Notre Dame Law Review

Volume 69
Issue 5 *Notre Dame Law School 125th Anniversary Issue*

Article 11

March 2014

Software Patent Protection: Debugging the Current System

Thomas P. Burke

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

Recommended Citation
Thomas P. Burke, *Software Patent Protection: Debugging the Current System*, 69 Notre Dame L. Rev. 1115 (1994).
Available at: http://scholarship.law.nd.edu/ndlr/vol69/iss5/11

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

NOTES

Software Patent Protection: Debugging the Current System

Immature artists imitate; mature artists steal.
Lionel Trilling[1]

The process of creating intellectual property requires a considerable expenditure of time, resources, and, most importantly, the exercise of the human creative spirit. A computer program demonstrates more than scientific skill. Well-written software reveals a simplicity and elegance characteristic of artistic expression.[2] To protect and encourage society's investment in these creations, various forms of legal protection have been crafted.

Thievery is not practiced only by the *artiste.* Last year, publishers of personal computer (PC) business software lost $7.4 billion to software[3] piracy—an improvement from $9.7 billion in 1992.[4] Straight copying of computer programs is generally prohibited by the Copyright Act, which protects the expression of original works

---

1 *Quoted in* RICHARD KENNEDY, COLLECTED STORIES 23 (1987). Steve Jobs, co-founder of Apple Computer, uses a similar expression which he attributes to Pablo Picasso, "When I was at Apple we had a saying that we kept on the wall. It said, 'Good Artists Copy, Great Artists Steal.'" Daniel J. Lyons, *Color by numbers,* PC WK., Apr. 5, 1988, at 117; Jacki Lyden, *The Macintosh Computer Celebrates Its 10th Anniversary,* NPR, ALL THINGS CONSIDERED, Jan. 22, 1994, *available in* LEXIS, News library, Script file.

2 ANTHONY LAWRENCE CLAPES, SOFTWARS: THE LEGAL BATTLES FOR CONTROL OF THE GLOBAL SOFTWARE INDUSTRY 10-13 (1993) ("Like a novel, and unlike say, an automobile, the physical medium in which that creativity is delivered to customers is insignificant. The intellectual content, not the package, is what is important and valuable about a computer program. Moreover, the intellectual content, not the package, is what costs money to create.").

3 An illustrated guide explaining software is provided in RON WHITE, HOW SOFTWARE WORKS 43-65 (1993).

4 These estimates, from the Software Publishers Association (SPA), underestimate the extent of the problem as they consider only piracy of packaged business applications software of SPA members (which represents a small fraction of the total amount spent on software development and sales); excluded are sales from lines such as operating systems, entertainment, scientific, and educational software. *Software publishers,* USA TODAY, Mar. 30, 1994, at 10B. US/Canadian sales of software in this SPA category in 1993 amounted to $6.3 billion while U.S./Canadian sales of all packaged software is estimated to have been $33.4 billion. U.S. DEP'T OF COMMERCE, U.S. INDUSTRIAL OUTLOOK 1994 27-5.

of authorship.[5] Protection for new ideas reduced to practice in software-related inventions is covered by the Patent Act.[6] Whether computer software should be protected by patents has always been controversial.[7] In recent months, three events have returned the issue of patent protection to the forefront.

First, the award of a patent to Compton's NewMedia Inc.[8] by the Patent and Trademark Office (PTO) for CD-ROM interactive search and retrieval technology brought howls from the emerging multimedia industry. The breadth of the patent's claims suggested the possibility that all existing and future multimedia products could be found infringing.[9] Compton issued a press release stating that it would seek royalties of one to three percent of revenues on infringing uses. The avalanche of public protest stirred PTO Commissioner Bruce Lehman *sua sponte* to launch a re-examination of the patent—a review which has tentatively led to the dismissal of all forty-one claims in the original patent.[10]

---

5 *See, e.g.*, Peter H. Lewis, *Student Accused of Running Network for Pirated Software*, N.Y. TIMES, Apr. 9, 1994, at 1. In 1980, the Copyright Act was amended to expressly include software. 17 U.S.C. § 101 (1988) ("A 'computer program' is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."), Pub. L. No. 96-517, § 10(a), 94 Stat. 3028 (1980). *See also* 17 U.S.C. § 117. Limitations on Exclusive Rights: Computer Programs (1988) (essential step and archival use limited exceptions).

6 Patent Act of 1952, Pub. L. No. 82-593, § 112, ¶ 1, 66 Stat. 792 (1952), *codified at* 35 U.S.C. §§ 101-376 (1988).

7 *See infra* notes 17, 34 & 43 and accompanying text; *c.f.* Steven J. Henry, *Software industry must lead fight for intellectual property law reform*, INFOWORLD, Dec. 6, 1993, at 60 (op-ed calling for greater industry-government cooperation given two decades of conflicts).

8 Reed et al., Patent Number 5,241,671. The application for the patent was filed on October 26, 1989 but was not issued until August 31, 1993.

9 Sales of multimedia products amounted to an estimated 10.3 million units in 1993, a increase of 114.2 percent from 1992. It is estimated that from 1993-96 the multimedia market will have a 26.9 percent annual compound growth rate. U.S. INDUSTRIAL OUTLOOK 1994, *supra* note 4, at 27-4.

10 Teresa Riordan, *Action was Preliminary On a Disputed Patent*, N.Y. TIMES, Mar. 30, 1994, at D7. *See also* Karl J. Kramer, *Multimedia Under Review*, LEGAL TIMES, Jan. 24, 1994, at S32 (discusses readily available prior art that would invalidate the patent as well as deficiencies in the patent's disclosure); J. Leigh Hunt Firestone, *Software Patents, in* SOFTWARE DEVELOPMENT: A LEGAL GUIDE appendix (Stephen Fishman ed., 1994) (provides an annotated analysis of the Compton patent).

Reaction to the patent was delayed until November when Compton announced at Comdex/Fall '93 in Las Vegas that it would be seeking royalties. A sampling of the industry responses to the patent include: Microsoft Chairman Bill Gates—the patent is a "joke"; Tomas Lansky of Munich Production Partners—"It's shameful you would even apply for such a patent—shameful, shameful." Michael Fitzgerald & Clair Whitmer, *Competitors cry 'shame'*, COMPUTERWORLD, Nov. 22, 1993, at 28. Multimedia Computing Corp. President Nick Arnett—"They're acting like they invented the multimedia industry, which

Second, eyebrows were raised when a jury awarded $120 million to Stac Electronics in its infringement suit against Microsoft over space-saving data compression technology.[11] Such an award provides a ready response to corporate financial officers who question the costs involved in prosecuting a patent.[12] Besides the money exchanged between the companies, the ruling has practical significance to millions of MS-DOS users. Microsoft's "DoubleSpace" has been stripped from new shipments of the oper-

---

is ludicrous." James Daly, *Patent pyrotechnics: CD-ROM publisher claims rights to advanced multimedia retrieval technology*, COMPUTERWORLD, Nov. 22, 1993, at 28. Jeffrey Tarter, Editor and Publisher of Soft Letter—"Clearly they [the PTO] have no understanding of the software industry, and they fail consistently to do their homework. There have been patents before that have sounded similarly apocalyptic that turned out to be essentially meaningless." Peter H. Lewis, *The New Patent That is Infuriating the Multimedia Industry*, N.Y. TIMES, Nov. 28, 1993, at 10. Fireworks President Robert Carberry—"Patenting multimedia is like patenting the English language."; Media Vision Executive Vice-President Min Yee—"I don't think it's that significant . . . . Still, I think developers are going to jump on them like a herd of elephants on a mambo snake." Clair Whitmer, *Compton's multimedia patent, licensing plan worry developers*, INFOWORLD, Nov. 29, 1993, at 14.

