This is the task I take up in Part III of this Article. In that Part, I briefly survey the rapid ascent of the written description doctrine and explain why it has emerged as an adjunct—if not a complete successor—to the traditional law of enablement. The *LizardTech* opinion itself reveals both the appeal and the shortcomings of this prominent new doctrine. On the positive side, the impetus behind the written description revolution is well illustrated by the facts in *LizardTech*: the patent claim at issue appears to have been broader than what the teachings of the specification could reasonably support. Yet I also argue in Part III that the Federal Circuit could have reached the same result using the reliable tools of traditional enablement doctrine. The thrust of my comments, then, is to praise the outcome of some of the recent written description cases while advocating a more parsimonious doctrinal approach: namely, reliance on the traditional "undue experimentation" standard of enablement under § 112.[6] I admit that this standard might be improved with some timely renovations. But, as *LizardTech* and other recent cases show, the court need not have buried the traditional standard altogether under the confusing rubble of a new and untested doctrine.

In the end, though, *LizardTech* is important more for its outcome than its reasoning. As I describe in Part III, if it is followed by other cases, *LizardTech* will signal an important trend. Together with other recent developments, particularly the hobbling of the formerly robust doctrine of equivalents, it may presage a cautiously narrow approach to software patent

---

(No. 05-1056) (involving the export of software components under 35 U.S.C. § 271(f), containing this exchange: "JUSTICE SCALIA: 'You can't patent, you know, on-off, on-off code in the abstract, can you?' MR. OLSON [Petitioner's Counsel] 'That's correct, Justice Scalia.'"), *available at* http://www.supremecourtus.gov/oral_arguments/argument_transcripts/05-1056.pdf. *See generally Ex part*e Bilski, Appeal No. 2002-2257 (B.P.A.I. Sept. 26, 2006), *available at* http://www.uspto.gov/web/offices/dcom/bpai/its/fd022257.pdf (holding unpatentable a process for hedging investment risk that is a good example of a noncomputer-dependent, or "pure," business method patent). And, as a completely separate matter, there are a good number of scholars who still believe as a normative matter that software patents are not a good thing. *See, e.g.*, James Bessen & Robert M. Hunt, The Software Patent Experiment (Mar. 16, 2004) (unpublished manuscript), *available at* http://www.researchoninnovation.org/softpat.pdf.

5. 424 F.3d 1336 (Fed. Cir. 2005).

6. 35 U.S.C. § 112.

scope. If so, an important theoretical issue will be joined. Loosely speaking, there have been two major schools of thought on the topic of patent scope. The "big, wide, early" school, which often rallies under the banner of Edmund Kitch's prospect theory, favors broad initial property grants to pioneers.[7] The "narrower, pro-competition" school favors narrower patents as a first principle, with varying degrees of deviation permitted based on the nature of the technology or industry in question.[8] The consensus among commentators on software patent scope, and a fair inference from other work in the literature, is that software is a good candidate for this narrower, pro-competition treatment.[9] So if these theoreticians are correct, *LizardTech* and any progeny it might spawn would seem to be a step in the right direction.

But another branch of theory—the more recent "anticommons" theory—would point out that too many narrow patents could coalesce into a dark cloud on the software industry's horizon.[10] The reason is transaction costs. Too many individual property rights in the hands of too many individual owners can—theoretically at least—wreak havoc on the creation and distribution of workable software products, which often encompass many potentially patentable components. So Part III describes how the anticommons logic may play out in the software industry. In this Part, I put my credentials as a "transaction cost optimist" to work, describing various means and mechanisms by which at least the worst version of the anticommons is likely to be avoided. In addition, I describe judicial tools that can be deployed to prevent unfair rent seeking via rogue patent owners, in particular the newly rediscovered powers of equity after the Supreme Court's 2006 decision in *eBay Inc. v. MercExchange, L.L.C.*[11]

---

7. *See* Edmund W. Kitch, *The Nature and Function of the Patent System*, 20 J.L. & ECON. 265, 271 (1977) (analogizing patent holders to mineral claimants to show the economic benefit of the prospect function of patents).

8. *See* Mark A. Lemley, *The Economics of Improvement in Intellectual Property Law*, 75 TEXAS L. REV. 989, 1047 (1997) (detailing problems in Kitch's prospect model for both patents and copyrights).

9. *See, e.g.*, Jay P. Kesan & Andres A. Gallo, *Why "Bad" Patents Survive in the Market and How Should We Change?—The Private and Social Costs of Patents*, 55 EMORY L.J. 61, 66 n.23, 76 (2006) (noting that software patents are often "bad patents," which are patents that impose significant social costs, including stifling innovation); Robert P. Merges & Richard R. Nelson, *On the Complex Economics of Patent Scope*, 90 COLUM. L. REV. 839, 843 (1990) (questioning the claim that granting patents with broad scopes to initial inventors induces more effective development and future invention).

10. *See, e.g.*, Michael A. Hellar, *The Tragedy of the Anticommons: Property in the Transition from Marx to Markets*, 111 HARV. L. REV. 621, 624 (1998) (explaining that the tragedy of the anticommons occurs when multiple actors have ownership power and exclusion over a scarce resource, resulting in underuse of the resource); Ronald J. Mann, *Do Patents Facilitate Financing in the Software Industry?*, 83 TEXAS L. REV. 961, 999 (2005) (examining claims that an abundance of patents creates an anticommons harmful to software innovation).

11. 126 S. Ct. 1837 (2006).

In the conclusion, I wrap all this up, trying to sound wise and prophetic without being redundant. If you keep faith with me until the end of the Article, you can judge whether I succeeded.

II. The Software Industry: Yesterday and Today

I begin with some history. Fifteen or so years ago, when it was becoming apparent that patents were becoming a fact of life in the software industry, many feared for the industry's future. An MIT forum in 1989, for example, brought together a fair cross section of people involved with the software industry in the Boston area. At that forum, Dan Bricklin—cofounder of the company that created VisiCalc (an early "hit" product in the Apple II era)—spoke up about his concerns.[12] The notes of the Forum read as follows:

> Bricklin emphasized that a sophisticated applications program may involve 10–10,000 patentable processes. He noted that if companies began spending money to obtain software patents for these processes then the current royalty structure would have to change in order for companies to remain profitable. Bricklin noted that there is an important distinction to be made [between] . . . most patents [and] . . . software patents. He explained that there is usually one patent that covers the whole product in the case of plant or chemical products. In contrast, a software product can easily involve hundreds of patents for a single product.
>
> Bricklin characterized the software industry as inherently cottage-based. He explained how most of the major advances in the PC industry seem to come out of small shops or out of small development teams. Some examples include WordPerfect Corp., Lotus Corp., and Software Arts. Bricklin noted that with even better tools today one programmer can do even more than he accomplished in the past. He believes that some products should be written by individuals or small groups to achieve better cohesiveness while there is still demand for large companies to handle the larger scale projects. In some cases, Bricklin notes that it is cheaper for a company to go outside and buy a software product rather than develop [it] themselves. He believes that if the industry had the copyright protection just on the source code, it would be cheaper to buy than to make.
>
> Bricklin commented that many people feel software must be "protectable" because it is a product of someone's hard work. In his opinion, "craftsmanship" is not protectable and he does not feel that just because you work hard on something, e.g., software, that it should

12. Daniel Bricklin, President, Software Gardens, Inc., Software Patents: A Horrible Mistake at the MIT Communications Forum (Mar. 23, 1989) (private seminar notes on file with author).

> be protectable. He believes we should have patents because patents advance technology, not because patents are inherently good.
>
> Bricklin also cited the problem of "mine fields" in that a software developer often finds out about a related patent after the product has been shipped. . . . As a developer of software he is also uneasy about patents since he admits having limited knowledge about intellectual property. In reaction to the increase in software patents, Bricklin noted he has been working on lower tech products which involve using information in the public domain.[13]

Bricklin was by no means alone; many others shared the same fear. In a 1994 manifesto alarmingly entitled *Software Patents: An Industry at Risk*, the League for Programming Freedom emphasized "the special properties of software that make the application of the patent system inappropriate," including "the complexity of software" and unprecedented rapid change and development.[14] Their prediction for a future with patents was bleak: "A vision of patents entrenched in the software industry is a vision of stagnation. A vision of IBM once again calling the shots. A vision of companies like Xerox and AT&T who have proven incapable of bringing innovative products to market stealing profits from those companies [that] can."[15]

A. *Fears of the Forefathers*

It is safe to say that software professionals such as Dan Bricklin and the members of the League for Programming Freedom were mostly, but not completely, wrong about the prospects for software patents. Patents have not killed the software industry, they have not led to a slowdown in entry, and they do not appear to have assisted in the entrenchment of large companies at the expense of smaller and newer ones. Despite the predictions of the League for Programming Freedom, the industry has not stagnated. Early leaders in acquiring patents, such as the legacy hardware-oriented firms like Xerox and AT&T, have not been able to prevent entry, slow innovation, or even slow down the evolution of the industry to any measurable degree.[16] (Indeed, some of these legacy firms are struggling to survive, in part due to the dynamism of software and other sectors of the technology-intensive

---

13. *Id.*

14. Gordon Irlam & Ross Williams, Software Patents: An Industry at Risk (Jan. 25, 1994), http://lpf.ai.mit.edu/Patents/industry-at-risk.html.

15. *Id.* § 4.3.

16. *See, e.g.*, Thomas P. Burke, *Software Patent Protection: Debugging the Current System*, 69 NOTRE DAME L. REV. 1115, 1127 (1994) (noting failures of industry giants IBM, AT&T, and Xerox to prevent competition and market share erosion); Bradford L. Smith & Susan O. Mann, *Innovation and Intellectual Property Protection in the Software Industry: An Emerging Role for Patents*, 71 U. CHI. L. REV. 241, 262–63 (2004) (discussing ways patents can help smaller firms and provide innovation incentives).

industries in which they operate.[17]) In particular, patents do not seem to have influenced overall industry concentration, nor do they appear to have affected the minimum efficient scale of firms in the industry.[18] In addition, I will present data showing that successful software firms are doing more than simply cynically stockpiling patents; they are putting real effort into seeking and obtaining patents that demonstrate at least some earmarks of quality.

The early patent-era predictions forecast an industry where entry would stagnate.[19] The idea was that patents would entrench established companies, slowing the pace of change and ultimately putting a damper on firm entry. In this Part of the Article, I discuss broad trends in firm entry in the software industry since the dawn of the patent era (1988–1994). As suggested earlier, I conclude that entry continues to be robust and, therefore, that the predictions just mentioned have turned out to be wrong.

So my observation amounts to this: patents have not killed software. Now I am the first to admit that this is hardly a ringing endorsement for the new regime of software patents. After all, who would buy something from a salesperson whose only claim was that the product would not kill you? However, given the history of debate in this area, this is a nontrivial point. For there was a time and there were people (sometimes, almost, myself included) who thought patents would very seriously harm the software industry. For them (including me, or, at least, some past version of me) it is good news indeed that they (we) were wrong. By almost any measure, the software industry in the United States is doing quite well. Whether this is because software patents are really in the end *good* for the industry, or whether the industry has just learned to get by with them and maybe at times put them to useful ends, no one really knows; though I make some guesses in Part IV, the conclusion. But the simple point is that the industry has survived the onslaught of patents, at least reasonably well and at least so far.

---

17. *See* Martin Campbell-Kelly, *Not All Bad: An Historical Perspective on Software Patents*, 11 MICH. TELECOMM. & TECH. L. REV. 191, 242 (2005) (discussing patent innovation and tracing historical competition for various software fields); Anthony Bianco & Pamela Moore, *Downfall X: The Inside Story of the Management Fiasco at Xerox*, BUSINESSWEEK, Mar. 5, 2001, at 82, *available at* http://www.businessweek.com/2001/01_10/b3722001.htm (reporting Xerox's woes over poor management, increased competition, and slipping edge on innovation).

18. *See* Richard S. Gruner, *Better Living Through Software: Promoting Information Processing Advances Through Patent Incentives*, 74 ST. JOHN'S L. REV. 977, 1065–66 (2000) (arguing that software patents are a major incentive for smaller companies to innovate despite their disadvantage competing with industry leaders).

19. *See, e.g.*, Kurt M. Saunders & Linda Levine, *Better, Faster, Cheaper—Later: What Happens When Technologies Are Suppressed*, 11 MICH. TELECOMM. & TECH. L. REV. 23, 27 n.10 (2004) (investigating early claims that companies might amass patents to harm rivals and erect barriers to start-ups); Simpson L. Garfinkel, *Software Makers Row Over Patents*, CHRISTIAN SCI. MONITOR, Sept. 12, 1989, at 8, *available at* http://www.csmonitor.com/1989/0912/fsoft.html (reporting on fears of abuse of the software patenting process to stifle competition); Rory J. O'Conner, *Software Firms Fear Patents May Stifle Innovation But They Also Want Their Work Protected*, SAN JOSE MERCURY NEWS, Mar. 6, 1989, at 1A (reporting that software developers feared a rush to the patent office would harm the overall software industry).

B. General Industry Structure: Predictions and Reality

Another prediction from the 1990s was that patents would promote the concentration of the software industry over time—meaning that there would be fewer companies, most of them large. Indeed, it was not uncommon for scholars to suggest that legal protection for software—at the time, usually copyright protection—would contribute to this trend.[20] The idea was that strong ownership over software, in particular "backbone" software having network-effect implications, would tend to "lock in" a dominant product and hence the company that owned it.[21] What this view ignored was the dynamic sources of growth in the industry. The antitrust aphorism stated that in software and other network industries, there is competition *for* markets rather than competition *in* markets.[22] The view from the 1990s underestimated how thoroughly *competitive* most of the industry is. At the same time, the 1990s view also overestimated the pervasiveness of competition over standards. For every backbone product—such as an operating system program—there are many applications and ancillary products that connect to the backbone. For these products, ownership rights do not appear to create lock in conditions on anything like the scale envisioned in the early 1990s so long as software firms do not need to be vertically integrated to survive, much of the action is in highly competitive sectors with little or no potential for lock in.

Industry concentration statistics tell the story here. The conventional measure of concentration, the Herfindahl–Hirschman Index (HHI), ranges from 0 to 10,000; the HHI for the software industry as a whole is less than 244 for software, compared to an average of 334 for U.S. manufacturing industries.[23] What this means is that the top twenty sellers of packaged software generate 61% of total industry revenues.[24] This compares very favorably to other industries, many of which are considered quite competitive: autos, airlines, and personal computers, for example.[25] Over time, there is evidence of significant turnover as well—a key indicator of a dynamic

---

20. *See* David A. Rice, *Public Goods, Private Contract and Public Policy: Federal Preemption of Software License Prohibitions Against Reverse Engineering*, 53 U. PITT. L. REV. 543, 599 (1992) ("Expansive software copyright . . . erects formidable barriers to interbrand competition and the development of compatible products.").

21. *See id.* (arguing that the prevalence of software copyright leads to the persistence of supracompetitive pricing, which in turn leads to "increasing industry concentration and the aggressive use of copyrights to limit competition").

22. *See* DEP'T OF JUSTICE, ANTITRUST DIVISION SUBMISSION FOR OECD ROUNDTABLE ON PORTFOLIO EFFECTS IN CONGLOMERATE MERGERS 9–10 (2001) (noting that the idea that antitrust laws protect competition, not competitors is "[p]erhaps the single most quoted aphorism in U.S. antitrust jurisprudence").

23. David S. Evans, *Policy and Programming: Government Preferences for Promoting Open Source Software*, *in* GOVERNMENT POLICY TOWARD OPEN SOURCE SOFTWARE 34, 36 (Robert W. Hahn ed., 2002).

24. *Id.*

25. *See* U.S. CENSUS BUREAU, EC02-31SR-1, CONCENTRATION RATIOS: 2002, at 57 (2006), *available at* http://www.census.gov/prod/ec02/ec0231sr1.pdf (listing HHI measurements for many fields).

industry. Of the top ten software companies in 1990, half did not make the list in 2000, either because they went out of business or were acquired.[26] This is remarkable turnover compared to some industries, such as pharmaceuticals, where similar comparisons from 1990 and 2000 show that eight of ten firms made both lists—and the ones that did not were acquired by others that did.[27]

There is solid evidence that changes in the industry were driven by innovation. Evans cites estimates that in 1986, research and development (R&D) by publicly traded software firms was 1% of total domestic R&D; by 2000, that had grown to 10%.[28] The transition in the industry from stand-alone enterprise and desktop computing to a fully networked, Internet-based computing environment was fueled in part by this massive R&D spending. (To be sure, government-funded and volunteer open-source projects contributed as well.) Many new firms took advantage of the opportunities created by this transition.[29] Changes in the composition of industry leadership thus belie the predictions of a stagnant industry.