See also Don Clark, *Patents May Raise Price of Information Highway*, WALL ST. J., Nov. 15, 1993, at B1 (brief discussion of other efforts to enforce software patents).

11 Ed Scannell & Stuart J. Johnston, *Patents may be key to survival*, COMPUTERWORLD, Mar. 7, 1994, at 30 (Gary Clow, Stac's President and CEO, commented afterwards, "I think we have shown a prototype for how small, innovative companies can compete . . . . We have shown the way for securing patents for intellectual property against a very strong competitor.") In the nature of biting the hand that feeds you, Clow was quoted to have commented on the Compton patent, "Compton's was a surprise . . . . That's a case where the patent system has failed us." Steve Hamm, *Patented problems: U.S. Patent and Trademark Office Responding to Criticism*, PC WK., Jan. 24, 1994, at A1.

The jury also returned a verdict in favor of Microsoft for $13 million on a counterclaim that Stac had violated Microsoft's trade secrets by reverse-engineering a pre-load feature. Reverse-engineering undocumented hooks and calls in DOS is a long standing practice dating back at least to Borland's "SideKick," a memory resident program popular in the early 1980's. *Cf.* Andrew Schulman, *LA Law*, DR. DOBB'S JOURNAL, May 1994, at 137 (critical of jury award). Trade secret protection is a matter of state laws and is beyond the scope of this note.

12 Patent prosecution is a term of art used to describe the progress of an application through the PTO through allowance and issuance or rejection and appeal. Patent infringement litigation describes a lawsuit, originating in a U.S. district court, between private parties, where at least one party accuses the other of violating the owner's rights in a validly issued patent. *See* DONALD S. CHISUM, 3 PATENTS § 11.01 (1993).

The fee schedule for acquiring and maintaining a patent can be found at 37 C.F.R. 1.16 to .28 (which are reduced by fifty percent for individuals or companies qualifying as small entities). For a straight-forward patent, the fees, drawings, and other expenses can be expected to cost around $5,000. Due to a variety of historical and ideological factors discussed in this Note, the cost of obtaining a software patent can quickly reach over $20,000.

ating system, and Stac's "Stacker" could lose an important feature.[13]

Third, PTO Commissioner Lehman held a series of public hearings early this year in San Jose, California and Crystal City, Virginia on the use and appropriate standards for software-related inventions under the patent laws.[14] The major players in the debate, ranging from IBM (strongly pro-patent) to The League of Programming Freedom (adamantly opposed to any patent protection), participated in these well-publicized hearings.[15]

This Note examines the issue of software patent protection and makes several recommendations for reforming the existing system. Part I provides an overview of the competitive framework in the computer industry. Part II sets out the argument for patent

---

13 Doug Barney, *Microsoft replacing DoubleSpace; IBM acquires Stacker compression*, INFOWORLD, Apr. 11, 1994, at 10; Ed Scannell & Stuart J. Johnston, *Ruling means DOS to lose compression*, COMPUTERWORLD, Feb. 28, 1994, at 4.

Microsoft also had a large patent-related expense in 1992. As part of the Microsoft/IBM divorce battle over PC operating systems, Microsoft paid IBM a one-time fee estimated to be $20-30 million to license IBM's portfolio of more than 1,00 patents relating to software. *IBM Deal to Cost Microsoft Millions*, SAN JOSE MERCURY NEWS, June 29, 1992, at 9D.

14 PTO, DEP'T OF COMMERCE, NOTICE OF PUBLIC HEARINGS AND REQUEST FOR COMMENTS ON PATENT PROTECTION FOR SOFTWARE-RELATED INVENTIONS (December 14, 1993). *See also* Victoria Slind-Flor, *New Patent Chief Reinvents His Job*, NAT'L L.J., Feb. 28, 1994, at 1; John Markoff, *A High-Technology Outcry Against the U.S. Patent System*, N.Y. TIMES, Jan. 3, 1994, at C16.

A prior effort at reform was undertaken in 1990 when the Secretary of Commerce established a commission to explore changes in all areas of patent law. The commission issued an invitation for public comment and received approximately 545 responses—over 400 of them discussed computer-related inventions. THE ADVISORY COMMISSION ON PATENT LAW REFORM, A REPORT TO THE SECRETARY OF COMMERCE 418 (1992), *portions reprinted in* 1 DAVID BENDER, COMPUTER LAW § 3A[7] (1993) (section of report on computer-related inventions).

15 Clair Whitmer, *Industry divided over software patents: Patent and Trademark Office weighs protecting vs. stifling innovation*, INFOWORLD, Feb. 28, 1994, at 20 ("As competition increases from overseas, patents will be the most important tool to protect original software innovation in the U.S. . . . . [Without patents] we'd be unilaterally disarmed relative to our competitors in Europe and Japan." Victor Siber, IBM Senior Counsel). In contrast, The League distributes a button depicting a snake consuming the Apple Computer logo with the legend, "Keep your lawyers off my computer." Victoria Slind-Flor, *Lawyers, Programmers Interface*, NAT'L L.J., Mar. 16, 1992, at 3. *See also* Gregory Aharonian, *Review of USPTO Hearings in San Jose on Software Patenting*, INTERNET PATENT NEW SERVICE, Jan. 29, 1994 (details position taken by each speaker who testified at hearings—electronic newsletter published on the Internet originating from patents@world.std.com); Gregory Aharonian, *Review of USPTO Hearings in Washington on Software Patenting*, INTERNET PATENT NEW SERVICE, Feb. 12, 1994 (Crystal City version of previous); Sabra Chartrand, *Ideas, advice and criticism spring forth on how, and whether, to grant patents involving software*, N.Y. TIMES, Feb. 14, 1994, at D2; Willem Knibbe, *Software's heavy hitters take shots at patent process*, INFOWORLD, Jan. 31, 1994, at 8.

protection and the requirements for the issuance of a valid software patent. Parts III and IV briefly discuss the already extensively reviewed cases on the "algorithm" test for determining statutory subject matter, a test which increasingly serves only to catch the unwary practitioner. Part V provides some recommendations for debugging the system.[16]

The public good is best served by a system that offers: (a) predictable and meaningful rewards to inventors as incentives for genuine advances in the state of the art; and (b) the open dispersal of knowledge that will serve as a foundation for the next generation of breakthroughs. This nation's long history of adapting the patent laws to new industries, along with some industry cooperation, can make this possible. "Nuts"[17] to the naysayers.

I. THE ECONOMIC STRUCTURE AND IMPORTANCE OF THE SOFTWARE INDUSTRY

A. Information Technology Economics

World-wide sales of packaged software were an estimated $71.8 billion in 1993 with U.S. producers controlling seventy-five percent of the global market.[18] While sales in the U.S. market were $32

16 Admiral Grace M. Hopper coined the term "bug" after she removed a moth that had stopped an early Mark II computer at Harvard in the 1940's. To debug is to fix a problem. Linda Runyan, *The Datamation Hall of Fame*, DATAMATION, Sept. 15, 1987, at 56.

17 Gen. Anthony McAuliffe, Replying to the German demand for surrender at Bastogne, Belgium during the Battle of the Bulge on Dec. 22, 1944, *quoted in To Nazi Surrender Plea: "Nuts!"*, N.Y. TIMES, Dec. 28, 1944, at 4. *Contra* Ed Foster, *With patents all the rage, why not one for a pile of bricks?*, INFOWORLD, Dec. 27, 1993, at 45; Michael J. Miller, *Software Patents Must Go*, PC MAG., Mar. 15, 1994, at 79 (Two non-lawyers wishing for WILLIAM SHAKESPEARE, THE SECOND PART OF KING HENRY THE SIXTH act 4, sc. 2.); *compare* Carl Dichter, *Patently wrong? software patents*, UNIX REVIEW, Nov. 1992, at 38; Jim Seymour, *The market, not the judiciary, rules*, PC WEEK, Mar. 28, 1994, at 87.

18 U.S. INDUSTRIAL OUTLOOK 1994, *supra* note 4, at 27-4, 5. *See also* MICHAEL E. PORTER, THE COMPETITIVE ADVANTAGE OF NATIONS 252-56 (1990) (Tie between manufactured goods and services strongest early in evolution of goods involved. Service providers tend to maintain market share where product is complex or technical connection between the goods and services is strong); Richard Brandt, *Industry Outlook: Software will Play Hardball Again*, BUSINESS WEEK, Jan. 10. 1994, at 82 (While media attention is focused on Microsoft's close to 50 percent share of PC software sales with a projected $4.9 billion in 1994 revenues, it should be kept in mind that this represents only a fraction of the money spent on software.); Ralph Oman, *Report from the Copyright Office to the ABA Annual Meeting*, PTC NEWSLETTER, Fall 1991, at 22 (Intellectual property now accounts for more than 25 percent of U.S. exports.); *The Harm of Patents*, THE ECONOMIST, Aug. 22, 1992, at 56 (In 1990, the U.S. had a positive balance of $12.6 billion in international patent and license transactions while Japan and Germany had negative balances, respectively, of $3.5 and $1.9 billion.).

billion, U.S. producers sold $47.6 billion worth of product.[19] Projections through 1997 place the industry on track for a 12.8% annual compounded growth rate. Since 1987, employment in this industry has grown at an annual rate of 6.6%. In fact, the industry now employs about 4% of the American work force.[20]

In comparison, the U.S. computer equipment (hardware) industry is expected to run a trade deficit of $17 billion in 1994 with imports accounting for sixty percent of the U.S. market.[21] Hardware product shipments of U.S. based-manufacturers in 1993 were an estimated $56.3 billion.[22] For the last few years, hardware manufacturers have restructured because of (a) the shift from mainframes and midrange systems to less expensive workstations and micro-computers; (b) the shift to open systems from more profitable proprietary systems; (c) the increased commodization of many products; and (d) the greater competition from foreign producers, particularly those in Asia.[23] Employment in the hardware industry fell seven percent in 1993. This represents the fifth straight year of decline and a thirty-three percent drop from 1988. The software industry now has more than double the number of employees as the hardware industry.[24]

As large as reported sales figures are for the software industry, they underestimate the amount of investment in and the importance of quality software to U.S. competitiveness. Software is a driver technology that propels technological advances in other industries.[25] The following items highlight what is often overlooked in the numbers: corporations spend billions of dollars purchasing information services, automating internal operations, proto-typing new products[26] and providing new customer servic-

19 U.S. INDUSTRIAL OUTLOOK 1994, *supra* note 4, at 27-4, 5.

20 NOTICE OF PUBLIC HEARINGS, *supra* note 14, at 3.

21 U.S. INDUSTRIAL OUTLOOK 1994, *supra* note 4, at 26-1, 7.

22 *Id.*

23 *Id.* at 26-3 to 26-26. In 1993, world-wide sales of computer equipment were estimated to be $120.7 billion with PCs accounting for $66.3 billion, mainframes $21.2 billion, midrange systems $21 billion, and workstations and supercomputers accounting for the rest. John C. Dvorak, *Inside Track*, PC WK., Apr. 25, 1994, at 95 (figures from the market research firm Dataquest).

24 U.S. INDUSTRIAL OUTLOOK 1994, *supra* note 4, at 26-1, 2 & 27-1.

25 OFFICE OF TECHNOLOGY ASSESSMENT, FINDING A BALANCE: COMPUTER SOFTWARE, INTELLECTUAL PROPERTY & THE CHALLENGE OF TECHNOLOGICAL CHANGE 14-15 (1992); *see also* ADVISORY COMMISSION, *supra* note 14, at 145.

26 Barnaby J. Feder, *Sophisticated Software Set for Exotic Financial Trades*, N.Y. TIMES, Mar. 30, 1994, at D1.

es;[27] microprocessor sales are governed by compatibility with the microcode of the industry's leading chipmakers;[28] the U.S. machine tools industry was on its death-bed until innovative easy-to-use software revitalized sales in the last few years;[29] television's high-tech future is controlled by the joint ventures among PC software companies, hardware vendors, Hollywood, and other U.S. based television production companies;[30] worldwide sales of software-dependent microcontrollers were an estimated 1.7 billion units in 1993.[31] To take one example, Bell Labs channels eighty percent of development expenditures for new phone switches into software development.[32]

B. Architectural Life Cycle of a Market: Small Players are Winnowed Out

Even with the size and degree of integration of software into the total U.S. industrial process, some argue that software, especially PC software, is a special case. The dominant mind-set of the special case advocates is based on the experience of the microcomputer industry in the 1980s when hundreds of small start-up companies with new ideas flourished, creating thousands of multi-millionaires.[33] Adept combinations of existing technologies and

---

27 In the early days of the computer industry, hardware costs were the constraint in budgeting systems, and human capital was relatively cheap. No longer. With the application of Moore's Law (the number of transistors on a computer chip doubles roughly ever two years), hardware has become a relatively cheap up-front expense. Bill Machrone, *Moore's Law vs. Machrone's Law*, PC WK., Nov. 24, 1992, at 87. When including software development and maintenance costs, software amounts to around 90 percent of the total life-time costs of an information system. Software expenses are almost all labor-related when you count the associated rent and utility bills to provide office space for the programmers working on the computer code. Barry W. Boehm, SOFTWARE ENGINEERING ECONOMICS 18 (1981), *cited in* OFFICE OF TECHNOLOGY ASSESSMENT, COMPUTER SOFTWARE & INTELLECTUAL PROPERTY: BACKGROUND PAPER 1 (1990).

28 Robert D. Hof, *The Party's Not Over Yet: Growth may slow to 14%, but chipmakers' profits will keep climbing*, BUSINESS WEEK, Jan. 10, 1994, at 83.