1. Patents and Innovation: Some Comparative Data.—How do patents relate to these trends? Consider some evidence from U.S. patents obtained by foreign-based software companies. In general, researchers find that as a foreign country moves up the learning curve in the software industry, inventors in the software industry from that country receive more patents.[30] For example, in data through 2002, inventors from Israeli software firms received far more patents than did those from Ireland and India.[31] The latter two countries have sizeable software industries; Irish software firms earned $18.0 billion in 2003, while Indian firms earned $1.5 billion in the same year.[32] The Israeli industry has revenues of $4.1 billion.[33] Nonetheless, Israeli inventors receive far more U.S. patents.

---

26. Evans, *supra* note 23, at 36.

27. *Id.*

28. *Id.*

29. *See* MARTIN CAMPBELL-KELLY, FROM AIRLINE RESERVATIONS TO SONIC THE HEDGEHOG: A HISTORY OF THE SOFTWARE INDUSTRY 308 (2003) (noting the proliferation of new software firms and regional software trade associations in the 1980s and 1990s alongside the growth of the Internet).

30. *See* Marco Giarratana et al., *The Role of Multinational Companies*, *in* FROM UNDERDOGS TO TIGERS: THE RISE AND GROWTH OF THE SOFTWARE INDUSTRY IN BRAZIL, CHINA, INDIA, IRELAND, AND ISRAEL 207, 219–20 (Ashish Arora & Alfonso Gambardella eds., 2005) [hereinafter FROM UNDERDOGS TO TIGERS] (finding a correlation between the relative technological advantage of a country and the number of patents granted to domestic assignees).

31. *See id.* at 215 ("The total number of patents granted to Israeli inventors is over three times the number of patents of Indian and Irish inventors altogether.").

32. Suma S. Athreye, *The Indian Software Industry*, *in* FROM UNDERDOGS TO TIGERS, *supra* note 30, at 7, 9; Anita Sands, *The Irish Software Industry*, *in* FROM UNDERDOGS TO TIGERS, *supra* note 30, at 41, 41.

These comparative data track qualitative assessments regarding the nature of the software industries in the three countries. Software firms in India and Ireland are less innovative than elsewhere.[34] Routine service-type programming is the norm in India,[35] while in Ireland, the industry is dominated by subsidiaries of foreign firms who add minor value to the parent company software and who are located in Ireland partly for tax reasons.[36] Israeli firms, which are perceived as being the source of more innovative software, patent much more heavily than their Indian and Irish counterparts. While a number of things might explain this pattern, it is certainly consistent with the view that patents correlate closely with R&D and innovation—which would tend to refute the early 1990s argument that patents are anathema to software innovation. In addition, the Israeli software industry is in no sense highly concentrated.[37] So the comparative data once again support the notion that the predicted concentration-increasing effects of software patents have failed to materialize.

C. Focus on Entry

Different industries reflect different sources of innovation. In some, a few large, established firms contribute significant innovations.[38] In others, much innovation comes from small start-ups.[39] The software industry shows some elements of both patterns: established firms and new start-ups have both been major sources of new products and other innovations.[40] Therefore,

33. Figures are as of 2001. Ashish Arora, *The Emerging Offshore Software Industries and the U.S. Economy*, *in* BROOKINGS TRADE FORUM 2005 OFFSHORING WHITE-COLLAR WORK 399, 401 tbl.2 (Susan M. Collins & Lael Brainard eds., 2005).

34. *See* Software Industry in India Project Team, The Globalization of Software: The Case of the Indian Software Industry 5 (2000), http://www.heinz.cmu.edu/project/india/pubs/Sloan_Report_final.pdf (noting that the Indian software industry "specializes in the export of low-end software development services"); *id.* at 18 ("[T]he Irish software industry does not appear to be as innovative or entrepreneurial as Israel.").

35. *See id.* at 23, 20–24 (observing that most Indian software revenue comes from exports of software services that are viewed by "[m]anagers at most of the US firms . . . [as] neither technologically very sophisticated nor critical to their business").

36. *See* Ashish Arora et al., International Outsourcing and Emergence of Industrial Clusters: The Software Industry in Ireland and India 14 (July 28, 2000), http://siepr.stanford.edu/conferences/silicon_papers/Aagtjuly.pdf (explaining the corporate tax structure of Ireland compared with the rest of the EU).

37. Simon Commander, *What Explains the Growth of a Software Industry in Some Emerging Markets* 10 (Ctr. for New and Emerging Mkts., London Bus. Sch., DRC Working Paper No. 22, 2003), *available at* http://www.london.edu/assets/documents/PDF/growth_of_software_industry.pdf ("A striking feature about the Israeli software industry is that it operates in a high number of technology niches . . . .").

38. *See* Dan L. Burk & Mark A. Lemley, *Policy Levers in Patent Law*, 89 VA. L. REV. 1575, 1581–89 (2003) (noting the high cost of research and development and large teams of scientists required for innovation in some industries).

39. *Id.* (contrasting the semiconductor and pharmaceutical industries to find fewer large corporate factors implicit in innovation).

40. Mann, *supra* note 10, at 965 (opting to research "innovation in the software industry itself . . . focus[ing] on firms like IBM, Microsoft, and their smaller competitors").

while a serious decline in the volume of start-up activity would not necessarily represent evidence of industry stagnation, a steady flow of new entrants would be in keeping with the industry's traditional "mixed" pattern of innovation.[41]

As Table 1 shows, new firms have entered the software industry at a healthy clip since 1970. Most importantly for purposes of this Article, none of the major signposts along the road to software patentability appear to have had any impact on the volume of start-up activity. After the earliest years tracked in the chart,[42] several important patent-related events occurred, but entry was still robust.

Table 1: Software Start-ups[43]

Year	Number of new firms formed
1970	102
1971	81
1972	123
1973	121
1974	119
1975	208
1976	224
1977	231
1978	342

41. *See* Campbell-Kelly, *supra* note 17, at 246 (discussing the growth of the software industry since 1960 and stating that though recent mergers and acquisitions have concentrated the software industry, "there remains an enormous number of very small players, and their number is growing"); Mann, *supra* note 10, at 970 ("A remarkable feature of the [software] industry as it has matured is its lack of concentration—a facet that has considerable implications for the competitive structure of the industry and its openness to innovation.").

42. The early data coincide with the 1972 Supreme Court decision in *Gottschalk v. Benson*, 409 U.S. 63 (1972). The *Benson* decision was decidedly "antipatent"; it cast serious doubt on the patentability of software. But in many ways, *Benson* simply confirmed what the long-standing assumption had been: that computer programs—or "algorithm inventions," as they were then referred to—were not a proper subject matter for patents. Under a crude "inverse relationship" hypothesis, *Benson* might have been expected to have led to greater start-up activity. Yet *Benson* was not enough of a "policy shock" to have any impact on start-up activity.

43. Data and methodology on file with author.

1979	428
1980	402
1981	480
1982	504
1983	530
1984	528
1985	551
1986	500
1987	430
1988	412
1989	436
1990	358
1991	379
1992	360
1993	337
1994	416
1995	469
1996	497
1997	353

More detailed data on start-up activity, including venture capital funding amounts, are shown in Table 2:

Table 2: Software Start-up Financing[44]

Year	# of first-time software financings	First-time financing amounts ($millions)	Total # of software financings	Total financing amounts ($millions)
1995	210	553	401	1,157
1996	305	822	655	2,283
1997	311	1,018	780	3,369
1998	309	1,181	931	4,493
1999	562	2,697	1,361	10,466
2000	829	6,023	2,047	24,435
2001	292	1,643	1,217	10,408
2002	260	1,178	959	5,306
2003	214	892	880	4,432
2004	234	1,180	887	5,247
2005	238	1,180	840	4,704

As the following graphs show, putting aside the 1999–2000 Internet financing bubble, software start-ups and financing activity have remained robust throughout the period of interest:

44. *See* PricewaterhouseCoopers & National Venture Capital Association, MoneyTree Report, http://www.pwcmoneytree.com/moneytree/nav.jsp?page=historical (providing historical trend data). The *MoneyTree Report* is based on data provided by Thompson Financial; the data is subject to change at any time and is periodically updated. PricewaterhouseCoopers & National Venture Capital Association, MoneyTree Report Definitions and Methodology, https://www.pwcmoneytree.com/moneytree/nav.jsp?page=definitions#method. The data included in the chart were collected in February 2006. A copy of the data is on file with the Texas Law Review.

Figure 1: Number of Software Financings (1995–2005)[45]

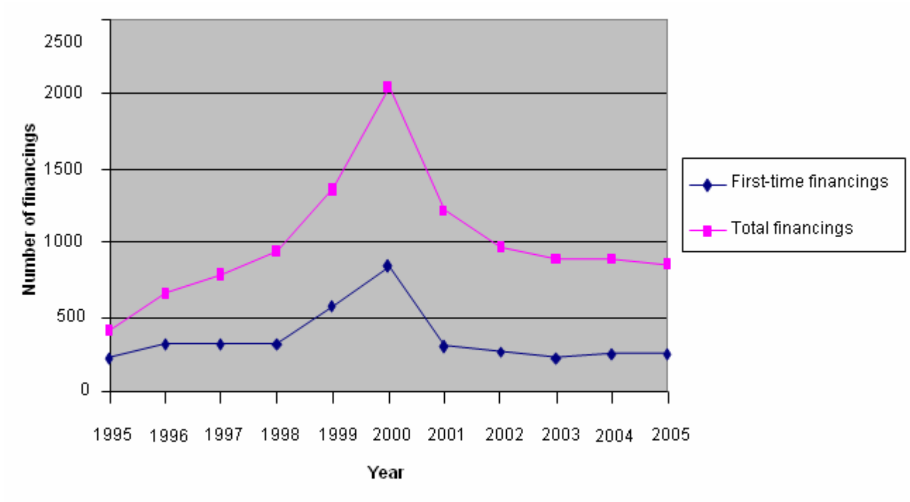
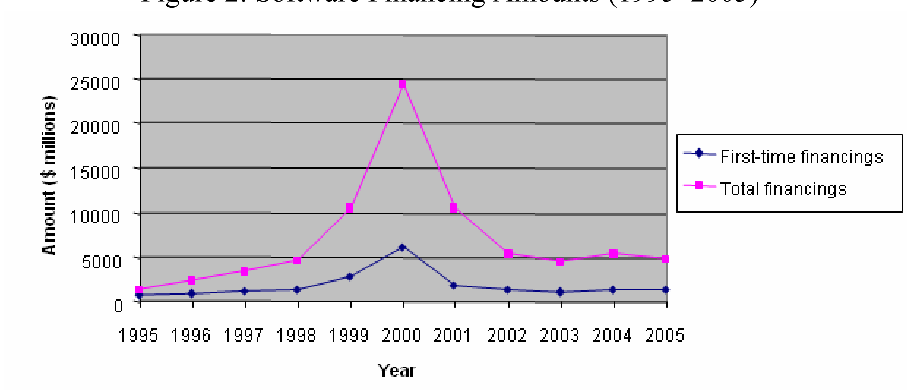


Figure 2: Software Financing Amounts (1995–2005)[46]



Indeed, a recent book by industry expert Michael Cusumano, aimed directly at software entrepreneurs, points out that currently, as has been true historically, only about 0.6% of all software-company business plans that are pitched to venture capitalists ultimately receive funding.[47] Cusumano lays out guidelines and suggestions for the budding software start-up founder[48]

45. *See supra* Table 2.

46. *See supra* Table 2.

47. *See* MICHAEL A. CUSUMANO, THE BUSINESS OF SOFTWARE: WHAT EVERY MANAGER, PROGRAMMER, AND ENTREPRENEUR MUST KNOW TO THRIVE AND SURVIVE IN GOOD TIMES AND BAD 198 (2004) (discussing statistics on venture capitalist financing from John Nesheim's research, which relied mainly on records from Saratoga Venture Finance).

48. *See id.* at 202–12 (presenting an eight-point checklist of elements that Cusumano deems necessary for a software start-up to succeed as a business and to raise venture capital).

and sprinkles cautionary tales of failure throughout[49]—solid evidence that entry is still competitive and robust in this important industry. At a minimum, Cusumano demonstrates that the industry is in no sense desperate for start-up activity; there are plenty of entrepreneurs trying to enter the industry.

D. Putting Legal Issues in Context

To review, patents have now become common in the software industry.[50] Despite earlier predictions,[51] the industry has not slowed to a crawl.[52] Why not?

This leads to my second observation. This one is just as prosaic, though maybe a bit more novel for law review readers: developments in technology and business have proven to be far more important than legal developments in the software industry since 1988. The almost miraculous increase in computing power (as predicted in Moore's law) that began in the 1970s has continued up to the present, creating the technological equivalent of the "permanent revolution" associated with a leading Mexican political party. The Internet is only the most recent manifestation of this. Earlier waves of innovation emanated from the minicomputer, the personal computer, video games, and artificial intelligence. Ever-increasing computing power and fatter broadband "pipes" for content are the revolutionary technological frontiers of today. This too shall pass—but another one will surely come, as long as computing power continues to grow.

On the business side, this breakneck innovation has, predictably, drawn the interest of people with a lot of money to invest. This investment push has exerted a huge influence on the software industry. External sources of finance mean that the industry's growth is not tied to its own current revenue. Independent entrepreneurs (or those who hope to be) pitch new product

---

49. *See, e.g.*, *id.* at 222–33, 242–46 (presenting three case studies of unsuccessful start-up software companies).

50. *See generally* Stuart J.H. Graham & David C. Mowery, *Intellectual Property Protection in the U.S. Software Industry*, *in* PATENTS IN THE KNOWLEDGE-BASED ECONOMY 219, 226–48 (Wesley M. Cohen & Stephen A. Merrill eds., 2003) (tracking and analyzing the development and expansion of software patentability since 1972).

51. *E.g.*, Bricklin, *supra* note 12. Some of these dire predictions are of a more recent vintage. *See, e.g.*, James Bessen & Robert M. Hunt, *An Empirical Look at Software Patents* 38–41 (Research on Innovation, Working Paper No. 03-17/R, 2004), *available at* http://www.researchoninnovation.org/swpat.pdf (predicting that increased strategic patenting might result in less innovation). For a response to Bessen and Hunt, see Robert W. Hahn & Scott Wallsten, A Review of Bessen and Hunt's Analysis of Software Patents (Nov. 2003), http://www.researchineurope.org/policy/hahn_wallsten.pdf.

52. *See* Campbell-Kelly, *supra* note 17, at 246 ("The anxieties expressed in the early 1990s about the effect patents would have on the software industry have not been realized."); Bronwyn H. Hall & Megan MacGarvie, *The Private Value of Software Patents* 22–23, 28–30 (Nat'l Bureau of Econ. Research, Working Paper No. 12195, 2006), *available at* http://papers.nber.org/papers/w12195 (calculating "abnormal" returns and stock market valuations for firms holding software patents and finding that software patents are often associated with increased firm value).

concepts and ideas for new companies to people with a lot of money to invest—typically principals of venture capital investment funds, or VCs for short.[53] The steady influx of VC money frees entrepreneurs from having to move up the chain of command in a large company, waiting for enough seniority and a rosy enough capital market inside the firm to pitch a new product idea.[54] It is easy to overlook how important this is. Not only does it allow talented and driven people to create new companies early in their careers, but the constant flow of start-up company activity keeps the pressure on the established firms—the incumbents—to stay current with new trends and continue to innovate. While a large installed base can be an advantage, having to maintain legacy products while trying to keep pace with new entrants can also pose a serious challenge to large software companies.