29 U.S. INDUSTRIAL OUTLOOK 1994, *supra* note 4, at 16-1 to 9. Although still second to Japan and Germany, a few U.S. companies are world leaders in computer-numerically-controlled (CNC) machine tools.

30 Steve Ditlea, *Digital Compression: Squeezing Video to Fit*, HEMISPHERES, Jan. 1994, at 105.

31 Robert E. Calem, *In Far More Gadgets, a Hidden Chip*, N.Y. TIMES, Jan. 2, 1994, at 9.

32 George Gilder, *A Major Work in the Making into the Fibersphere*, FORBES, Dec. 7, 1992, at 111.

33 *See* ROBERT LEVERING ET AL., THE COMPUTER ENTREPRENEURS: WHO'S MAKING IT BIG AND HOW IN AMERICA'S UPSTART INDUSTRY (1984).

small incremental advances in the state of the art frequently provided the base for new fortunes. The argument is made that patent protection should therefore be withheld in the hope that the heady entrepreneurial days of the past can be sustained into the future.[34]

This argument's premise is a static view of industrial structure. Recent mergers and other consolidations demonstrate that the market has matured along predictable lines. This maturation has occurred apart from any patent protection intellectual property regime. Industry analysts Ferguson and Morris have adapted the product life cycle theory to describe the competitive environment for new technologies. The model substitutes the "product" in life cycles theories with a broader "architectural" emphasis to explain how market segments invariably standardize into product designs which are compatible with the segment leader. The stages are:

I. **Early-Stage Chaos**—Technology opens a new market; early entrants probe for space.
II. **Diffusion-Race**—Architectural contenders appear, fight for coverage through alliances, licensing, and early products supporting competing standards.
III. **A Winner Emerges**—The market chooses a de facto standard which accumulates supporting infrastructure and third-party add-on products which supplement the leader's product.
IV. **Harvesting and Extension**—The winner expands and enhances its architectural coverage, favoring its own and allies' products. Proprietary control tightens.
V. **Obsolescence and Migration**—As the architecture ages, the

34 *See Software Patents: Into the Breach Again*, DR. DOBB'S DEVELOPER UPDATE, Apr. 1994, at 1 (provides an account of the various positions taken at the PTO hearings); *see also* Simon L. Garfinkel, et al., *Why Patents are Bad for Software*, ISSUES IN SCIENCE & TECH., Fall 1991, at 50; Richard Stallman, et al., *Viewpoint: Against Software Patents; software should be copyrighted, not patented*, COMMUNICATIONS OF THE ACM, Jan. 1992, at 17, 121; Pamela Samuelson, et al., *Developments on the intellectual property front*, COMMUNICATIONS OF THE ACM, June 1992, at 33,; *contra* Paul Heckel, *The Software-Patent Controversy*, COMPUTER LAW., Dec. 1992, at 13.

> sponsor develops new layers to facilitate migration, giving it a jump-start on a new diffusion cycle.[35]

In mainstream markets, entrepreneurial companies either grow, merge, or die. The pattern has become familiar in the computer industry. IBM buried the "BUNCH" with its 360/370 series of mainframe computers.[36] Data General and Prime stumbled in the minicomputer market when they encountered fierce competition from IBM and DEC.[37] The market for mainframe software is now dominated by Computer Associates and IBM.

Strong patent protection might alter the distribution of profits around the boardrooms of Silicon Valley, but it will not stop the consolidation of industry *sales* into the hands of a few key players. Strong patent protection would, however, continue to encourage innovation in software *design* by small players. In the sectors that have already consolidated, the standard practice has been for companies to broadly cross-license patent libraries for nominal amounts.[38] The winnowing out process with respect to sales is now occurring in the market for PC application programs. These programs are increasingly sold in all-in-one software "suites," encompassing word processors, spreadsheets, and other applications.[39] Assuming that announced mergers survive antitrust review, Microsoft, Lotus, and Novell will control 89% of the word process-

---

35 In deciding to compete or co-exist with an established architectural leader, a company can: a) seek out a niche in a specialty market, b) clone the leader's architecture, c) steal the standard by licensing technology and attaching proprietary extensions, d) establish a franchise in a adjacent market inhabiting a different competitive space, e) acquiesce to the standard and sell products within the framework provided by leader, or f) radically innovate to leapfrog and supplant the architecture. CHARLES H. FERGUSON & CHARLES R. MORRIS, COMPUTER WARS: HOW THE WEST CAN WIN IN A POST-IBM WORLD 146, 161 (1993). *See also* JOHN A. CZEPIEL, COMPETITIVE MARKETING STRATEGY 175-253 (1992) (provides a framework for incorporating technology into product strategy decisions and life cycle management).

36 Burroughs, UNIVAC, NCR, Control Data, and RCA/Honeywell. *See* Paul Carroll, BIG BLUES: THE UNMAKING OF IBM 3-4 (1993); *see generally* Parker Hodges, *Three decades by the numbers; graphic survey of data processing growth*, DATAMATION, Sept. 15, 1987, at 77 (Datamation's thirtieth anniversary issue with a selection of articles and illustrations providing an overview to the history of computing).

37 *See generally* TRACY KIDDER, THE SOUL OF A NEW MACHINE (1981) (Pulitzer Prize winning story of Data General's attempt to radically innovate and supplant competitors as the architectural leader in minicomputers.).

38 FERGUSON & MORRIS, *supra* note 35, at 88, 188.

39 Peggy Wallace, *IS software purchasing in the age of the suite*, INFOWORLD, Mar. 28, 1994, at 62.

ing market and 97% of the spreadsheet market for PCs.[40] In 1993, these three companies had $1.65 billion in software sales revenue for word processor programs and $1.33 billion in spreadsheet sales.[41] With industry sales and marketing power becoming highly concentrated, individuals and companies outside of the big three lack any incentive to innovate without some form of intellectual property protection.

On the general state of the software industry, one industry commentator has compared the situation to the automobile industry:

> In 1920, there were about 300 full-line American automobile makers. By 1930, there were 25. By 1940 there were 10. Today there are three, if you ignore the Hummer. The same thing is happening in the software business, only faster. We took 70 years to accelerate from zero to 100 million cars, but we torqued to 100 million PCs in less than 20 years. So expect the software shakeout to take five years, tops, starting a year ago.[42]

It is in this environment of consolidation that the role for patent protection must be examined.[43] For example, a small inventor would lack the financial incentive to develop an airbag system if the large automakers could freely copy the invention. Similarly, many categories of packaged software are dominated by a small number of products with extensive marketing, user, and third party support. High switching costs make it unlikely that a new entrant could displace significant sales from the segment's market-share leaders in the absence of an underlying architecture change.

---

40 Lawrence M. Fisher, *Novell to Acquire Wordperfect*, N.Y. TIMES, Mar. 22, 1994, at D1 (Market shares are based on 1992 shipments). *See* G. Pascal Zachary, *Consolidation Sweeps the Software Industry; Small Firms Imperiled*, WALL ST. J., Mar. 23, 1994, at A1.

41 Lynda Radosevich & Elisabeth Horwitt, *Desktop leaders face off*, COMPUTERWORLD, Mar. 28, 1994, at 1.

42 Robert X. Cringely, *If Novell is Ford and Lotus is Chrysler, does that make Borland Hudson?*, INFOWORLD, Mar. 28, 1994, at 98. The pace of consolidation has picked up. Recent mergers include (along with their announcement dates): Symantec/Central Point (4/4/94), Novell/WordPerfect (3/21/94), Adobe Systems/Aldus (3/17/94), Electronic Arts/Broderbund (2/9/94), Novell/AT&T Unix Systems (12/21/92), and Borland/Ashton-Tate (7/10/91). Derek Slater, *Industry Almanac: Mega deals*, COMPUTERWORLD, Apr. 4, 1994, at 101.

43 *See The Academic Debate: considered opinion and advocacy; the question whether software can and should be patented*, COMMUNICATIONS OF THE ACM, June 1992, at 125 ("The industry has flourished and innovated without patents. A more accurate statement would be: 'The industry has flourished and innovated with little realization in the software community that patent protection was available.'").

Patent protection may be the only means by which small companies can enter and effectively compete in mature markets.

II. WHY HAVE PATENTS? WHAT INVENTIONS ARE ELIGIBLE?

A. Background

The software industry has been historically distrustful of big-company bureaucracy. The legendary start-ups were founded by mavericks with a passion for writing great computer code. Fostered by the ethos of the open sharing of ideas (and especially source code[44]) at our top universities, thousands of maverick start-ups have been founded over the years in the hope of upstaging the competition and writing the next "killer-app."[45] From this research oriented point of view, the legal system would appear to impede progress by awarding inventors property rights for the fruits of their labor, thereby stifling unrestricted development.

In conformity with this view, intellectual property rights in software have been opposed since the early days of the industry. Until the 1980s, IBM made nearly all its profits from hardware

---

44 Before the age of personal computers, the dominant computer environment on college campuses was to have mini-computers running the UNIX operating system written in the C programming language. UNIX and C were written at Bell Labs and made freely available in source code format along with the significant enhancements made at UC Berkeley. *See* FERGUSON & MORRIS, *supra* note 35 at 102-07. The software was buggy, but you couldn't complain about the price. Thousands of programmers, including the author, got their start in the field by wading through, debugging, and contributing to the vast library of UNIX code freely distributed among college campuses and research institutions. Operating environments have proliferated, industries matured, and Novell, which purchased UNIX from AT&T, charges a license fee for UNIX. *See* Karen D. Schwartz, *Novell's USL (Unix Systems Laboratories) buy offers Unix alternative*, GOV'T COMPUTER NEWS, Jan. 4, 1993, at 6.

45 The "killer-app" (killer application) is the Holy Grail of the software industry. Write it, and the world will beat a path to your door. The latest such application is Mosaic, a point-and-click graphical search tool for exploring the Internet. *See, e.g.*, Garry Ray, *Mosaic: The Killer App*, COMPUTERWORLD CLIENT/SERVER J., Feb. 1994, at 72. Mosaic was co-authored by a 22-year old at the federally funded National Center for Supercomputer Applications (NCSA) at the University of Illinois.. Enhanced versions of Mosaic are now being developed for sale by commercial software publishers. Kimberly Patch, *Companies tap Mosaic for Internet services*, PC WK., Mar. 28, 1994, at 53.

What is often overlooked is that universities have been heavily dependent on substantial government subsidies from such sources as the Department of Defense's Defense Advanced Research Projects Agency (DARPA) and NASA. A common pattern has been that computer scientists conduct their pioneering work at universities and then proceed to attract the venture capital to start their own companies once they have a product which is commercially viable. *See, e.g.*, OFFICE OF TECHNOLOGY ASSESSMENT, AFTER THE COLD WAR: LIVING WITH LOWER DEFENSE SPENDING 103-26 (1992); Fumio Kodama, *Technology Fusion and the New R&D*, HARV. BUS. REV., July/Aug. 1992, at 70.

sales. A year after the invention of the first personal computer in 1975, now-billionaire Bill Gates, co-founder of Microsoft, was widely criticized for publicly complaining of "hobbyists who he said were 'ripping off' his software."[46]

In the mobile job market, new programming techniques were quickly spread by employees switching positions. The lax patent protection for software was the result of a series of historical anomalies. Software initially represented only about eight percent of the value of the earliest systems.[47] Much as car dealers used to throw in floor mats and a full tank of gas, hardware vendors would include the software to make their systems operational.[48] By the 1960s, software had risen to forty percent of delivered value.[49] Yet manufacturers of computer systems chose to tie in operating and other utility software as part of a package deal leaving software without a distinct identity in the minds of consumers.[50] The remaining business applications software was typically custom written in-house. For the U.S. and foreign software competition that did spring up, IBM and AT&T were required to license their patent libraries inexpensively under 1956 consent decrees with the U.S. Department of Justice.[51]

The standard practice was for industry competitors to cross-license their patent libraries thereby allowing for the near royalty-free use of the co-signatories' inventions. Cross-licensing did not have a large impact in the domestic competitive environment as IBM and AT&T's chief competitors also heavily invested in research and development. These pacts became an efficient way to

---

46 JAMES WALLACE & JIM ERICKSON, HARD DRIVE: BILL GATES AND THE MAKING OF THE MICROSOFT EMPIRE 105 (1992). *See* PAUL FREIBERGER & MICHAEL SWAINE, FIRE IN THE VALLEY: THE MAKING OF THE PERSONAL COMPUTER 169 (1984) (Text of Bill Gates' Feb. 3, 1976 letter in the Homebrew Computer Club newsletter).

47 FERGUSON & MORRIS, *supra* note 35, at 7.

48 *See* Karen Hooten, *Is copyright right? the author's right to original expression*, COMPUTER LANGUAGE, Mar. 1991, at 99 ("Traditionally (at a time when tradition meant what IBM did), distribution of software, including source code, was free to hardware owners. Software of every conceivable type was available—much of it bad or buggy, but since you had the source, you could fix it yourself.").

49 FERGUSON & MORRIS, *supra* note 35, at 7.

50 THOMAS J. WATSON, JR. & PETER PETRE, FATHER SON & CO.: MY LIFE AT IBM AND BEYOND 380-82 (1990); *see generally* Lawrence T. Festa, III, Comment, Eastman Kodak Co. v. Image Technical Services, Inc.: *The Decline and Fall of the Chicago Empire?*, 68 NOTRE DAME L. REV. 619 (1993).