Taken together, technological change and VC financing have largely shaped the U.S. software industry. The legal system, and specific legal rules in particular, have played only a modest, background role. Perhaps out of hubris, those who predicted bad things for software with the coming of patents missed this essential truth. What it means for present debates is just this and nothing more: we are talking about tinkering at the margins. Patent law may exert some small force on the overall shape and direction of software. But in the retrospective vector analysis that historians might construct, it will be a distinctly minor force indeed.[55] Changing technological paradigms and capital market conditions, in particular within the VC industry, will be far more important.

And so we in the legal community should be reassured: there is little chance we will mess up this tremendous progress in any fundamental way. Yet of course we are not off the hook completely. For even if it is only at the margin, the legal rules do matter. It is incumbent upon us to do as well as we can to foster, promote, and support the development of this important industry. This we can do by designing and implementing intelligent, workable legal rules for its efficient operation. My point in emphasizing the predominance of technological and business forces in the industry is merely to remind us that whatever legal rules we design will play out in the context of a dynamic, expanding industry. We would be wise to keep this context in mind when designing the rules (e.g., regarding patent scope) and in predicting their impact, as I do in the next Part.

---

53. *See* Mann, *supra* note 10, at 974–75 (stating that when software entrepreneurs exhaust their resources they turn to institutional investors, most commonly venture capitalists).

54. *See* CUSUMANO, *supra* note 47, at 195, 195–214 (offering advice for entrepreneurs, "intrapreneurs," and venture capitalists on evaluating "the viability of a software start-up and its business plan").

55. *Cf.* CAMPBELL-KELLY, *supra* note 29, at 107–08 (discussing intellectual property protections in only two pages of a 311-page history of software); MARK A. LEMLEY ET AL., SOFTWARE AND INTERNET LAW 7–9 (3d ed. 2006) (describing the historical development of software without reference to intellectual property law).

III. Enablement and "Written Description" in the Software Industry

So legal developments exert only a marginal effect on the direction of the software industry. But this is not the same as saying they are irrelevant. By definition, in a close situation, with other factors roughly in equipoise, legal rules can make a difference. One important set of rules involves the strength, or scope, of each patent. Of the various doctrines that collectively determine patent scope, those involving disclosure are highly important. (Others include utility and claim interpretation, including the doctrine of equivalents.) Of particular contemporary importance are doctrines governing the relationship between the disclosure portion of a patent (the specification) and the legal claims that follow it.[56] These matters are governed by § 112 of the Patent Act.[57] I consider these doctrines in this Part.

A. Very Rapid History of Disclosure Law: Emergence of the "Classical" Enablement Standard

Early in their history, patents were special royal favors or privileges handed out to those whom the sovereign wanted to reward.[58] Because of this, the relevant state authorities did not consider it important to publish detailed disclosures of patented inventions.[59] Inventions were valued for the contribution they made to the state, in the form of practical benefits that flowed from their construction and operation.[60] The idea of "letters patent"

---

56. Technically, as the statute makes clear, the claims are not distinct from the disclosure portion of the patent. But by common usage, practitioners often refer to the disclosure portion as the "specification" to distinguish it from the claims at the end, so I use that terminology here. *See* 35 U.S.C. § 112 (2000) ("The specification shall conclude with one or more claims . . . ."). The Federal Circuit has consistently in recent years used "written description," or more verbosely, "written description portion of the specification" in its opinions, but I find this confusing and so resist it. The shorter phrase "written description" is easily mistaken for the written description *doctrine*, doubling the confusion associated with that term; while "written description portion of the specification" uses six words where one ("specification") would serve. I am all for slavish consistency when it will serve a useful purpose, but it becomes for me a small-minded defense against a mythical hobgoblin when it takes the form of pedantic repetition.

57. This section reads in part:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same, and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Id.

58. *See* Herbert Hovenkamp, *Technology, Politics, and Regulated Monopoly: An American Historical Perspective*, 62 TEXAS L. REV. 1263, 1264 (1984) (noting that historically, the Crown granted patents largely "as payment for certain political favors").

59. *See* John N. Adams & Gwen Averley, *The Patent Specification: The Role of* Liardet v. Johnson, 7 J. LEGAL HIST. 156, 158–59 (1986) (adopting the account of Wyndham Hulme that technical specifications, or disclosures, were not required at the outset because patents previously were used to grant monopolies for industries, and indicating that disclosure requirements were only adopted after inventors sought to distinguish their patent applications from competitors).

60. *See, e.g.*, Hovenkamp, *supra* note 58, at 1264 (indicating that patents were granted in part to encourage imports of manufactured goods); E. Wyndham Hulme, *On the Consideration of the*

being valuable for their *information content* did not arise until the eighteenth century in Britain.[61] And when the idea did emerge, it was as part of a general effort on the part of the British courts to limit and restrain the rights of patentees—a hangover, so to speak, from the "antimonopoly" era of the seventeenth century that put such a distinctive mark on early British patent law.[62] Thus British patent cases through the early nineteenth century imposed harsh disclosure rules on patentees that had the overall effect of invalidating a good number of the patents that those courts were called upon to review.[63] A leading British treatise from 1830, for example, goes on at length enumerating the detailed and technical grounds on which patent specifications may be deemed insufficient.[64] The courts were quick in the older cases to impute moral culpability to inventors in such circumstances; they did not inquire into the benefits of the inventions at issue, or practical problems of construction or implementation that might have been present in carrying out the invention.[65] Although in the latter part of this period the courts sometimes apologized for these results,[66] restrictive disclosure rules were an established part of the law.

Patent Grant Past and Present, 13 L.Q.R. 313, 317 (1897) (recounting that patents were originally granted for industries "such as copper, lead, gold, and silver mining" or for manufacturers), *quoted in* Adams & Averley, *supra* note 59, at 158.

61. *See* Adams & Averley, *supra* note 59, at 156–62 (arguing that the English patent enrollment requirement, which first appeared in 1711, reflected a new doctrine valuing the instructive potential of patent law).

62. The harsh views of the British patent authorities were noted by an American treatise writer in 1810. *See* THOMAS GREEN FESSENDEN, AN ESSAY ON THE LAW OF PATENTS FOR NEW INVENTIONS iv–v (2d ed., Boston, Charles Ewer 1822) (1810) ("In Great-Britain, however, the prejudices which formerly subsisted against patents for new and useful inventions seem to have subsided, and the [nation was] actuated by that sound and liberal policy, which is alone calculated to call forth and secure to the use of the public the exertions of genius . . . ."). Fessenden may have been overoptimistic in his assessment that the British courts had changed their ways, however. *See, e.g.*, RICHARD GODSON, A PRACTICAL TREATISE ON THE LAW OF PATENTS FOR INVENTIONS AND OF COPYRIGHT 122 (2d ed., London, William Benning & Co. 1844) (1823) ("If, in a manufacture something well known be used, and the inventor give a design of it which appears to be of a different thing, though he means that the thing known should be used, the specification is in terms ambiguous; and it will be considered as being worded with an endeavour to conceal the invention and deceive the public.").

63. *See* Turner v. Winter, (1787) 1 Term. Rep. 602 (Buller, J.), *reprinted in* FESSENDEN, *supra* note 62, at 190 ("Many cases . . . have been decided against the patentees, [for] not having made a full and fair discovery of their inventions. Wherever the patentee has made a fair disclosure, I have always had a strong bias in his favour, because in that case he is entitled to the protection which the law gives him.").

64. EDWARD HOLROYD, A PRACTICAL TREATISE OF THE LAW OF PATENTS FOR INVENTIONS 108–30 (London, John Richards 1830).

65. *See, e.g.*, GODSON, *supra* note 62, at 133 ("If any considerable part of a manufacture be *unnecessary* to produce the desired effect, it will be presumed that it was inserted [into the specification] only with a view to perplex and embarrass the enquirer.").

66. *See, e.g.*, Morgan v. Seaward, (1837) 2 M. & W. 544, 562, 150 Eng. Rep. 874, 881 (Exch.) (Parke, B.) ("We cannot help seeing on the face of this patent . . . that an improvement in steam engines is suggested by the patentee, and is part of the consideration [offered by him in exchange for the patent] grant: and we must reluctantly hold, that the patent is void, for the falsity of that suggestion.").

American jurists, sensitive to the different conditions they faced in the new world, diverged from the British doctrine. For them, patents were not a vestige of corrupt political maneuvering, but instead represented the flourishing of a vital, energetic young nation.[67] Perhaps in this they were influenced by the perception that the new nation faced a chronic labor shortage, which mechanical inventions might help to redress.[68] Whatever their motivation, Justice William Story and other notable judges in the early federal period fashioned a much more liberal disclosure standard in patent law.[69] They looked for a real inventive contribution, and if they found it, they did not scrutinize the inventor's specification for technical defects or minor deficiencies.[70] This is not to say that they were unreservedly pro-

67. *See* R. KENT NEWMYER, SUPREME COURT JUSTICE JOSEPH STORY: STATESMAN OF THE OLD REPUBLIC 139 (1986) (noting that the Constitution and early statutes made it "clear that law was expected to serve the national interest by encouraging invention" and arguing that circumstances like "the growing emphasis on economic individualism [and] the explosion of invention" caused "American lawmakers to liberalize English law to fit American needs").

68. For the *locus classicus* for this hypothesis, see H.J. HABBAKUK, AMERICAN AND BRITISH TECHNOLOGY IN THE NINETEENTH CENTURY: THE SEARCH FOR LABOR-SAVING INVENTIONS (1962), in which the author extensively discusses the claim that a scarcity of industrial labor accelerated technological investment in nineteenth-century America. *See also* DAVID E. NYE, AMERICA AS SECOND CREATION: TECHNOLOGY AND NARRATIVES OF NEW BEGINNINGS 4, 11, 4–11 (2003) (arguing that "after the Revolution, and particularly in the nineteenth century, Americans developed another way to understand their settlement of the western hemisphere: as the technological transformation of untouched space," according to which "[a] surplus of mechanical force was taken to be axiomatic, making possible new landscapes, boomtowns, sudden profits, personal success, and national progress").

69. An example of the liberal disclosure standard can be seen in *Ames v. Howard*, 1 F. Cas. 755, 756 (C.C.D. Mass. 1833) (No. 326):

> [I]f the court can clearly see, what is the nature and extent of the claim, by a reasonable use of the means of interpretation of the language used, then the plaintiff is entitled to the benefit of it, however imperfectly and inartfully he may have expressed himself. And for this purpose we are not to single out particular phrases standing alone, but to take the whole in connexion.

Id. at 756; *see also* NEWMYER, *supra* note 67, at 140 ("Story never abandoned his determination to inject precision into patent law, but he did—following the lead of [Justices John] Marshall, [Henry] Baldwin, and [Brockholst] Livingston and the advice of [Daniel] Webster—modify his early strictness in favor of a more liberal and pragmatic approach.").

70. *See, e.g.*, Carver v. Braintree Mfg. Co., 5 F. Cas. 235, 240 (C.C.D. Mass. 1843) (No. 2485) (involving a patent on an improvement to the cotton gin). One component of the invention had two surfaces, and the patent specification described the distance between the surfaces in terms of the length of a cotton fiber. *Id.* at 237. Defendant argued that this rendered the patent invalid because some varieties of cotton were much longer than others. *Id.* Justice Story, riding circuit in Massachusetts, refused to invalidate the patent on this ground and referred the issue to the jury. *Id.* at 242. His discussion of the point rejected the formalism associated with the British cases and opened the door to a much more forgiving interpretation of the patent specification:

> I should suppose, that the inequalities of the different fibres of the same kind of cotton would not necessarily present an insurmountable difficulty. It may be, that the adjustment should be made, according to the average length of the fibres, or varied in some other way. But this is for a practical mechanic to say, and not for the court. What I mean, therefore, to say on this point is, that, as a matter of law, I cannot say, that this description is so ambiguous, that the patent is upon its face void. It may be less perfect and complete, than would be desirable; but still it may be sufficient to

patentee, however. When the facts demanded, they were quite willing to invalidate a patent whose claims appeared to far outrun the value of the information disclosed in its specification. This was the case, for example, with an 1840 case over a patent for a new type of textile loom.[71]

> enable a skilful mechanic to attain the end. In point of fact, is it not actually attained by the mechanics employed by Carver, without the application of any new inventive power, or experiments? If so, then the objection could be answered as a matter of fact or a practical result.

Id. at 237. There is a good discussion of the policy rationale behind this approach in Story's opinion in *Ames*. He states:

> Patents for inventions are not to be treated as mere monopolies odious in the eyes of the law, and therefore not to be favored; nor are they to be construed with the utmost rigor, as strictissimi juris. The constitution of the United States, in giving authority to congress to grant such patents for a limited period, declares the object to be to promote the progress of science and useful arts, an object as truly national, and meritorious, and well founded in public policy, as any which can possibly be within the scope of national protection. Hence it has always been the course of the American courts, (and it has latterly become that of the English courts also,) to construe these patents fairly and liberally, and not to subject them to any over-nice and critical refinements. The object is to ascertain, what, from the fair sense of the words of the specification, is the nature and extent of the invention claimed by the party; and when the nature and extent of that claim are apparent, not to fritter away his rights upon formal or subtile objections of a purely technical character.

Ames, 1 F. Cas. at 756.

71. Stone v. Sprague, 23 F. Cas. 161 (C.C.D.R.I. 1840) (No. 13,487) (Story, J.). In *Stone*, the patentee had apparently invented a new way to transmit power between two components of a loom, the "reed" and the "yarn beam." The patent's specification was not so limited, however. The patentee in effect claimed *all ways* of transmitting power between these components—and Story invalidated the patent:

> [I]f [the specification] be construed to include all other modes of communication of motion from the reed to the yarn beam, and for the connexion of the one to the other generally, it is utterly void, as being an attempt to maintain a patent for an abstract principle, or for all possible and probable modes whatsoever of such communication, although they may be invented by others, and substantially differ from the mode described by the plaintiff in his specification. A man might just as well claim a title to all possible or practicable modes of communicating motion from a steam-engine to a steamboat, although he had invented but one mode; or, indeed, of communicating motion from any one thing to all or any other things, simply because he had invented one mode of communicating motion from one machine to another in a particular case.

Id. at 162. Justice Story was not alone in this respect; other cases from this era show courts acting equally suspicious of overbroad claims. *See, e.g.*, Sullivan v. Redfield, 23 F. Cas. 357 (C.C.D.N.Y. 1825) (No. 13,597). This was one of many cases of infringement of a patent on steamboat technology. It appears to have been an attempt to enforce a federal patent over one way of towing barges by steamboats—likely a remnant of the long struggle over state steamboat franchises culminating in the Supreme Court's famous opinion in *Gibbons v. Ogden*, 22 U.S. 1 (1824). In *Sullivan*, Justice Thompson, on circuit in New York, refused to grant an injunction, arguing that the specification was probably too vague to support a valid patent:

> The patentee cannot surely claim as his invention the towing of one boat after another. But the manner of attaching the two together would seem to be the right he asks to have secured to him. If he has discovered any important improvement in this respect, it should have been described in the specification with more certainty and precision. To say that the two boats must be so attached as to be kept always at convenient distance, does not seem to be that full explanation which, after the expiration of the patent, would leave the public much wiser than they were before. What is a convenient distance, and the particular manner of attaching the one to the

This liberal but balanced view of the law of disclosure prevailed throughout the nineteenth century.[72] By the late nineteenth century, it had come to be stated as the undue experimentation standard, invoked famously in *The Incandescent Lamp Patent* case of 1895.[73] Under this standard, a patent claim is valid if a skilled artisan can construct all or most of the embodiments covered by a patent claim without excessive or undue experimentation.[74] (By this time, claims had taken on their modern form—in full text, describing the outer boundaries of the invention, placed at the end of the specification.) Undue experimentation carried forward the flexible test of the early years. It admitted some experimentation, again in distinction to the early British cases. Yet it also provided that a claim was invalid if it would take too much additional work for someone in the relevant field to move from the description in the specification to the actual making or use of the claimed invention. Sometimes, the then-emerging law of "invention" (today, nonobviousness) was invoked in service of this test: if the research required to span the gap from specification to claimed embodiments was so great that it constituted patentable invention, then the specification was *a fortiori* deficient.[75]

The story of the disclosure requirement in the twentieth century is one of gradual refinement and adaptation.[76] Doctrinal nuance was introduced in the application of basic disclosure principles to emerging new scientific and technical fields, even as the foundational principle of undue experimentation remained fixed in place. So, for example, the very large families of compounds that came to be claimed (using the Markush claim format) in chemical patent practice gave rise to some important cases in the law of "chemical enablement," but the basic contours of the doctrine did not change.[77] This was true also for biotechnology, which emerged later in the

---

> other, will still have to be ascertained by experience. If, according to the patent, the invention claimed is an improvement in the steam tow-boat, the specification, to be complete, should describe the one previously in use, that it might be seen clearly in what the improvement consisted, as the patent cannot cover more than the improvement claimed. These are some of the objections to the patent itself, which present such strong doubts in the mind of the court, as to its validity, that it is deemed improper to interpose an injunction until the validity of the patent has been tried at law.