51 FERGUSON & MORRIS, *supra* note 35, at 88, 238. United States v. IBM, 1956 Trade Cas. (CCH) ¶ 68,245 (S.D.N.Y. 1956); United States v. Western Electric Co., 1956 Trade Cas. (CCH) ¶ 68,246 (D.N.J. 1956). *But see* RICHARD THOMAS DELAMARTER, BIG BLUE: IBM'S USE AND ABUSE OF POWER 23 (1986).

avert extensive licensing negotiations regarding each new invention and avoid costly patent battles.[52] Larger corporations hesitated to seek legal redress against their smaller competitors due to antitrust concerns.[53] Now that the hegemonic positions of IBM, AT&T, and Xerox have eroded, they can no longer subsidize the industry by handing out the fruits of their large research budgets.[54]

Since the emergence of the PC in the 1980s, the industry's newer entrants have been the most prone to use litigation. Apple, Lotus, and Hayes have all used the copyright laws to protect their positions as architectural leaders.[55] There is truth in the saying that, "a rising tide lifts all boats." In an expanding market with rewards dispersed among several fiercely competitive players, it simply does not make economic sense for corporate officers to invest the time or money in expensive litigation. The potential rewards of enforcing one's intellectual property rights through litigation usually only become compelling at Ferguson and Morris' third stage, "A Winner Emerges," when a product has become a de facto industry standard.[56] By then, it often is too late. The majority of profits to be earned on many high-tech products come during the first few months of sales.[57] Companies have relatively short windows before their products are "leap-frogged" by new technologies or "me-too" products flood the market.[58] Even with competitive market conditions and a relatively hostile legal environment, a study of cases from 1979 to 1991 revealed outcomes

---

52 FERGUSON & MORRIS, *supra* note 35, at 154-57.

53 WATSON & PETRE, *supra* note 50, at 378-82.

54 A 1982 estimate put pure research expenditures (which excludes product development costs) at: IBM $500 million, Bellcore (the research arm of the seven regional Bell operating companies) $150 million, and Xerox $120 million. Elizabeth Corcoran, *Redesigning research*, SCIENTIFIC AMERICAN, June 1992, at 6. In comparison, IBM's 1982 R&D budget (combining pure research and product development) was $7 billion. CAROLL, *supra* note 36, at 340.

55 *See infra* note 62.

56 FERGUSON & MORRIS, *supra* note 35, at 161; *see also* John P. Summer and Steven W. Lundberg, *Software Patents: Are They Here to Stay?*, COMPUTER LAW., Oct. 1991, at 8 (pragmatic bargaining reasons for building a strong patent portfolio).

57 *See also* CZEPIEL, *supra* note 35, at 258-72 (empirical evidence of the rewards to successful pioneers).

58 *See* Joseph Kattan, *Antitrust analysis of joint ventures: allocative efficiency and the rewards of innovation*, ANTITRUST L.J., Mar. 22, 1993, at 37.

favorable to the patentee in software infringement suits by more than a two-to-one margin.[59]

In the area of software protection, it has been remarked that "the 1970s was the decade of trade secret litigation. The 1980s was the decade of copyright litigation. The 1990s is shaping up to be the decade of patent litigation."[60] In 1980, the Copyright Act was amended[61] to explicitly include software. A series of cases followed that fleshed out the scope of protection.[62] The battlelines now are drawn over patent protection. The first widely recognized pure software patent was not granted until 1982. That patent covered the recalculation of values in a spreadsheet.[63] Where to draw the line on patents is the subject of the debate raging in such forums as PTO hearings,[64] traditional printed publications,[65] and on-line discussion groups.[66] So, why have software patents?

---

59 The study uncovered 31 reported and unreported district court infringement cases on software patents. Of the 17 cases resolved, twelve were in favor of the patentee and 5 against. Michael J. Lennon, *A Statistical Analysis of the Enforcement of United States Patents Relating to Computer Software*, 8 COMPUTER L. ASS'N BULL. No. 2 p. 3 (1992).

60 David Bender, *The More Things Change, The More they Stay the Same: An Unhurried Reflection on Software Protection Over the Years*, 16 RUTGERS COMPUTER & TECH. L.J. 309, 321 (1990).

61 *Supra* note 5.

62 *See* Morgan Chu & Andre Brunel, *Post-Altai Computer Copyright and Trade Secret Decisions*, COMPUTER LAW., Jan. 1994, at 1; David L. Hayes, *What's Left of 'Look and Feel': A Current Analysis*, COMPUTER LAW., May 1993, at 1 (Part I) & June 1993, at 1 (Part II); Trevor W. Nagel, *Software Development: The Limits of Existing Legal Protection*, 9 HARV. INT'L REV., Feb./Mar. 1987, at 46 (limitations of international copyright protection).

63 In re Pardo, 684 F.2d 912 (CCPA 1982). The patent was awarded after twelve years in patent prosecution. CLAPES, *supra* note 2, at 112-15. Previously there had been hundreds of "software-related" patents issued. These older patents were written as apparatus claims limited to running on specified devices. *See* Dichter, *supra* note 17, at 38-39; *infra* notes 132-38. After *Diehr*, it is clear that software can be written as an element of a claim. Writing a pure software patent, with only prefunctionary referrals to hardware, can still lead to lengthy delays spent arguing about the proper application of the "mathematical algorithm" test with the PTO. *See infra* Parts III & IV and accompanying text. Terms such as "software" or "computer program" are rarely used in patent applications. Claims drafting language such as a "means for" are the routine. For example, the Compton multimedia patent avoids using "software" in favor of "search system." Fishman, *supra* note 10, at App. p.3.

64 *See supra* notes 14-15.

65 *See supra* notes 17, 34 & 43.

66 For instance, the misc.int-property and misc.legal.computing Usenet newsgroups on the Internet have had running debates on the merits of software patents for the past few months.

B. The Tradeoff: Gaining a Limited Monopoly in Exchange for Public Disclosure

Patents are an exclusively federal form of intellectual property protection found in the Constitution.[67] The owner of a patent enjoys a limited monopoly to control the use of the invention for seventeen years—soon to be increased to twenty years.[68] The public policy aims are to encourage innovation and disperse knowledge.

1. An Incentive to Create

The patent system seeks to spur private sector innovation and promote the dissemination of new technologies by granting a property right to the original inventor.[69] For example, U.S. drug companies spend billions of dollars each year on research and development. Once discovered, tested, approved, and marketed, a new drug can be copied at little cost by other pharmaceutical companies. Absent patent protection to reward investment, the revolutionary advances in pharmacology of the last forty years would not have occurred.[70]

Perhaps counter-intuitively, patents can spur innovation in fields where numerous patents already exist. Companies hate sending royalty checks to their competitors. "The patent system encour-

---

67 U.S. CONST. art. I, §8, cl. 8:

> The Congress shall have Power . . . . To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries; . . . .

68 35 U.S.C. § 154 (1988). As part of the Uruguay Round of the GATT (General Agreement on Tariffs and Trade) trade negotiations, the United States agreed to extend patent protection to twenty years. *See* OFFICE OF U.S. TRADE REPRESENTATIVE, FINAL ACT EMBODYING THE RESULTS OF THE URUGUAY ROUND Part II, Annex 1c: Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPS), § 5, Article 33 ("The term of protection available shall not end before the expiration of a period of twenty years counted from the filing date.").

69 FINDING A BALANCE, *supra* note 25, at 183-99.

70 Using someone else's labor without compensation is called the "free rider" problem in economic literature. There is a growing body of law and economics literature critically examining how large a role patents serve as an incentive to invent. *See generally* CLAPES, *supra* note 2, at 116-21; M. Grady & J. Alexander, *Patent Law and Rent Dissipation*, 78 VA. L. REV. 293 (1992); Robert P. Merges, *Uncertainty and the Standard of Patentability*, 7 HIGH TECH. L.J. 1 (1992); Robert P. Merges, *Commercial Success and Patent Standards: Economic Perspectives on Innovation*, 76 CAL. L. REV. 805 (1988); A. Samuel Oddi, *Beyond Obviousness: Invention Protection in the Twenty-First Century*, 38 AM. U. L. REV. 1097 (1989).

ages invention, not only in that it rewards the inventor with a patent but it spurs the competitors to put forth their mightiest effort to produce a product as good, yet different, from the patent's . . . ."[71] Mirroring the current debate, Thomas Jefferson was initially against patents. He subsequently modified his position, authored the Patent Act of 1793, and directed the patent board for several years.[72]

As the software industry matures, antitrust laws exist to protect consumers and foster free competition by regulating any anti-competitive behavior by the industry's competing firms.[73] It is up to

---

71 James P. Marsh Corp. v. U.S. Gage Co., 129 F.2d 161, 165 (7th. Cir. 1942). In this context, abusive practices such as spurious blocking patents and patent flooding need to be prevented through use of such mechanisms as fraud on the patent office. *See, e.g.*, Albert E. Fey & David C. Planche, *Recent Developments in Inequitable Conduct, in* ELECTRONIC & COMPUTER PATENT LAW, at 559 (PLI Patents, Copyrights, Trademarks, & Literary Prop. Course Handbook Series No. 292, 1990); William J. Gilbreth & William H. Steinmetz, *The Patent Misuse Defense: Its Expansion and Contraction*, J. PROPRIETARY RTS., Aug. 1992, at 7; Donald M. Spero, *Patent Protection or Piracy—A CEO Views Japan*, HARV. BUS. REV., Sept./Oct. 1990, at 58.

72 Act of Feb. 21, 1793, § 1, 1 Stat. 319. As recounted by Justice Tom Clark:

> Jefferson, like other Americans, had an instinctive aversion to monopolies. It was a monopoly on tea that sparked the Revolution and Jefferson certainly did not favor an equivalent form of monopoly under the new government. His abhorrence of monopoly extended initially to patents as well. From France, he wrote to Madison (July 1788) urging a Bill of Rights provision restricting monopoly, as against the argument that limited monopoly might serve to incite "ingenuity," he argued forcefully that "the benefit even of limited monopolies is too doubtful to be opposed to that of their general suppression,"
>
> His views ripened, however, and in another letter to Madison (Aug. 1789) after the drafting of the Bill of Rights, Jefferson stated that he would have been pleased by an express provision in this form:
>
> > Art. 9. Monopolies may be allowed to persons for their own productions in literature & their own inventions in the arts, for a term not exceeding __ years but for no longer term & no other purpose.
>
> And he later wrote:
>
> > Certainly an inventor ought to be allowed a right to the benefit of his invention for some certain time . . . . Nobody wishes more than I do that ingenuity should receive a liberal encouragement.

Graham v. John Deere Co., 383 U.S. 1, 7-8 (1965) (citations omitted).

See also THE FEDERALIST NO. 43 (James Madison) ("The utility of this power will scarcely be questioned. The copyright of authors have been solemnly adjudged in Great Britain to be a right of common law. The right to useful inventions seems with equal reason to belong to the inventors. The public good fully coincides in both cases with the claims of individuals."); Diamond v. Chakbrabarty, 447 U.S. 303, 308-09 (1980) (references Jefferson's writings).

73 United States v. Aluminum Co. of America, 148 F.2d 416, 429-30 (2d Cir. 1945) (Hand, J.) ("It does not follow because 'Alcoa' had such a monopoly, that it 'monopolized' the ingot market; it may not have achieved monopoly; monopoly may have been thrust upon it. . . . This notion has usually been expressed by saying that size does not

the patent laws to see that new entrants and smaller firms have the protection to commit to a program of investment, innovation, and introductions of competing products.

Unless enforced, a patent is a "paper tiger."[74] Since 1982, the perceived value of patents has increased in the U.S. with the establishment of the Court of Appeals for the Federal Circuit.[75] The Federal Circuit's opinions have reduced forum shopping and uncertainty by rendering uniform legal standards where before there

---

determine guilt; that there must be some 'exclusion' of competitors; that the growth must be something else than 'natural' or 'normal'; that there must be a 'wrongful intent,' or some other specific intent; or that some 'unduly' coercive means must be used. At times there has been emphasis upon the use of the active verb, 'monopolize' . . . . The successful competitor, having been urged to compete, must not be turned upon when he wins."). *See generally* EARL W. KINTNER & JOSEPH P. BAUER, FEDERAL ANTITRUST LAW §§ 9-17 (1980 & Supp. 1994).

74 The U.S. sought to expand international protection for computer-related inventions as part of the GATT TRIPS negotiations. The eventual agreement has language that likely will be interpreted to cover software in most signatory countries. *See* TRIPS, *supra* note 68, § 5, Article 33 ("[P]atents shall be available for any inventions, whether products or processes, in all fields of technology, ... patents shall be available and patent rights enjoyable without discrimination as to place of invention, the field of technology and whether products are imported or locally produced."). *See also* Howard G. Pollack, *The Gordian Algorithm: An Attempt to Untangle the International Dilemma Over the Protection of Computer Software*, 22 LAW & POL'Y INT'L BUS. 815 (a brief review of software patent protection in the EC, Japan and Canada); PROPRIETARY RIGHTS COMMITTEE, FOREIGN SOFTWARE PATENTS (1990); *see generally infra* note 231.

75 Appeals on patent cases litigated in the various federal district courts now go directly to the Federal Circuit as opposed to the regional circuit courts. In addition, the Federal Circuit is the successor of the Court of Customs and Patent Appeals (CCPA) in hearing appeals from the PTO's Board of Patent Appeals and Interferences. *See* Federal Courts Improvement Act of 1982, Pub. L. No. 97-164, §§ 126-27, 96 Stat. 25, 37 (1982), *codified at* 28 U.S.C. §§ 1292-95 (1988); *see also* 1982 U.S. CODE CONG. & ADMIN. NEWS 11, 12-17, 28-32 (legislative intent of the act).

In 1981, Intel Chief Counsel Roger S. Borovoy was quoted saying, "In the electronics industry, patents are of no value whatsoever in spurring research and development. We use them because we have to. You can't be the only holdout against the angry hordes or else you pay everyone." *The Patent is Expiring as a Spur to Innovation*, BUS. WK., May 11, 1981, at 44E. Perceptions have changed since then. A guide to the literature asserting that the Federal Circuit is strongly pro-patentee is provided in: John C. Jarosz, *The Federal Circuit and its Patent Damages Decisions*, 1 U. BALT. INTELL. PROP. L.J. 17 nn. 2 & 4 (1992).

had been circuit splits.[76] From 1982 to 1992, there were one hundred and fifty-two reported patent infringement cases with damage awards totalling $1.73 billion.[77]

2. Public Access to Innovations

The patent system fosters the dispersal of knowledge by requiring patent applicants to reveal the "best mode" for practicing their inventions.[78] For a patent to withstand litigation, it must not only recite the invention claimed, but must also enable someone (a fictional person ordinarily skilled in the relevant field) to practice the invention.[79] The patent bargain is that in exchange for informing the public about the invention—and, hopefully stimulating further advances—the patentee gains the right to exclude others from profiting on the claim without approval.