Sullivan, 23 F. Cas. at 361.

72. *See generally* 2 WILLIAM C. ROBINSON, THE LAW OF PATENTS FOR USEFUL INVENTIONS 85–93 (Boston, Little, Brown, and Company 1890) (summarizing disclosure cases).

73. Consol. Elec. Light Co. v. McKeesport Light Co., 159 U.S. 465 (1895).

74. *See id.* at 474–75 (expounding as the standard for claim validity whether there is sufficient clarity and precision in the patent claim to permit a person to use the discovery without undertaking independent experimentation).

75. *See* 2 ROBINSON, *supra* note 72, at 91–92 (noting that a patent is invalid if "inventive skill" is required to practice the invention described in the specification).

76. *See* DONALD CHISUM, CHISUM ON PATENTS § 7.02[4] (1978 & Supp. 2006).

77. *See, e.g.*, *In re* Gardner, 427 F.2d 786, 789 (C.C.P.A. 1970). In *Gardner*, the inventor claimed to have discovered antidepressant activity in 2-aminomethyl-1,3-benzodioxole compounds. *Id.* The PTO rejected the patent application in part because the specification failed to disclose how to use the invention, i.e., what dosages of the drug were effective. *Id.* The court found that one

century.[78] And the few early cases that applied disclosure law in the area of software patents recited similar principles as well,[79] though these have been criticized in the academic literature.[80]

Beginning in the 1960s, a small number of patent opinions were premised on a written description requirement distinct from "classical" enablement.[81] But these were few in number and limited to a certain narrow fact situation. Typically, an inventor filing an application overseas would file a counterpart application in the United States, adding or modifying one or more claims. If the added claims lacked solid support in the specification, courts would say that the claims had not been adequately described in the original specification and were therefore invalid. Many of these cases involved chemical inventions.[82] It seems evident that only a minor adjustment would have been necessary to characterize this as an enablement issue.[83]

skilled in the art could, after "*a great amount of work*," eventually find out how to use the invention. *Id.* The court held that "the law requires that the disclosure in the application shall *inform* them *how* to use, not how to find out how to use for themselves." *Id.*

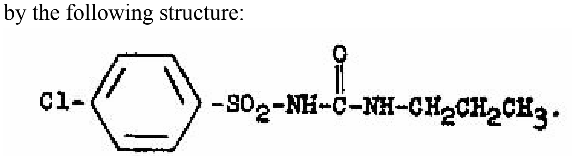
78. *See* David J. Weitz, *The Biological Deposit Requirement: A Means of Assuring Adequate Disclosure*, 8 BERKELEY TECH. L.J. 275, 276–79, 281–97 (1993) (discussing the basic "undue experimentation" requirement in biotechnology patents, describing nuances that developed in the case law, and suggesting a disclosure rule for such patents).

79. *See, e.g.*, N. Telecom v. Datapoint Corp., 908 F.2d 931, 941 (Fed. Cir. 1990) ("A decision on the issue of enablement requires a determination of whether a person skilled in the pertinent art, using the knowledge available to such a person and the disclosure in the patent document, could make and use the invention without undue experimentation.").

80. *See, e.g.*, Dan L. Burk & Mark A. Lemley, *Is Patent Law Technology-Specific?*, 17 BERKELEY TECH. L.J. 1155, 1164 (2002) ("[I]t is remarkable that the Federal Circuit is willing to find the enablement requirement satisfied by a patent specification that provides no guidance whatsoever on how the software should be written.").

81. *See, e.g.*, *In re* Sus, 306 F.2d 494, 497 (C.C.P.A. 1962) (affirming rejection of a patent application because an applicant cannot claim a broader invention than that set forth in the written description claimed in his specification); Vernay Labs. v. Indus. Elec. Rubber Co., 234 F. Supp. 161, 163 (N.D. Ohio 1964) (holding that when there was a question of material fact as to the support contained in the written description of the patent, summary judgment could not be granted).

82. *In re* Ruschig, 379 F.2d 990 (C.C.P.A. 1967), involved a family of compounds described by the following structure:



83. Indeed, the examiner's rejection at issue in *Ruschig* might well have been for lack of enablement: the application, he states, "is rejected as having no specific support in this disclosure." *Id.* at 992. He went on to state: "The compound of claim 13 is not named or identified by formula and it can find support only as choices made between the several variables involved. This is not regarded as adequate support for a specific compound never named or otherwise exemplified in the specification as filed." *Id.* The examiner then cited *Fried*, a similar chemical case holding that a parent application that suggested but did not disclose the making of specific compounds failed to provide support for those compounds as claimed in a later application. *Id.* (citing *In re* Fried, 312 F.2d 930, 936–37 (C.C.P.A. 1963)). According to the court, "the invention disclosed in the appealed application was not in fact sufficiently disclosed in the parent application" as provided under § 112, and thus, the claim was invalid. *Fried*, 312 F.2d at 937.

After all, from an early time, the enablement requirement has been measured from the date a patent is filed; hence the later filed claims in these cases might well have been said not to have been enabled by the earlier application.[84] But this seemingly sensible solution was foreclosed because the opinions in the cases insisted that the earlier application did enable the later claims.[85] The defect in disclosure was characterized as a separate matter: that the specification did not properly identify or focus on the later claimed subject matter.[86] I will have more to say on this complex issue later; the important point here is just to note that it entered the law through these cases.

This additional disclosure standard—identified in later cases as the written description requirement—lay dormant for a good while. It was not until the 1990s that this requirement was dusted off and applied with a vengeance, giving the settled law of enablement a violent twist. First in cases that followed the older pattern (initial specification, then broadened claim), and later in a series of biotechnology cases, patents were found enabled but invalid for failure to meet the written description requirement.[87] The doctrine has grown so quickly, and been applied so widely, that enablement rarely decides disclosure cases in the Federal Circuit today. The written description requirement now dominates the scene.

1. Infatuation with Written Description.—In the next section, I describe a subset of today's written description cases that really do present novel issues for traditional enablement law. Unfortunately, the Federal Circuit has now gone well beyond these challenging cases in fashioning written description doctrine. That court has chosen to extend written description principles into many cases best decided under traditional enablement doctrine. For

---

84. *See, e.g.*, Consol. Elec. Light Co. v. McKeesport Light Co., 159 U.S. 465, 475 (1895) (holding that a patent application must include a description that enables others to construct the patented device).

85. *See, e.g.*, *In re* Ruschig, 379 F.2d at 995–96 (holding that the basis for rejection was not an issue of enablement, as presumed by the appellants, but rather a question of whether the specification clearly conveyed that the appellants had invented that specific compound); *In re* Fried, 312 F.2d at 937 (concluding that the "invention disclosed in the appealed application was not in fact sufficiently disclosed in the parent application" and therefore the claims were not entitled to the benefit of the filing date of the parent application); *In re* Sus, 306 F.2d at 505 (affirming the holding of the board that the claims were unpatentable under 35 U.S.C. § 112 as they claimed a broader invention than set forth in the specification).

86. *See, e.g.*, *In re* Ruschig, 379 F.2d at 995–96 (deciding patent validity based on the written description requirement instead of enablement); *In re* Fried, 312 F.2d at 937 (same); *In re* Sus, 306 F.2d at 505 (same).

87. *See* Janice M. Mueller, *The Evolving Application of the Written Description Requirement to Biotechnological Inventions*, 13 BERKELEY TECH. L.J. 615, 633 (1998) (analyzing the Federal Circuit's "significant departure" from established precedent in holding a patent enabled but invalid for failure to meet the written description requirement). *See generally* Burk & Lemley, *supra* note 80, at 1156 (noting that the Federal Circuit "has imposed stringent enablement and written description requirements on biotechnology patents that do not show up in other disciplines").

example, consider *University of Rochester v. G.D. Searle & Co.*[88] In this case, scientists had discovered a novel physiological mechanism for blocking the action of a common receptor, which held the promise of delivering a new family of painkilling drugs devoid of some negative side effects that had plagued the prior art.[89] The difficulty with the patent application in the case was that no actual painkilling compounds were identified in the specification; it listed only the mechanism by which the as-yet hypothetical drugs would operate, claimed in process terms.[90] The Federal Circuit rightly affirmed the invalidity of the claims at issue.[91] It wrongfully based its decision on the written description requirement.[92] This case, easily decided under conventional enablement doctrine, stands as a fine example of unnecessary doctrinal proliferation.

University of Rochester is in no sense unique. Another case, *Enzo Biochem, Inc. v. Gen-Probe Inc.*,[93] centered on the deposit of microbiological materials claimed in a patent—an issue litigated, fleshed out, and settled in a series of cases in the 1970s and 1980s concerning deposit requirements under the enablement standard.[94] Another example is *Moba v. Diamond Automation, Inc.*,[95] a written description case that might easily be described in terms of conventional enablement law.[96] To the same effect is *Reiffin v.*

88. 358 F.3d 916 (Fed. Cir. 2004).

89. *Id.* at 917–18.

90. *Id.* at 919.

91. *Id.* at 930.

92. *Id.* at 920.

93. 323 F.3d 956 (Fed. Cir. 2002); *see also* Regents of the Univ. of Cal. v. Eli Lilly & Co., 119 F.3d 1559, 1567 (Fed. Cir. 1997) (requiring a precise definition of a DNA sequence in the patent specification for it to meet the written description requirement). I would argue that (1) *Lilly* was actually an enablement case, and (2) it was wrongly decided under whichever doctrine governed it. *See* Mueller, *supra* note 87, at 633–49 (arguing that *Lilly* was wrongly decided for various reasons).

94. *See, e.g.*, *In re* Argoudelis, 434 F.2d 1390, 1394 (C.C.P.A. 1970) (deciding a microorganism claim under the enablement standard). Such a deposit has been considered adequate to satisfy the enablement requirement of 35 U.S.C. § 112 (2000), when a written description alone would not place the invention in the hands of the public and physical possession of a unique biological material is required. *See, e.g.*, *In re* Wands, 858 F.2d 731, 735 (Fed. Cir. 1988) ("Where an invention depends on the use of living materials . . . it may be impossible to enable the public to make the invention (i.e., to obtain these living materials) solely by means of written disclosure."); *In re* Lundak, 773 F.2d 1216, 1220 (Fed. Cir. 1985) ("When an invention relates to a new biological material, the material may not be reproducible even when detailed procedures and a complete taxonomic description are included in the specification."). *See generally* Berge Hampar, *Patenting of Recombinant DNA Technology: The Deposit Requirement*, 67 J. PAT. & TRADEMARK OFF. SOC'Y 569, 607 (1985) ("The deposit requirement is a nonstatutory mechanism for ensuring compliance with the 'enabling' provision under 35 U.S.C. § 112.").

95. 325 F.3d 1306, 1321 (Fed. Cir. 2003) (per curiam) (noting that there is no written description problem in the specification because each element of the claim was described fully in the specification in terms intelligible to one skilled in the art). Note that Judge Randall Rader has consistently dissented from recent developments in written description doctrine. *See, e.g.*, *id.* at 1322 (Rader, J., concurring) ("By making written description a free-standing disclosure doctrine, [the Federal Circuit] produces numerous unintended and deleterious consequences.").

96. *See* Stephen J. Burdick, Moba v. Diamond Automation, Inc.*: Questioning the Separate Written Description Requirement*, 19 BERKELEY TECH. L.J. 133, 151 (2004) ("*Moba* illustrates the

Microsoft Corp.,[97] a software patent case in which the Federal Circuit stated that the purpose of the written description requirement is to "ensure that the scope of the right to exclude, as set forth in the claims, does not overreach the scope of the inventor's contribution to the field of art as described in the patent specification."[98] One familiar with the older enablement cases[99] could be forgiven for thinking upon reading this passage, "I thought that's what enablement was all about."[100]

In deciding cases such as these, the Federal Circuit has performed a double disservice. First, it has violated the ancient principle of Occam's Razor, or intellectual parsimony: do not invoke two concepts when one will do.[101] Second, by introducing a new doctrine into a settled area of law, it has opened the door to creative arguments and novel strategies that threaten basic, established principles. A lawyer handed a case that is a sure loser under established enablement principles need only reframe the issue as a written description question. Appealing to this novel and unformed body of law just might get the desired result. In the process, the meandering process of common law evolution—so prized when new issues demand case-by-case legal development—is unleashed unnecessarily. The resulting destabilization of established principles comes at a steep cost: the loss of certainty and predictability.

The second disservice is this: in its zeal to defend and expand written description doctrine, the Federal Circuit has diverted attention from an important issue. The fact is that some of the early written description cases—in particular, *Gentry Gallery*—did present facts that posed a challenge to traditional enablement doctrine.[102] Here was a unique issue that really did call for some sort of doctrinal innovation. But the challenging

---

problems associated with the separate written description requirement. . . . Its effects are redundant with the enablement and new matter requirements of patent law.").

97. 214 F.3d 1342 (Fed. Cir. 2000).

98. *Id.* at 1344–45.

99. *See, e.g.*, *In re* Fisher, 427 F.2d 833, 839 (C.C.P.A. 1970) (holding that the disclosure standard "requires that the scope of the claims must bear a reasonable correlation to the scope of enablement provided by the specification to persons of ordinary skill in the art").

100. *See also* Union Oil Co. of Cal. v. Atl. Richfield Co., 208 F.3d 989, 997 (Fed. Cir. 2000) ("The written description requirement does not require the applicant 'to describe exactly the subject matter claimed, [instead] the description must clearly allow persons of ordinary skill in the art to recognize that [he or she] invented what is claimed.'" (alterations in original) (quoting *In re* Gosteli, 872 F.2d 1008, 1012 (Fed. Cir. 1989))).

101. This is the spirit of a kindred critique of written description doctrine. *See* Robin Feldman, *The Inventor's Contribution*, 2005 UCLA J.L. & TECH. 6 (critiquing the need for a separate written description requirement).

102. *See* Gentry Gallery, Inc. v. Berkline Corp., 134 F.3d 1473, 1474–75 (Fed. Cir. 1998) (explaining that the patentee alleged infringement of its patent for a reclining sofa; that the patent claims as originally filed required that the recliner controls be located upon a center console of the sofa; that claims later amended to permit the controls to be located outside the console; that the purpose of the amendment was to cover a competitor's product that located the recliner controls off the center console; and that the patent was invalidated for failure to comply with the written description requirement).

issues presented by this new type of case are now lost in the haze of confusion caused by applying the written description requirement virtually every time a § 112 issue is raised in litigation. The unique content of these new cases—the legitimate subject of new doctrinal expansion—is obscured in a much wider (and unnecessary) battle over what written description means. There is no reason for this to be so. As I describe in the next section, the novel issues raised by cases such as *Gentry Gallery* can be dealt with separately, without the need for doctrinal clutter. To see this, we have to now take a look at the nature of the novel issues these cases present.

2. Nonenablement Written Description Cases: "Misappropriation by Amendment."—The nub of the issue is how written description differs from what I have called classical enablement. To be frank, the courts have not been especially helpful in providing an answer to this crucial question. The perceived need for something beyond enablement originated in the early chemical cases; perhaps it was a function of the very liberal chemical enablement standard.[103] The various standards that have been announced all suffer from a lack of analytical rigor. The most common one centers on the notion of "possession."[104] A claim will fail under the written description requirement if the inventor cannot show in the specification that the claimed

103. In *In re Fried*, 312 F.2d 930 (C.C.P.A. 1963), for example, the parent specification whose content was the key to the case claimed a large family of chemical compounds in the usual manner, by virtue of a structural chemical formula and a "Markush" claim. *Id.* at 932. This type of claim, named after an old Patent Board opinion, recites that individual elements are "selected from the group consisting of" a list of elements. *Id.*; *see also Ex parte* Markush, 1925 Dec. Comm'r Pat. 126. Given the large number of constituents in a complex chemical structural formula, and the ability to claim each constituent as one among a list of variables, such a claim can easily encompass thousands of embodiments. The defect in cases such as *In re Fried* was a lack of guidance about how to choose among the long list of claimed variables:

> [W]hile appellant's parent application indicates that the 17-keto group of the steroid "compounds of the invention" therein may be converted to the corresponding . . . steroids, it is clear, as pointed out by the examiner, that there is no disclosure of a specific method of preparation of the specific compounds claimed here and, as pointed out by the Board of Appeals, that there is no disclosure of a specific working example for preparing one compound here claimed. Since compounds here claimed are not named or identified by formula in the parent application, *they can find support only as choices are made between the several variables involved*.

In re Fried, 312 F.2d at 936 (emphasis added). Note the conceptual similarity between this thought and the statement from the seminal enablement case, *The Incandescent Lamp Patent*:

> If, as before observed, there were some general quality, running through the whole fibrous and textile kingdom, which distinguished it from every other, and gave it a peculiar fitness for the particular purpose, the man who discovered such quality might justly be entitled to a patent; but that is not the case here. An examination of materials of this class carried on for months revealed nothing that seemed to be adapted to the purpose . . . .

159 U.S. 465, 475 (1895).

104. *See* Vas-Cath Inc. v. Mahurkar, 935 F.2d 1555, 1563–64 (Fed. Cir. 1991) ("The purpose of the 'written description' requirement is broader than to merely explain how to 'make and use'; the applicant must also convey with reasonable clarity to those skilled in the art that, as of the filing date sought, he or she was in possession *of the invention*.").

subject matter was within his or her possession when the patent was filed.[105] As distinct from enablement, the possession requirement seems to suggest that the inventor must have a firm grip on the claimed subject matter, as evidenced by the specification. This notion of "having hold of" is distinct from the traditional enablement test, which is concerned with what the specification *teaches*.[106] When one reads the cases, it is apparent that the Federal Circuit has become convinced that an inventor can teach a great deal in a specification that he or she does not have a firm grasp of or does not "consider to be his or her invention."[107] In other words, there are embodiments that fall within the scope of a specification's teachings that cannot be said to be fair game for the patentee to claim. It is this notion of the shortcomings of the enablement standard in maintaining a commensurate relationship between specification and claims that has given rise to the written description revolution.

In some cases, a patentee has filed a specification that hones in on a particular set of embodiments, which are claimed in an initial application. Then one or more claims is amended to cover either a competitor's product or an item suggested by the prior art. In each of these, the original application, by failing to claim initially the technology later claimed in an amendment, signals that these embodiments are not particularly important or even relevant to the inventor. It is the actions of a third party that give them salience. When a third party introduces the disclosed but unclaimed variant, it suddenly acquires salience for the applicant. This seems unfair to the courts involved.[108] By virtue of a claim amendment, a patentee attempts to encompass embodiments he or she did not envision as belonging to the real heart of the invention.

This type of unfairness might be described as "misappropriation by amendment." The patentee attempts to appropriate the effort of a competitor or the contributions of a prior art reference, such as an earlier technical article or patent. The unfairness is straightforward: these embodiments are more

105. *Id.*

106. *See id.* at 1563 (reaffirming that 35 U.S.C. § 112 (2000) requires a written description of the invention, which is separate and distinct from the enablement requirement). For a fine recent statement of this, see Timothy R. Holbrook, *Possession in Patent Law*, 59 SMU L. REV. 123 (2006) (lauding elevation of (proof of) "possession" over "teaching" function in § 112 law). Holbrook may well be correct that the "teaching" function of § 112 is overrated, but "possession" is as much a part of enablement as of "written description." In addition, too much focus on actual possession of claimed embodiments undermines the important "option function" of patent applications, described below in notes 110 and 113.

107. Amtel Corp. v. Info. Storage Devices, 198 F.3d 1374, 1381 (Fed. Cir. 1999).

108. *See, e.g.*, Univ. of Rochester v. G.D. Searle & Co., 358 F.3d 916, 930 (Fed. Cir. 2004) (affirming the district court's grant of summary judgment that the university's patent was invalid on written description grounds); Regents of the Univ. of Cal. v. Eli Lilly & Co., 119 F.3d 1559, 1575 (Fed. Cir. 1997) (affirming the district court's finding of invalidity of claims based on the written description requirement).

properly attributed to the labor of others. They are not rightfully within the ambit of an inventor's patent right.

The doctrinal difficulty in these cases stems from a perhaps overgenerous view of enablement. It is sometimes plausible to argue, on the basis of current understanding of the undue experimentation standard, that the later claimed embodiments were in fact "taught" by the initial specification. In *Gentry Gallery*,[109] for example, one might argue that a skilled furniture designer could easily deduce from the original disclosure that the seatback controls could be moved off the center console as easily as they were moved in the initial design from their conventional location on the armrests. Traditional enablement law thus presents a deficiency: it cannot deal with cases such as this, where a general set of teachings enables a host of embodiments but does not specifically mention or suggest particular variants that later come to light through the efforts of others. In order to guard against claim amendments that effectively misappropriate these others' efforts, courts apply the written description doctrine.[110]

The impetus behind these cases is surely right. Misappropriation by amendment is not to be condoned or encouraged. The filing of a patent application ought not to be a fishing license, enabling an applicant to troll the waters for promising variants painstakingly developed by others. This practice suffers from the same defect as the now-condemned practice of "submarine patents," made famous by the *Lemelson* case.[111] There is simply no place in a self-respecting system for patents that permit this sort of blatant, rent-seeking gamesmanship.

But does that conclusion inevitably lead to the written description requirement as currently conceived and applied? In my opinion, not necessarily. I would argue that there is more suppleness in the fabric of conventional enablement doctrine than the Federal Circuit has so far appreciated. Judges are not condemned, in other words, by the ineluctable limits of the undue experimentation standard to search for an additional requirement. There was no need to identify a subset of the embodiments taught or enabled in a patent specification that could satisfy an additional, more stringent

109. Gentry Gallery, Inc. v. Berkline Corp., 134 F.3d 1473 (Fed. Cir. 1998).

110. *See* Mark A. Lemley & Kimberly A. Moore, *Ending Abuse of Patent Continuations*, 84 B.U. L. REV. 63, 90 (2004) (noting that the Federal Circuit has enhanced the written description requirement to abate abusive practices). It is evident that the real culprit here is the overgenerous—and perhaps even dysfunctional—rules that allow virtually endless prosecution of patent applications. Others have taken aim at these rules, and reform proposals in this area are now common. *See, e.g.*, *id.* at 83–84 (observing the ample opportunity for patent continuation under the rules governing patent applications). At the same time, it is important to retain the "option" feature of patent applications, for reasons explained just below. As a consequence, strict or rigid limits on the filing of patent continuations may not be the best way to solve this problem.

111. *See* Symbol Techs., Inc. v. Lemelson Med., Educ. & Research Found., 422 F.3d 1378, 1385 (Fed. Cir. 2005) (opining that purposely delaying the issuance of a patent for business reasons is an abuse of the patent system).

requirement. Rather than concoct a new requirement, courts could have recalibrated the old one.

As a matter of technical disclosure doctrine, there are a number of paths to this result. One possible solution is that a patent applicant could be estopped from later claiming subject matter that was (1) initially enabled but unclaimed and (2) covered by a claim amendment that includes features copied from a competitor's product or the prior art.[112] This is an eminently fair rule, as it balances the right of a patentee to make amendments against the rights of competitors to introduce new products free of the risk that they will be intentionally ensnared in someone else's pending patent application. It recognizes that both parties in this situation have rights. The applicant has a right to amend his or her claims as a research project unfolds, as more funding becomes available, or in keeping with general technological developments in a field.[113] But this right to amend is limited by the rights of the applicant's competitors, who should be free to introduce product variants without fear that their designs will be misappropriated by amendment on the part of the patentee.[114]

112. Two quick points on this idea: first, how would a fact finder know whether the competitor's product motivated the amendment or whether the competitor and the applicant happened upon the same variant of the claimed invention at around the same time? One easy way would be to presume the amendment was motivated by the competitor's product and put the burden on the applicant to show that it wasn't, through internal R&D memoranda, minutes of project team meetings, or the like. Evidence that the applicant inventive team was "on the trail" of the variation later covered by the amendment would tend to rebut the presumption that the amendment was motivated by the competitor's project. A lack of such evidence would tend to reinforce the presumption. The second point is this: as with prosecution history estoppel, the "estoppel" label here is not precisely accurate as a matter of legal terminology. Classical estoppel prevents a legal actor from changing his or her legal argument when a third party has relied—changed its position for the worse in some way—on the original argument or position. This will not typically be the case in the amendment scenario under discussion. The original claims will often not be publicly available before the applicant amends them. Thus, there can be no reliance, strictly speaking. I am using estoppel here in the same loose sense as those of us in patent law do when we speak of prosecution history estoppel under the doctrine of equivalents. I am indebted to my colleague James Gordley for this insight with respect to prosecution history estoppel.

113. In this sense, an originally filed patent application has two components: subject matter that is disclosed and claimed, and subject matter that is disclosed but not yet claimed. The latter can be seen as a set of options. The doctrinal proposal I am making here can be viewed this way: the applicant may exercise the "option to claim" any time up until a competitor introduces a competitive product that is disclosed but not yet claimed by the applicant. To state the rule in the language of options, the option to claim expires when the competitor introduces its product. It seems to me that this phraseology helps in two ways. First, it highlights the fact that the scope of the patentee's property right is *not limited* by his or her initial claims; the disclosure of unclaimed subject matter creates a valid set of future options. Second, it clarifies the effect of the competitor's product introduction on the patentee's rights. This is a termination event; the option to claim disclosed but unclaimed embodiments expires when one of those embodiments is introduced by a third party.

114. A patent traditionalist might argue that my proposal runs afoul of a longstanding principle of patent law: that enablement is to be judged at the time of filing, rather than by the standards of some later moment in time. *See In re* Hogan, 559 F.2d 595, 606 (C.C.P.A. 1977) (declaring that considering a subsequent improvement to show a lack of enablement in a prior application would "wreak havoc" on the patent system); *see also* Mark A. Lemley, *The Changing Meaning of Patent*

Whatever the doctrinal mechanism chosen, the point is straightforward: enablement ultimately involves some policy judgment. Just as with causation, negligence, and many other legal principles, the doctrine can and should be adjusted to respond to overarching policy goals. Incremental adjustment of this sort is the essence of the common law method, and even though patent law is statutory, it is a statute with a common law feel.[115] Its basic principles have not changed since the first Patent Act in 1790, nor has the goal of promoting innovation. If the Federal Circuit had recognized this, it would not have been necessary to invent a new, amorphous doctrine such as the written description requirement.

B. *Disclosure Doctrine and the Software Industry*

I am advocating a simple and stable disclosure doctrine[116] through the mechanism of a revamped enablement rule true to its liberal but balanced historical roots—a modern disclosure rule that Justice Story might recognize. In the preceding subpart, I took a detour through contemporary disclosure doctrine, with the aim of critiquing it and suggesting a better alternative. In this subpart, I return to discussion of the software industry. Having established earlier that software patents have not harmed the industry in any appreciable way,[117] and that they are very likely here to stay, it is important

---

Claim Terms, 104 MICH. L. REV. 101, 107 (2005) (noting that federal law mandates that the meaning and scope of patent terms should be determined at the date of application). Under my approach, someone might argue that a claim would be enabled at time one and then later unenabled (by virtue of the introduction of a third party product) at time two. I have two answers. First, many doctrines in patent law operate to take away inchoate rights before they fully vest. Section 102(e) has this *nunc pro tunc* effect: it invalidates a patent, call it Patent A, only when an earlier filed application meeting certain requirements later issues as Patent B. *See* 35 U.S.C. § 102(e) (2000 & Supp. IV 2004). If the earlier filed application never issues as Patent B, Patent A remains valid. We are willing to live with this *nunc pro tunc* effect because it seems unfair to divest Patent A of validity until Patent B issues; yet it seems unfair to permit Patent A to persist in a valid state after Patent B issues. Just as Patent A may be said to be "conditionally valid" under § 102(e), I am arguing that unclaimed embodiments ought to be considered "conditionally enabled" until some later divesting act occurs. My second argument is less abstract: consistent with the idea of "purposive construction," we ought to keep in mind that the Patent Act's overall goal is to promote innovation. This includes not only the innovation of patentees, but also (as is sometimes forgotten) innovation by competitors. In our context, fidelity to this overarching principle requires a loosening of our rigid adherence to a binary conception of enablement—an embodiment either is (for all time) enabled or it's not. This formalistic approach must be rejected in service of the larger goal of promoting overall innovation.

115. Consider that the language of § 101 has not changed in any important respect since 1790. *See* 35 U.S.C. § 101. The basic enablement standard has undergone terminological evolution but has not changed in essential outline. Novelty and statutory bars are much the same as they were in the early eighteenth century. And so on. This explains why the Supreme Court often refers to its own earlier cases—many from the nineteenth century—when construing the 1952 Patent Act.

116. Enablement is determined by the fact finder—a jury, in the typical patent case. Because of this, there will inherently be some variation in the application of the doctrine. The stability I am arguing for can nevertheless be obtained, however, through traditional mechanisms of jury control and review.

117. *See supra* subparts II(A) and II(B).

now to promote a discussion of how detailed patent law doctrines ought to apply to the software industry to best promote its growth. As the doctrines that collectively constitute patent scope are among the most important of these detailed rules, it makes sense to begin with one of them: enablement/written description.

1. A Case Study of Software Patent Scope: LizardTech, Inc. v. Earth Resource Mapping, Inc.—A recent written description case, *LizardTech, Inc. v. Earth Resource Mapping, Inc.*,[118] provides an excellent vehicle for a discussion of patent scope in the software industry. The decision in *LizardTech* invites comparison with the modified enablement doctrine described earlier. In addition, the facts of the case offer a fascinating glimpse into business strategies of the parties and the role of patents in carrying them out. This will flesh out some of the points made earlier about the impact of patents on the software industry.

The following paragraphs make these basic points. Software companies now often acquire at least a few patents as a matter of course, for various purposes.[119] They may help attract financing.[120] They may pave the way for new lines of business.[121] They may be strictly defensive, providing only "freedom to operate."[122] In any event, the ability of these companies to attract venture capital demonstrates once again that patents have not fundamentally harmed the software industry. In addition, the *LizardTech* case demonstrates, consistent with recent research, that patents are used differently by different types of firms in the software industry. This should make courts and other policy makers hesitate to formulate software patent policy on the basis of assumptions about patents' impact on a monolithic software industry. Since patents mean different things to different firms, patent policy will affect different firms differently. Finally, since the patent was invalidated in *LizardTech* as claiming too much given what was disclosed, it ought to assuage, at least somewhat, fears that software patents will inevitably be overbroad and therefore deleterious to the industry.

2. Background: The Companies Involved.—

a. LizardTech.—LizardTech is a small software company that develops and sells a number of data compression programs for customers in several industries.[123] Some of their customers must view and analyze large, complex maps in an effort to locate and develop natural resources such as oil

118. 424 F.3d 1336 (Fed. Cir. 2005).

119. Mann, *supra* note 10, at 972.

120. *See id.* at 984 (discussing venture capitalists' evaluations of patent protection for portfolio companies).

121. *See id.* at 985 (noting profits from licensing).

122. *Id.* at 994.

123. LizardTech - Our Business, http://www.lizardtech.com/company/business.php.

and gas.[124] These maps are stored on computers in massive data files, occupying multiple terabytes.[125] Companies that use this sort of data have workers scattered all over the globe; the ability to send these files to each other and quickly access the graphical data makes them much more productive. But to store and transmit these large files, they must first be drastically compressed. LizardTech has developed a number of computer programs that compress data in a way that is both efficient and accurate, i.e., so the data lose little or no detail after they are transmitted and decompressed.[126] The defendant in LizardTech's patent suit, Earth Resource Mapping, sells competing products in the same industry.[127]

LizardTech also sells compression software to the publishing industry. Its most prominent customer in this area is *The New Yorker* magazine, which adopted LizardTech's technology when it wanted to make its entire historical publishing output—eighty years of weekly magazines—available on DVD disks.[128] Marvel Comics, home of superheroes such as Spiderman and Thor, recently used LizardTech technology to make available early issues of many of its most popular comic books.[129] As with the mapping applications, the appeal of the LizardTech software to publishers is that it can compress the relevant files into small enough space to be manageable, yet restore essentially all of the detail to the stored images when they are decompressed for viewing, all while running quickly and using a limited amount of RAM.[130]

LizardTech was founded in 1992 on the basis of research performed at Los Alamos National Laboratory in New Mexico.[131] In 1996, the company moved to Seattle "to raise venture capital."[132] The move seems to have paid off. By February 2000, LizardTech had already received two early rounds of financing; in that month, it received a third round consisting of $15 million,

---

124. *See* LizardTech - Customers, http://www.lizardtech.com/products/geo/customers.php (listing customers in the oil, gas, and utility industries).

125. *See, e.g.*, LizardTech, Case Study: Pinnacle Mapping Technologies, Inc., http://www.lizardtech.com/files/geo/casestudies/Pinnacle_Case_Study.pdf (describing one customer's map data as exceeding six terabytes). A terabyte is one thousand billion bytes, or one thousand gigabytes. Most computer hard drives hold forty to one hundred gigabytes.

126. *E.g.*, LizardTech - Products - GeoExpress - In Depth, http://www.lizardtech.com/products/geo/indepth.php.

127. ER Mapper - Geospatial Imagery Solutions GIS, http://www.ermapper.com.

128. *See* LizardTech, Case Study: The New Yorker, http://www.lizardtech.com/files/doc/casestudies/The_New_Yorker_Case_Study.pdf (discussing LizardTech's business with *The New Yorker* magazine).

129. Press Release, LizardTech, LizardTech Revolutionizes the Way Newspapers and Magazines are Delivered by Introducing the Publishing Industry to Efficient Digital Replicas of Print Editions at the Folio Show in New York (Nov. 15, 2004), *available at* http://www.lizardtech.com/press/news.php?item=11-15-2004.

130. LizardTech, What is Document Express with DjVu, http://www.lizardtech.com/products/doc/whatis.php.

131. Brier Dudley, *Files Slither Down in Size with LizardTech Product*, SEATTLE TIMES, Sept. 19, 2005, at E1.

132. *Id.*

bringing the total to approximately $20 million.[133] This round of financing was intended to "enable LizardTech to develop its wavelet-based imaging language, called MrSID(TM) (Multiresolution Seamless Image Database) for new platforms and applications."[134]

LizardTech received $25 million of venture capital financing, primarily from Mitsubishi Corporation, during the week of November 20, 2000.[135] "Other investors in the round include[d] Oak Investment Partners, Encompass Ventures, SeaPoint Ventures, Digital Partners, Summit Ventures, Kirlan Ventures and Zeron Group."[136] At this point, CEO John Grizz Deal "refused to disclose sales figures," but he previously reported quadrupled revenues totaling "$2 million for fiscal year 1999."[137] The number of employees also grew from 50 to 200.[138] However, the same report said that although LizardTech had raised $45 million in venture capital funding, it had yet to turn a profit.[139] Most recently, LizardTech was acquired by Celartem Technology USA, Inc., the U.S. arm of a Japanese software holding company, for $11.25 million in cash.[140]

On the product side, early industry response to LizardTech's DjVu technology seems to have been positive, as indicated by the tone of a technical discussion Web site from the year 2000.[141] Over the next few years, the company seems to have focused on developing its image compression technology but did not actually begin commercialization until 2004.[142] Then it began licensing its software to other sellers ("value added resellers") of its geospatial imaging technology in an attempt to integrate its software with other imaging and archival software products.[143]

---

133. *Oak Investment Partners Leads $15 Million Investment in LizardTech*, INTERNETNEWS.COM, Feb. 2, 2000, http://www.internetnews.com/bus-news/article.php/298351#lizard. The primary investor in February 2000 was a venture capital firm known as Oak Investment Partners, but two others—SeaPoint Ventures and Encompass Ventures—also participated. *Id.* "Encompass Ventures, along with Kirlan Ventures, Summit Ventures and Staenberg Private Capital, led earlier private investments in LizardTech," i.e., the first two rounds, prior to February 2000. *Id.* These earlier rounds totaled roughly $5 million. *See id.*

134. *Id.*

135. John Cook, *LizardTech Taps Mitsubishi Riches*, SEATTLE POST-INTELLIGENCER, Nov. 20, 2000, at E4, *available at* http://seattlepi.nwsource.com/business/liza20.shtml.

136. *Id.*

137. *Id.*

138. *Id.*

139. *Recent Layoffs at Area Technology Companies: LizardTech*, SEATTLE POST-INTELLIGENCER, Apr. 2, 2002, http://seattlepi.nwsource.com/venture/layoff.asp?id=439.

140. *LizardTech and Applanix Acquired*, GIS MONITOR, June 26, 2003, http://www.gismonitor.com/news/newsletter/archive/062603.php.

141. *A New Web Image Format*, SLASHDOT, Nov. 22, 2000, http://slashdot.org/articles/00/11/21/2312220.shtml.

142. *See* LizardTech - Press Room - Press Releases, http://www.lizardtech.com/press/news.php?archive=2004 (listing press releases from 2004 announcing commercialization).

143. *Id.*

In 2004, LizardTech had its strongest fiscal year "for its geospatial line of products," primarily because of a program called GeoExpress. It reported a 34% increase in sales over 2003 (but gave no dollar figure).[144] In 2005, LizardTech launched several new versions of software and had some measure of commercial success in deals with magazine companies and distributors, such as *The New Yorker* and North American Publishing Company.[145] This is still a fairly small company, however; it had $29 million in sales in 2005, a 52% increase in compound annual growth since 2001.[146] As of 2005, the company had 170 employees in Seattle, Portland, New York, San Rafael (Cal.), and England.[147] LizardTech has continued to create new versions of its GeoExpress software, including a new version in August 2006.[148]

b. Earth Resource Mapping.—Earth Resource Mapping Ltd. (ERM) got its start when it received a $1,139,450 grant in 1999 from Australia's Industry Research and Development Board to help finance its Image Web Server product.[149] Whereas LizardTech seems to deal broadly in digital image storage and distribution, ERM seems focused specifically on geoimagery—specialized software used to map and model various aspects of the earth.[150] ERM's goal "has always been to make image processing easier to use as a tool, so that professionals of all skill levels and disciplines can effectively utilize the power of geoprocessing and remote sensing technologies."[151] In addition to its image processing tool, ER Mapper, the company offers "ER Radar for radar and SAR data processing."[152] The company's Web site elaborates:

> ER Mapper is used by professionals in a wide range of industries including oil and gas, mining, forestry, defense, agriculture, environmental, state and local government, and telecommunications.

144. Press Release, LizardTech, LizardTech Reports Strongest Fiscal Year for its Geospatial Solutions in 12-Year Company History (Aug. 3, 2004), *available at* http://www.lizardtech.com/press/news.php?item=08-03-2004.

145. LizardTech - Press Room - Press Releases, http://www.lizardtech.com/press/news.php?archive=2005.

146. Dudley, *supra* note 131, at E1.

147. *Id.*

148. LizardTech - Press Room Homepage, http://www.lizardtech.com/press/.

149. AusIndustry, Perth Company's R&D Start Project a Big Hit With Image Processing (July 28, 1999), http://www.ausindustry.gov.au/content/content.cfm?ObjectID=CF0D93E3-36B6-4678-8164B6F174505356&L3Keyword=benefit.

150. *See* GISCafe.com, Earth Resource Mapping - Corporate Listing, http://ecat.giscafe.com/corpprofile.php?vendor_id=3000512 (explaining that ERM's software products are designed to meet the "needs of today's earth scientists"). GISCafe is a professional association for companies in the Geographic Information Systems market and lists both Earth Resource Mapping and LizardTech as members. *Id.*

151. EoPortal, Earth Resource Mapping Spain S.L. Presentation, http://directory.eoportal.org/pres_EarthResourceMappingSpainSL.html.

152. *Id.*

> Anyone managing the earth's natural resources or the urban infrastructure has an application.
>
> ER Mapper is available only from resellers who market to vertical industry types and geographic regions on a non-exclusive basis. These resellers are experts in the application of remote sensing to their industry and provide local support and training.[153]

An ERM press release following the demise of a third company, Mapping Science—which resulted from litigation by LizardTech—gives some sense of the state of the market at the time.[154] According to ERM CEO Stuart Nixon, ERM intended to continue a push for open standards in GIS (geographic information science) imagery, a stance that Mapping Science had taken as well.[155] ERM substantiated this claim by supplying a range of licenses for their ECW JPEG 2000 SDK, two of which were free. A company press release described: "Only strictly commercial ventures who want unlimited compression in commercial products have to pay a once-off and royalty free payment for the SDK, and when they do they get the complete source code."[156] The press release also stated that pricing schemes differed drastically between the companies in that LizardTech charged a per megabyte price for image compression and management, while Mapping Science and ERM did not.[157]

ERM reported 95% growth in 1994.[158] In 2006 it added customers such as Shell Exploration and Production, The Sidwell Company, Terralink International, and several universities.[159] There are no current revenue figures available for the company, but it seems to be a small yet viable player in a specialized niche software market.

3. File Formats, Business Strategy, and Patents: A Quick Case Study.—LizardTech's business strategy parallels that of Adobe, Inc., sponsor of the popular personal document format (PDF) file format. Adobe gives away, at no cost, copies of its basic Acrobat document viewing software.[160] The company does this to "seed" the market for its profitable document design software, which is optimized to work well with PDF format files. Adobe has understood the value of selling software that works seamlessly

153. *Id.*

154. Press Release, ER Mapper Geospatial Imagery Solutions, ER Mapper Reaffirms Commitment to JPEG 2000 and Offers Discount to Mapping Science Licensees (Feb. 16, 2004), *available at* http://www.ermapper.com/NewsView.aspx?t=11&n=83.

155. *Id.*

156. *Id.*

157. *Id.*

158. GISCafe.com, Earth Resource Mapping - Corporate Listing, http://ecat.giscafe.com/corp profile. php?vendor_id=3000512.

159. *See* ER Mapper - News, http://www.ermapper.com/NewsView.aspx?t=11&ny=2006 (listing press releases announcing new customers).

160. *See* Adobe - Products and Services, http://www.adobe.com/products/ (offering cost-free downloads of Adobe viewing software).

with files stored in a popular format. In economic terms, it seeks to take advantage of and profit from the "network externalities" effect: the fact that as more people use PDF documents, PDF compatible software becomes more valuable.[161]

Like Adobe, LizardTech sponsors a document storage file format that in its basic version is free to all comers.[162] Its format, designed to display scanned documents more efficiently and with less degradation than the PDF format, is called DjVu ("déjà vu").[163] As with Adobe, LizardTech sells sophisticated software designed to work seamlessly with files stored in the DjVu format.[164] Its business model is based on the same idea as Adobe's: the more files stored in the DjVu format, the more valuable LizardTech's software will be. This file format was developed at AT&T in the late 1990s and licensed and maintained by AT&T until LizardTech took over in the year 2000.[165] (Interestingly, LizardTech took assignment of this AT&T patent in the same month it received a large infusion of venture capital.)

In the graphical image area, LizardTech has a different strategy. It sponsors a proprietary graphics file format called MrSID.[166] But in this area, ERM has introduced file formats that compete with LizardTech's MrSID. The ERM formats, called ECW and JPEG 2000, are touted as open standards: anyone is invited to use them to create and store files.[167]

So, to summarize: the litigation in this case pitted two smallish but viable software companies against each other in a battle over one software market—imaging software. The companies were (and still are) engaged in a larger struggle to gain market share, in which their competing software products are tied to competing file formats. With this business background in mind, we turn now to a consideration of the patent portfolios of the firms.

4. Patent Portfolios of the Two Firms.—The following table shows the patents held by the two firms involved in *LizardTech*. All the patents listed

---

161. *See, e.g.*, CARL SHAPIRO & HAL R. VARIAN, INFORMATION RULES: A STRATEGIC GUIDE TO THE NETWORK ECONOMY 189 (1999) (describing free distribution of Adobe Acrobat, i.e., PDF software, to promote network externalities); *see also* ROBERT M. GRANT, CONTEMPORARY STRATEGY ANALYSIS: CONCEPTS, TECHNIQUES AND APPLICATIONS 351 (2002) (listing Adobe's PDF document format as an example in a chart describing companies that control industry standards).

162. *See* LizardTech - Products - Digital Replicas, http://www.lizardtech.com/products/doc/dr.php (offering free downloads of the basic viewer for LizardTech's file format).

163. *Id.*

164. *See* Reel/Frame 011089/0349 (recorded Nov. 11, 2000), *available at* http://assignments.uspto.gov/assignments/q?db=pat&qt=rf&reel=011089&frame=0349&pat=&pub=&asnr=&asnri=&asne=&asnei=&asns= (assigning U.S. Patent 5,900,953, which covers image compression file formats, from AT&T to LizardTech).

165. *Id.*

166. *See* MrSID Technology Primer (2004), http://www.lizardtech.com/files/geo/techinfo/MrSID_Tech_Primer.pdf (describing LizardTech's MrSID file format).

167. Press Release, ER Mapper, *supra* note 154.

under "LizardTech" were eventually assigned to LizardTech; but as the table shows, a number were first assigned to other entities.

LizardTech

Patent/app. no.	Inventor(s)	Original assignee	Date filed	Date issued
5,130,701	James M. White, Vance Faber, Jeffrey S. Saltzman (Los Alamos)	U.S. Dept. of Energy → U.C. Regents	05/12/89	07/14/92
Description	Digital color representation			
Notes	This patent has undergone numerous assignments, mergers, and security agreements.			
5,467,110	James M. White, Vance Faber, Jeffrey S. Saltzman (Los Alamos)	U.S. Dept. of Energy → U.C. Regents	09/28/90	11/14/95
Description	Population attribute compression			
Notes	This patent has undergone numerous assignments, mergers, and security agreements.			
5,710,835	Jonathan Bradley (Los Alamos)	U.C. Regents	11/14/95	01/20/98
Description	Storage and retrieval of large digital images			
5,900,953	Leon Bottou, Yann Andre Lecun	AT&T	06/17/97	05/04/99
Description	Method and apparatus for extracting a foreground image and a background image from a color document image			
App. no. 20020124035	Vance Faber, Randall L. Dougherty	LizardTech, Inc.	12/03/01	–
Description	Method for lossless encoding of image data by approximating linear transforms and preserving selected properties for image processing			

Earth Resource Mapping

Patent/app. no.	Inventor(s)	Original assignee	Date filed	Date issued
6,201,897	Stuart Nixon (San Diego)	Earth Resource Mapping, Inc. (San Diego)	11/09/98	03/13/01
Description	Transformation and selective inverse transformation of large digital images			
6,442,298	Stuart Nixon (San Diego)	Earth Resource Mapping, Ltd. (West Perth, Austl.)	10/26/00	08/27/02
Description	Transformation and selective inverse transformation of large digital images			
Notes	This is a continuation of the '897 patent. Together, these cover the allegedly infringing technology at issue in the LizardTech suit. They are not at issue in the case, however; ERM did not counterclaim.			

6,633,688	Stuart Nixon, Simon Cope (Marangaroo, Austl.), Mark Sheridan (Kewdale, Austl.)	Earth Resource Mapping, Inc. (San Diego)	04/28/00	10/14/03
Description	Method system and apparatus for providing image data in client/server systems			

5. *Patents and Business Developments.*—LizardTech's February 2000 venture capital infusion occurred two years after the issuance of the '835 patent it later deployed in litigation against ERM. The general pattern of patent issuance and later round venture financing is consistent with the recent aggregate empirical research of Professor Ronald Mann and Professor Thomas Sager.[168]

There is some direct evidence that the patents involved in this case facilitated financing. Security interests in several LizardTech patents were recorded by Silicon Valley Bank in 1999, and a release from this security interest was recorded later in 2003.[169] LizardTech's patents served as solid assets that were used to secure a bank loan. In this case, at least, company patents converted the firm's tacit, or soft, knowledge into a firm, bankable asset. The deployment of patents in this way is an oft-overlooked advantage of the advent of patents in the software industry.[170] It is not that software

168. *See* Ronald J. Mann & Thomas W. Sager, *Patents, Venture Capital, and Software Start-ups*, 36 RES. POL'Y 193, 202, 201–02 (2007) (finding that each additional patent held by a firm relates to an additional 0.144 rounds of financing, and that "the effects of patents are plainest in differentiating between firms that move on to the expansion stage and those that do not").

169. The assigned interest was in U.S. Patents 5,130,701; 5,467,110; and 5,710,835 (the subject of the dispute in the *LizardTech* litigation). *See* Reel/Frame 009958/0719 (recorded May 21, 1999), *available at* http://assignments.uspto.gov/assignments/q?db=pat&qt=rf&reel=009958&frame=0719&pat=&pub=&asnr=&asnri=&asne=&asnei=&asns= (assigning the above patents from LizardTech to Silicon Valley Bank as a security agreement). The security interest on these three patents was later released. *See* Reel/Frame 014409/0528 (recorded Aug. 22, 2003), *available at* http://assignments.uspto.gov/assignments/q?db=pat&qt=rf&reel=014409&frame=0528&pat=&pub=&asnr=&asnri=&asne=&asnei=&asns= (assigning the above patents from Silicon Valley Bank to LizardTech as a release).

170. An aside: it might be thought that one of the dangers of the recent Supreme Court decision in *eBay Inc. v. MercExchange, L.L.C.*, 126 S. Ct. 1837 (2006), is that the less certain availability of injunctions post-*eBay* will make it more difficult for companies that obtain patents in bankruptcy proceedings, foreclosures, and the like to obtain or threaten large damage awards—therefore damaging the "market for patents" and, indirectly, the willingness of an entity such as Silicon Valley Bank to make a loan using patents as security. This concern is overblown. Although injunction-based settlements may not be available to all patent holders in all circumstances post-*eBay*, that ruling was very far from prohibiting injunctions in all cases where patents are obtained in bankruptcy proceedings and the like. That depends on whether the patent holder can convince the court in question that an injunction is consistent with the overall requirements of equity. More importantly, perhaps, *eBay* says nothing about the ability to collect damages for past infringement, and perhaps ongoing infringement in the rare case where an injunction is denied. While some have argued that the "market for patents" must be curtailed severely through various measures, see, for example, Amy L. Landers, *Liquid Patents*, 84 DENV. U. L. REV. 199, 239 (2006), I would not go nearly so far. The discounting effect of *eBay* ought to naturally cool this market to the extent it was overheated by expectations of a (rent-seeking) windfall in the patent litigation game.

firms never received investment capital in the past, only that patents help to codify and propertize, as it were, the intangible knowledge of software firms. This may be one of the reasons that patents are associated with firm success in the form of later round venture capital financings.

6. The Court's Decision in LizardTech.—The *LizardTech* court invalidated claim 21 in LizardTech's '835 patent, which it had asserted against ERM.[171] The claim, according to the court, failed to comply with the written description requirement because although a specific algorithm was recited in the patent specification, the asserted claim had been broadened (by dropping a limiting feature present in the algorithm described in the specification).[172] To see why, it is important to understand LizardTech's algorithm and how it differed from the one used by ERM.

a. The Technology: Data Compression Algorithms.—Computers represent graphic images—such as maps—as long strings of numbers.[173]

---

171. Claim 21 of U.S. Patent 5,710,835 reads as follows:

> A method for selectively viewing areas of an image at multiple resolutions in a computer having a primary memory for data processing and a secondary memory for data storage, the method comprising the steps of:
>
> storing a complete set of image data array I(x,y) representing said image in a first secondary memory of said computer;
>
> defining a plurality of discrete tile image data T.sub.ij (x,y) subsets, where said complete set of image data I(x,y) is formed by superposition of said discrete tile image data T.sub.ij (x,y);
>
> performing one or more discrete wavelet transformation (DWT)-based compression processes on each said tile image data T.sub.ij (x,y) in a selected sequence to output each said discrete tile image data T.sub.ij (x,y) as a succession of DWT coefficients in a succession of subband sets, where one subband of each set is a low-resolution representation of said discrete tile image data T.sub.ij (x,y) to form a sequence of low-resolution representations of said image data array I(x,y) to selected resolutions;
>
> selecting a viewing set of said image data array I(x,y) to be viewed at a desired resolution:
>
> determining a viewing subset of said DWT wavelet coefficients that support said viewing set of said image data at said desired resolution; and
>
> forming from said subset of said DWT wavelet coefficients a computer display of said viewing set of said image data at said desired resolution.

U.S. Patent No. 5,710,835 cols.13–14 (filed Nov. 14, 1995). Claim 1 reads as follows, in part:

> maintaining updated sums of said DWT coefficients from said discrete tile image T.sub.ij(x,y) to form a seamless DWT of said image and storing said sums in a first primary memory location of said computer;
>
> periodically compressing said sums and transferring said compressed sums to a second secondary memory to maintain sufficient memory in said primary memory for data processing, wherein said second secondary memory contains stored DWT wavelet coefficients . . . .

Id. at col.11.

172. LizardTech, Inc. v. Earth Res. Mapping, Inc., 424 F.3d 1336, 1346 (Fed. Cir. 2005).

173. *See* DAVID SALOMON, DATA COMPRESSION: THE COMPLETE REFERENCE 253 (3d ed. 2004) (stating that with respect to graphic images it is common for computers to have a pixel "represented internally as a 24-bit number").

Each number represents a distinct value, for example, the color of an individual computer screen picture element, or pixel.[174] One way to transmit an image is to simply assign a number to each pixel location and put it in a large table.[175] Because there are so many pixels on a screen, this can quickly use up the available memory in a computer. The problem is exacerbated considerably when one wants to store a huge image—say, a detailed, high-resolution map of an entire county or state. Very few standard computers could hold a single file that stored a sample bitmap of all this data. Even when a computer can store such an image, it is prohibitively difficult to load the image into RAM and manipulate it.

To address this problem, software engineers use various types of compression algorithms.[176] These are mathematical operations that take raw data (such as pixel colors) as their input and produce as their output a compressed version of the data.[177] There are a number of ways to do this mathematically. One is to take all pixel color data that is very close to zero (e.g., close to white in color) and simply set it equal to zero. Then, all the nonzero data can be stored in a table, along with information about the total number of pixels. When this table is unpacked—decompressed—any value in the table that has no data associated with it can be treated as a zero. In this fashion, a great deal of storage space is saved. In effect, every zero value gets stored at very low "cost" in terms of space in the table. The decompressed image will not be a perfect, exact copy of the original, but it will be close enough.

This is a very simple example of data compression. LizardTech's algorithm was much more sophisticated. It was based on the idea of a mathematical transform.[178] Transforms work by repeatedly performing a mathematical operation on a string of data, converting it into a format that can be stored in a smaller amount of space. For example, in a very short string of only two numbers, one can take their average and then record the difference between them. (These numbers, the result of the mathematical transform operation, are referred to as coefficients.) These two coefficients store exactly the same information as the original two numbers but in a different way. The original data can be regenerated from the coefficients by in

174. *See id.* at 25 (describing digital images as consisting of arrays of small dots called pixels).

175. *See id.* (describing pixels as consisting of one or more numerical "bits" that indicate a particular color and explaining the storage of pixels in a table called a "bitmap," which is the input stream for an image).

176. *See generally* KHALID SAYOOD, INTRODUCTION TO DATA COMPRESSION 3–5 (3d ed. 2006) (describing the basic types and uses of compression algorithms).

177. *See id.* at 1 (describing the use of data compression algorithms "to reduce the number of bits required to represent an image or a video sequence or music").

178. *See* LizardTech, Inc. v. Earth Res. Mapping, Inc., 424 F.3d 1336, 1337 (Fed. Cir. 2005) (describing LizardTech's technology as using "wavelet transforms," which "allow digital images to be greatly compressed with very little loss of information").

effect performing the inverse of the mathematical operation used in the compression transform.

If the transform is chosen cleverly and applied to a large data set, it can separate the important data that most affects how the image appears from the unimportant data that has only a marginal effect on the appearance of the image.[179] Then, the program can discard the unimportant data (because, as discussed above, the values are close to zero) and store the important data, resulting in a large decrease in file size. Of course, the reverse transform will not restore the precise image that was originally stored, but any visual difference will be negligible.

Sophisticated techniques start from this simple concept. Increased compression can be achieved a number of ways. One common technique is to perform an operation—such as the "take the average and compute the difference" operation just described—and then perform it again on the first result.[180] That is, in the parlance of the field, perform the operation *recursively*. This can result in a very significant degree of compression. Of course, each time the operation is performed, a new set of coefficients results. And to regenerate the initial numbers, the process must be recursively undone.

Transforms can employ another mathematical concept called a filter.[181] Filters are themselves strings of numbers in a specific format—technically, arrays—that perform operations on a series of numbers in a data string that is being compressed.[182] A filter can be "moved down" a string of numbers, performing an operation on every second, third, fourth, etc., number in the string. For example, the operation might be to multiply each sampled value in the string by a certain number, and then multiply the neighboring numbers by other numbers in the filter array. This will produce a smaller array that in effect samples various points in the data to be compressed, but also captures mathematically a rough sense of the value of neighboring numbers.

The specific transforms that LizardTech and ERM use in their algorithms are called discrete wavelet transforms, or DWTs.[183] DWT works by splitting data using two filters.[184] The low-pass filter uses filtering values and techniques that retain the low-frequency data, i.e., rough data about large

179. *See id.* (describing the ability of wavelet transforms to "transform image data into a form in which it is easier to determine what information in the data is relevant, so that irrelevant and redundant data can be filtered out").

180. *See* SALOMON, *supra* note 173, at 531–32 (describing how a transform can be used to compress an image using an approach of taking the average and computing the difference).

181. *See id.* at 556 (comparing image transforms with subband transforms, which use filters).

182. *See LizardTech*, 424 F.3d at 1337–38 (describing filters as values that are applied to image pixels to derive coefficients). *See generally* SALOMON, *supra* note 173, at 549 ("A filter is a linear operator defined in terms of its filter coefficients.").

183. *LizardTech*, 424 F.3d at 1337.

184. *Id.*

groupings of values.[185] In a large image file, this might represent broad areas of color. The high-pass filter retains the high-frequency data, i.e., high-variation data with a lot of relatively large differences between adjacent values.[186] In image data, this might represent data about the edges of map features or textures such as different elevations.

All of the data in a map file necessarily fall into one of these categories, so nothing is lost in the splitting process. First, the filters are each run over the array in the row direction to create two new arrays of high-band and low-band coefficients.[187] Because this effectively doubles the amount of data, creating two new arrays from the original array, half of this data can be deleted with no loss.[188] This is called downsampling.[189] After downsampling, both filters are run over each of the arrays calculated in the previous step, but in the column direction.[190] This creates a total of four subbands: low-low, low-high, high-low, and high-high subbands. The arrays are downsampled once again to keep the total amount of data equal to that in the original image.[191] At this point, no data has been lost. Rather, the image has been transformed into a different encoding.

The data compression in DWT comes when this reconfigured array is recursively transformed, and data with values close to zero are eliminated.

One nice property of DWT is the ability to display the image in lower resolution. In order to do this, the algorithm can display one of the low-low decompositions instead of reassembling the image. While some of the details stored in the other decompositions will not appear, the broader contours of the image will be visible.[192] Lower and lower resolutions can be displayed by showing the low-low decomposition of successive DWT iterations.[193]

185. *See id.* (noting that the low-pass filter retains the low-frequency information and filters out the high-frequency information).

186. *Id.*

187. *Id.* at 1338.

188. *Id.*

189. *Id.*

190. *Id.*

191. *Id.*

192. This also provides a good metaphor for understanding conceptually what DWT is doing. In effect, an iteration of DWT separates the data of an image into three categories. First, the broad contours, the "most important" data, are stored in the low-low decomposition. Second, the fine-grained details, the "less important" data, are stored in the other three decompositions. Third, the finest grained details, the "least important" data, which consists of the numbers very close to zero, are deleted for the purpose of compression.

193. This is an extremely useful technique, the best example of which is the program Google Earth. Google Earth, http://earth.google.com/. I'm not certain what transform they use, but the idea is the same. When the user wants to see a map, Google only needs to send the coefficients for the low-low decomposition at that resolution for that area rather than sending the data for the entire map, thus yielding a large increase in speed and a large decrease in bandwidth. Moreover, "zooming in" merely requires sending another set of coefficients. Another benefit is that the client computer, rather than the server computer, does the calculation to reassemble the image.

A major problem with DWT is that it requires storing the entire array in computer memory (such as RAM) at one time,[194] which is prohibitively difficult for large images. The obvious solution to this problem is to split the image into "tiles" and calculate the DWT on each of those tiles. However, this creates a new problem. Because there is no access to the rest of the image when calculating DWT on a tile, the values outside the tile are set to zero.[195] That is, the filter can only incorporate the data in a given tile into its calculations because it does not "see" the entire row or column as it would if the whole image were present. This creates edge artifacts throughout the image—false representations of the data caused when the array of coefficients is decompressed.[196]

LizardTech's patented method deals with this problem by using the fact that DWT is linear to split the calculation of the DWT coefficients.[197] The method first splits the image into tiles, as above.[198] Then it calculates the DWT for these tiles starting from one corner, progressing in diagonal stripes.[199] This part of the process is taught by the prior art.[200]

The trick is what it does in addition to this. The prior art centers the filter on each element in the tile.[201] The LizardTech method also centers the filter on elements *outside the tile* that have been set to zero.[202] This is a means of calculating not only the coefficients for the tile loaded into memory, but also the effect of that tile on later calculated coefficients. In effect, the sums are broken up and calculated in pieces. This results in a seamless transform that is exactly the same as if DWT had been run on the entire, unbroken image.[203]

The basic DWT calculation, centering the filter on a pixel and calculating the sum of products, is a linear operation. It is the sum of products. Therefore, it does not matter whether values in that sum are added in the same order or instead calculated partially at first with the remainder completed later. The resulting coefficient will be the same as long as all parts of the sum are included at some point in the algorithm. Thus, the LizardTech method is effective because it breaks up the sums but still manages to include all of the parts of the sum.

---

194. *LizardTech*, 424 F.3d at 1339.
195. *Id.*
196. *Id.*
197. *Id.*
198. *Id.*
199. *Id.* at 1338–39.
200. *Id.* at 1337–39.
201. *Id.* at 1337.
202. *Id.* at 1339.
203. *Id.*

b. The Court's Decision.—LizardTech claimed a number of embodiments of its data compression process.[204] Claim 1 is both representative of and important in the court's decision. It reads:

> A method for selectively viewing areas of an image at multiple resolutions in a computer . . . comprising the steps of:
>
> storing a complete set of image data array I(x,y) representing said image . . . ;
>
> defining a plurality of discrete tile image data $T_{ij}(x,y)$ subsets, where said complete set of image data I(x,y) is formed by superposition of said discrete tile image data $T_{ij}(x,y)$;
>
> performing one or more discrete wavelet transformation (DWT)-based compression processes on each said tile image data $T_{ij}(x,y)$ in a selected sequence to output each said discrete tile image data $T_{ij}(x,y)$ as a succession of DWT coefficients . . . ;
>
> *maintaining updated sums* of said DWT coefficients from said discrete tile image $T_{ij}(x,y)$ to form a seamless DWT of said image and storing said sums in a first primary memory location of said computer;
>
> *periodically compressing said sums* and transferring said compressed sums to a second secondary memory . . . ;
>
> selecting a viewing set of said image data array I(x,y) to be viewed at a desired resolution . . . .[205]

The italicized terms figure prominently in the '835 patent's specification. They represent the incremental, "creeping" calculation of coefficient values as arrays for individual "tiles" are transformed. They are central to LizardTech's approach, as they represent the key step differentiating LizardTech's algorithm from prior art DWT techniques.

Without doubt, the accused infringer, ERM, did not infringe claim 1. ERM had a completely different technique for using DWT to transform data; the ERM technique does not break the data into tiles. Thus it does not need to "maintain[] updated sums" for tiles, or to "periodically compress[] said sums." The ERM technique was described in general terms in a declaration at trial by company founder and chief researcher Stuart Nixon:

> ERM Uses a Continuous Sliding Window Approach: This technique never breaks up or tiles the image, so it does not introduce any edge artifacts. Instead, it relies on the critical observation that contrary to what was previously thought, the DWT process does *not* need to generate the entire intermediate images before generating the output sub-band images. The newly-patented ERM method uses this observation to perform a standard prior art DWT technique, but does

204. U.S. Patent No. 5,710,835 col.11 (filed Nov. 14, 1995).

205. *Id.* (emphasis added).

> so by structuring the data flow to ensure that only the minimum amount of data required is stored in memory at any one time.[206]

But LizardTech had another claim—claim 21—to fall back on in its infringement suit. According to the court, "[c]laim 21 of the '835 patent is identical to claim 1 except that it does not contain the 'maintaining updated sums' and 'periodically compressing said sums' limitations [of that claim]."[207] That is, it eliminated the key phrases that placed ERM's technique outside the bounds of the patent claim. This led the trial court to find the patent invalid under the written description requirement.[208] And it led LizardTech to appeal.

In its decision on appeal, the Federal Circuit focused on the omission of the "maintaining updated sums" limitation as critical:

> The trouble with allowing claim 21 to cover all ways of performing DWT-based compression processes that lead to a seamless DWT is that there is no support for such a broad claim in the specification. The specification provides only a single way of creating a seamless DWT, which is by maintaining updated sums of DWT coefficients. There is no evidence that the specification contemplates a more generic way of creating a seamless array of DWT coefficients.[209]

This seems absolutely correct on the facts. The idea of maintaining updated sums refers to the idea that calculation of the sum is split up. Thus, the LizardTech method must maintain updated sums for each coefficient until all parts of the sum have been calculated. Because LizardTech did not describe all ways to obtain a seamless DWT, this claim could cover a host of methods that are not supported by the specification. The only example in the specification maintains updated sums, and there is no indication of how the algorithm might be performed without doing so.

The court admitted in framing its holding that—consistent with the argument earlier in this Article—enablement might explain this outcome just as well as the written description requirement:

> Those two requirements [enablement and written description] usually rise and fall together. That is, a recitation of how to make and use the invention across the full breadth of the claim is ordinarily sufficient to demonstrate that the inventor possesses the full scope of the invention, and vice versa. This case is no exception. Whether the flaw in the specification is regarded as a failure to demonstrate that the patentee possessed the full scope of the invention recited in claim 21 or a failure to enable the full breadth of that claim, the specification

206. Declaration of Stuart Nixon at ¶ 14, LizardTech, Inc. v. Earth Res. Mapping, Inc., 2000 WL 35453681 (W.D. Wash. Oct. 12, 2000) (No. C99-1602C).

207. *LizardTech*, 424 F.3d at 1343.

208. *Id.* at 1346–47.

209. *Id.* at 1344.

provides inadequate support for the claim under section 112, paragraph one.[210]

The question recurs: why two requirements, instead of one? This aside, the decision seems quite apt. It also represents an answer, in part, to many of the early fears about software patents.

7. Reprise: The Role of Patents in Competition Between LizardTech and ERM.—The opinion in *LizardTech* does not state why LizardTech brought suit against ERM. What is clear from the business press and the other available evidence is that the firms compete vigorously in the market for geospatial data handling software.[211] It appears that the patent suit was just another front in a multidimensional battle for market share and corporate survival—business as usual, as it were.

The very unremarkable nature of the patent infringement suit in this case shows how routine patents and all their trappings (including the occasional infringement suit) have become in the software industry. Quite contrary to all those predictions in earlier years, patents are evidently not strangling either of these companies or the industry sector in which they operate. Patents may have played a role in helping one or both of these firms attract investment capital. (The record on ERM is sketchy in this regard, but the evidence presented earlier for LizardTech is suggestive and perhaps convincing.)[212] Patents are surely playing a role in the age-old battle over "shelf space" in this competitive industry. But, contrary to predictions early and late, neither company is asserting its patents willy-nilly throughout the industry; there is no evidence that patents represent a massive transaction-cost burden on the industry. None of the evidence demonstrates a huge burden of licensing in the creation of the LizardTech and ERM products, a major fear in the early days of software patents. (Indeed, the *open* licensing of patents, to build support for a standard, belies the "transaction costs will choke the industry" argument altogether.) And neither firm is a behemoth either. Both are on the small side, by U.S. corporate standards. This belies

---

210. *Id.* at 1345.

211. *See, e.g.*, Press Release, ER Mapper, ER Mapper's ECW JPEG2000 SDK Advances Open Standards, *available at* http://www.eijournal.com/cat_content.asp?contentid=1059&catid=173 (detailing the geospatial data handling software offered by ER Mapper); Press Release, ER Mapper, Release of ER Mapper's ECW JPEG 2000 SDK Advances Open Standards and Uptake of JPEG 2000 (Mar. 17, 2004), *available at* http://www.directionsmag.com/press.releases/index.php?duty=Show&id=8916 (same); Joe Francica, *LizardTech GeoExpress 6—Product Overview*, DIRECTIONS MAGAZINE, Feb. 6, 2006, http://www.directionsmag.com/article.php?article_id=2086&trv=1 (detailing the geospatial data handling software offered by LizardTech); Joe Francica, *Product Review: GeoExpress 5.0 with MrSID by LizardTech*, DIRECTIONS MAGAZINE, Feb. 1, 2005, http://www.directionsmag.com/product_reviews.php?feature_id=122 (same); Susan Smith, *Dr. David Maguire on the ArcGIS 9.0 Product Family Release*, GIS WEEKLY, May 24–28, 2004, http://www10.giscafe.com/nbc/articles/view_weekly.php?articleid=208790&page_no=4 (detailing the geospatial data handling software offered by ER Mapper).

212. *See supra* notes 131–40 and accompanying text.

early predictions that the "patent overhead"—the costs of acquiring and administering patents—would drive small firms out of the software industry. Whatever else patents may have done, they have not shut down these two small, innovative companies. Nor, to judge by concentration statistics and other data, have they done so in other sectors of the software industry.

a. Patents and "Open Standards."—One interesting feature of the two firms' strategies is the widespread licensing of competing data compression protocols or file formats: DjVu in the case of LizardTech and ECW JPEG 2000 for ERM. As mentioned, the strategy here is dictated by the realities of network externality-driven businesses. Each company reasons, on the basis of examples like Adobe's PDF format, that making its compressed file format a standard will help it make more money in the long term.[213]

The interesting point to note is that both formats appear to be covered by patents. What this means is that both firms are choosing to *give away* to many people free copies of patented software. This is surely not what the early critics of software patents predicted. And indeed, many contemporary devotees of free, open-source software are quite wary of software patents as well.[214] This is due in part to fears that software patents will clog the arteries of commerce and innovation.[215] Critics point out that the software field grew up, after all, without the specter of patents, and that the higher transaction costs that presumably accompany the advent of patents for software inventions can only cause harm.[216]

This may yet come to pass. But the fact remains that many customers are receiving free copies of these firms' software, despite the fact that they are patented. This business strategy suggests that misguided patents may not automatically produce the dire consequences that open-source advocates fear. The decision to patent, for LizardTech and ERM at least, is separate and distinct from the question of whether to adopt an open-software-licensing

213. There is both a positive (PDF/DjVu) and negative (JPEG 2000) side to this strategy depending on whether a single firm entirely controls a format. Ironically, when a single firm controls a format, it has a significant incentive to give an irrevocable open license to encourage wide support. However, when a format is purportedly open from the beginning, firms have significant incentives to claim some stake in it, however tenuous. This effect seems particularly strong for firms that (1) view patents as assets and (2) may actually develop technology but are not very profitable (e.g., the firm is trying to recoup otherwise lost venture capital). Recognition of this effect might allow courts to better weigh damages and injunctions in the wake of *eBay Inc. v. MercExchange, L.L.C.*, 126 S. Ct. 1837 (2006).

214. *See, e.g.*, Grant C. Yang, *The Continuing Debate of Software Patents and the Open Source Movement*, 13 TEX. INTELL. PROP. L.J. 171, 191 (2005) (listing a litany of criticisms of software patents made by the open-source community).

215. *See* David S. Evans & Anne Layne-Farrar, *Software Patents and Open Source: The Battle Over Intellectual Property Rights*, 9 VA. J.L. & TECH. 10, ¶¶ 44–68 (2004), http://www.vjolt.net/vol9/issue3/v9i3_a10-Evans.pdf (recounting and responding to the theory that software patents will stifle innovation).

216. *See id.* at ¶¶ 7–14, 44–68 (chronicling the development of the pure software patent).

strategy. The automatic pairing of "patented" with "closed/proprietary" licensing or dissemination strategies does not apply here.

A careful look around reveals that these firms are not unique in this respect. Obtaining patents represents a strategic choice quite distinct from whether to make a technology available to some or all users. There are many reasons why a firm might patent software and still license freely to many users. Much of the thinking is driven by the dynamics of network industries. Patents may be held in reserve, to be deployed (if at all) against direct competitors, while being essentially waived via open licensing to other users.

The table below tries to capture the difference between open-licensed patent strategies and closed-licensed patent strategies. Most of the examples will be familiar. The fourth quadrant, with the example of the EnCase forensic disk analysis standard, may not be. EnCase is a software technique for making a precise copy of a hard drive whose contents are to be studied for forensic purposes (i.e., to obtain evidence of the disk's contents in a legal proceeding).[217] The creator of the standard, Guidance Software, uses the EnCase protocol in its own products, but does not license it to others.[218] This file format has not been patented (as yet, anyway);[219] the company seems to maintain it as a trade secret. It thus serves as an example of an unpatented technology whose owner has chosen a closed licensing strategy.

Table 3: Strategy Grid

	Open	Closed/proprietary
Patents	Adobe Acrobat; LizardTech DjVu format	Apple iTunes music format; LizardTech MrSID graphics format
No patents	Open-source software, e.g., Linux Operating System	EnCase Forensic Disc Analysis software[220]

The point of the table is simple: patents can and often do coexist with open licensing strategies. Thus, there is no direct link between obtaining

217. Guidance Software, EnCase Forensic, http://www.guidancesoftware.com/products/ef_index.asp.

218. Guidance Software's Web site does not mention licensing opportunities and compares its product favorably to that of its competitors. *Cf.* Guidance Software, EnCase Forensic Corporate, http://www.guidancesoftware.com/downloads/getpdf.aspx?file=/downloads/95-00-01029_encase_forensic_corp_and_dlx_v505.pdf ("No other computer forensic solution has this track record of credibility established by its users, independent organizations and courts.").

219. Guidance does have one patent, but it is unrelated. Enterprise Computer Investigation System, U.S. Patent No. 6,792,545 (filed June 20, 2002).

220. Guidance Software, EnCase Forensic, http://guidancesoftware.com/products/ef_index.asp.

software patents and locking up technology against all comers. Indeed, a single firm can both obtain and enforce patents (against some), and freely license those patents (to others). It might even be said that the choice to selectively waive the property right is one of the key advantages of obtaining patents. In addition, as the LizardTech example shows, a single firm might opt in one technology area for an open-access strategy and in another area for a closed/proprietary strategy—even though it has patents in both areas.

b. Patents and "Interoperability."—In recent years, one objection to patents in the software industry is that they may hinder software company efforts to design products that easily plug into each other. This is the problem of patents and "interoperability."[221] Although *LizardTech* is only one case, it is instructive for the interoperability debate for two reasons. First, as mentioned, both firms here held numerous patents. Yet this did not prevent them in any way from giving their products away, strategically licensing their technology, and, in general, doing everything possible to establish their file formats as a standard. Patents did not prove incompatible with the firms' interest in promoting interoperability.[222] Secondly, perhaps the key point is that LizardTech's attempt to claim its compression algorithm broadly was defeated by the Federal Circuit. This is a point often overlooked by those who fear that patents will hinder interoperability. To do so, a patent has to be valid. Only an *innovative* interface, protocol, or file format—one that can pass the requirements of patentability—will create this potential problem. Admittedly, not all issued patents are really valid (though patent quality reform remains a real goal today). Nevertheless, though software companies would like to patent all interface points with their software, or with popular software with which their products interact, in many cases they will not be able to. In other words, the desire to control interoperability via patents will not always translate into the ability to do so.[223] Remember the big lesson here: LizardTech lost.

---

221. *See, e.g.*, Yang, *supra* note 214, at 195 (noting that obvious software patents could threaten interoperability in the software industry by forcing companies "to migrate to a completely different standard"); Aaron C. Chatterjee, *Europe Struggles Over Software Patents*, IEEE SPECTRUM, Sept. 2004 at 61, 62 (recognizing the concern that patents that cover standards and protocols could hinder the interoperability of different computing systems).

222. Admittedly, the failure of the firms to actually establish successful standards could conceivably have been caused in part by the fact that potential customers and adopters were reluctant to adopt a file format that was covered by a patent. Of course, this has not been the case with Adobe's PDF standard, but it is conceivably a potential problem in other cases. In this regard, a slight doctrinal wrinkle may someday be necessary. Firms holding patents on part or all of a technology that becomes widely adopted as a standard could use a "bait and switch" strategy, hiding or withholding enforcement of key patents until after the standard is firmly entrenched and later bringing lawsuits against any and all adopters. The obvious solution is using a doctrine of "standards estoppel" to take care of the problem. *See* Robert P. Merges & Jeffrey Kuhn, The Reliance Interest in Standards (Apr. 2007) (unpublished manuscript, on file with author).

223. It is an interesting question whether a patent that is valid and that covers an interoperability point might be handled differently by a court that is convinced that full enforcement

IV. Conclusion

Entry and competition are robust in the software industry. Firms are obtaining patents to assist in financing and to use as strategic weapons in ongoing battles over market share. Courts are limiting software patents in ways calibrated to adjust the value of the property right to the quantum of technical contribution represented in the patent's specification.

Many features of the system can no doubt be improved. Two that come to mind immediately are patent trolls (entities that acquire patents purely for litigation and that perform no ongoing research and development activity) and poor-quality patents. Both these related issues are relevant to the software industry. One major problem that trolls exploit occurs in industries where a single patent may cover one component of a highly complex product containing perhaps hundreds of components. The availability (or simply the threat) of an injunction in such a situation can give a patentee highly disproportionate leverage over an accused infringer. Software is such an industry. The *eBay* case mentioned earlier gives courts a good weapon to prevent the worst effects of patent trolls on the industry, though this depends on the lower courts applying it wisely. The second problem, low patent quality, also significantly affects software companies. Current institutional arrangements do not make it easy for companies to challenge the validity of "mistake" patents, though proposed reforms such as postgrant patent invalidation proceedings may help.[224] For the time being, however, and possibly even if such proceedings become possible, low-quality patents may continue to pose problems for the industry.

Thus the patent situation in the software industry is surely not the best of all possible worlds. But as I have sketched it in this Article, the overall picture does not look anything like a fiasco to me. Given the early predictions, that is a reassuring thought.

---

of the patent would bestow disproportional leverage on a patentee. I have speculated in the past about the enforcement of property rights in this sort of situation. *See* Robert P. Merges, *Who Owns the Charles River Bridge? Intellectual Property and Competition in the Software Industry* (Univ. of Cal. Berkeley Sch. of Law, Public Law and Legal Theory Working Paper No. 15, 1999), *available at* http://ssrn.com/abstract=208089. Actually, this is only one aspect of a broader conceptual issue in intellectual property law. *See* Robert P. Merges, The Proportionality Principle in Intellectual Property Law (Apr. 2007) (unpublished manuscript, on file with author).

224. *See* Joseph Farrell & Robert P. Merges, *Incentives to Challenge and Defend Patents: Why Litigation Won't Reliably Fix Patent Office Errors and Why Administrative Patent Review Might Help*, 19 BERKELEY TECH. L.J. 943 (2004) (discussing potential reforms).