The bargain may be lopsided. Patent holders prefer to gain the profits from a broad range of inventions while revealing no more information than necessary. A patent issued by the PTO is presumed valid.[80] However, patent invalidity is a legitimate affirmative defense in a patent infringement lawsuit.[81] In the realm of disclosure, three lines of attack are available. Each is founded upon a separate provision of section 112 ¶ 1 of the Patent Act which states:

---

76 The issues include: the presumption of validity of issued patents, treating a patent as a whole rather than dissecting it for the heart of the invention, and the rule of reason standard for patent misuse. Daniel F. Perez, *Exploitation and Enforcement of Intellectual Property Rights*, COMPUTER LAW., Aug. 1993, at 10.

77 These figures only include cases report in West's Federal Reporter series that reached final judgment. Ronald B. Coolley, *Overview and Statistical Study of the Law on Patent Damages*, 75 J. PAT & TRADEMARK OFF. SOC'Y 515, 518. Excluding the two largest awards, damages totalled $654 million. Damages were broken down into these categories: 33% interest, 28% reasonable royalty, 25% lost profits, 9% increased damages, 4% price erosion, and 1% attorney's fees. *Id.*

One view has it that the Federal Circuit is strongly pro-patentee. A more apt explanation would be that the court tends to uphold the factual findings, both pro and con, of the trial court. Jarosz, *supra* note 75. *See also* Paul R. Michel, *Appellate Advocacy—One Judge's Point of View*, FED. CIR. B.J., Summer 1991, at 1 (Judge Michel has been on the Federal Circuit since March 1988); Victor G. Savikas, *Survey Lets Judges Render Some Opinions About the Patent Bar*, NAT'L L.J., Jan. 18, 1993, at S7 (From 1982 to 1992, the number of patent jury trials increased from 30 to 51 while non-jury trials dropped from 72 to 49.).

78 35 U.S.C. § 112 ¶ 1 (1988).

79 *Id.*

80 35 U.S.C. § 282 (1988).

81 *Id.*

> 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.[82]

Parsed out, the requirements are that a patent: (1) describe, (2) enable, and (3) provide the best mode for practicing an invention.[83] Each of these elements is discussed separately below.

C. Mandated Disclosure Rules

1. Written Description

The statute requires a patent to contain the elements of a well-written speech: tell 'em what you are going to say, tell 'em, and, then, tell 'em what you said. The last part is the claims language—codified separately in section 112 ¶ 2.[84] However, the first two parts have a tendency to become confused. Part one is the written description requirement which introduces the invention. Part two, which takes up the bulk of a typical patent, teaches the invention to others through satisfaction of the separate requirements of enablement and best mode.

---

82 *See supra* note 78.

83 *See generally* David Bender & Anthony R. Barkume, *Disclosure Requirements for Software-Related Patents*, 10 COMPUTER LAW., Oct. 1991, at 1; D.C. Toedt III, *Patents for Inventions Utilizing Computer Software: Some Practical Pointers*, 9 COMPUTER LAW., Oct. 1992, at 12.

84 The "tell 'em what you said" portion consists of the claim language that must be at the end of a patent. In both senses, this is the pay-off as, "To support an infringement determination, an accused device must embody exactly each claim limitation or its equivalent." Key Manufacturing Group v. Microdot, 925 F.2d 1444, 1449 (Fed. Cir. 1991). *See also* Graver Tank v. Linde Air Products Co., 339 U.S. 605 (1950) (doctrine of equivalents triple identity test requiress proof that infringing device must have substantially same function, way, and result); SRI International v. Matsushita Elec. Corp., 775 F.2d 1107, 1117-18 (Fed. Cir. 1985) (en banc) (Infringement is determined by a patent's claims, not a patent specification's description of the examples or preferred embodiments. The prosecution history and specification are useful tools of claim construction.) Claim construction is a question of law. The statute requires:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

35 U.S.C. § 112 ¶ 2 (1988).

The leading case on the description requirement is *Vas-Cath Inc. v. Mahurkar*[85] decided in 1991. Judge Giles Rich wrote the opinion in this case which involved a patent for an improved arterial catheter. Judge Rich recounted the pragmatic purpose of the description requirement. Applications sometimes take years in prosecution at the PTO before claim language is finalized and a patent is issued. Continuations,[86] amendments,[87] reexaminations,[88] and reissues[89] all can significantly extend the period in which the inventor holds onto the original filing date for priority purposes. "Adequate description of the invention guards against the inventor's overreaching by insisting that he recount his invention in such detail that his future claims can be determined to be encompassed within his original creation."[90]

The patent application is viewed as a whole in satisfying the description requirement. The prose, specifications, and drawings may each alone, or in combination, suffice. The test is whether the application has "convey[ed] 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."[91] By looking at the breath of the description, the PTO and other interested parties can call "foul" if the applicant seeks a later modification of the claim's language to cover a more expansive and likely more valuable invention.[92]

This requirement is particularly important for software applications due to the long backlog of applications.[93] Judicial enforcement of this provision should discourage applicants from writing overly broad descriptions and engaging in delay tactics during the application process in the hope of netting a best-selling application program.[94] Not to be confused with each other, "'[t]he de-

85 935 F.2d 1555 (Fed. Cir. 1991).

86 *See* U.S. PATENT AND TRADEMARK OFFICE, MANUAL OF PATENT EXAMINING PROCEDURE § 201.07 to .08 (5th ed. 1983, rev. 1992) [hereinafter MPEP].

87 *See* MPEP § 714 Amendments, Applicants Action.

88 *See* MPEP §§ 2209-94.

89 *See* MPEP §§ 1401-90.

90 *Vas-Cath*, 935 F.2d at 1561. *See also* In re Hayes Microcomputer Prods., 982 F.2d 1527 (Fed. Cir. 1992) (For a software-related invention, source code need not be disclosed to satisfy the description requirement.).

91 *Vas-Cath*, 935 F.2d at 1563-65.

92 *See* MPEP § 714 Amendments, Applicants Action.

93 *See infra* note 206 and accompanying text.

94 To analogize, "Are you Dr. Galazkiewicz? . . . Yes, I am." *Coach K's Spoof a Hit; CBS to Show Encore*, ARIZONA REPUBLIC, Apr. 2, 1994, at E2.

scription requirement is found in 35 U.S.C. § 112 and is separate from the enablement requirement of that provision.'"[95]

2. Enablement

Through enablement, the patentee empowers others to replicate the invention. This is an enhanced version of the publication process in scientific journals. In science, inadequate disclosure can lead to disrespect, derision, and censure by one's colleagues—witness the outcome of the cold fusion hoax.[96] In the patent field, holding back useful information to practice the invention can lead to the patent being declared invalid. The Federal Circuit has stated:

> Although the statute does not say so, enablement requires that the specification teach those in the art to make and use the invention without "undue experimentation." . . . That some experimentation may be required is not fatal; the issue is whether the amount of experimentation required is "undue."[97]

Whether a claimed invention is enabled is a question of law reviewable *de novo* on appeal.[98] In patent litigation, the party challenging validity must show non-enablement by clear and convincing evidence.[99]

In the software field, the question of enablement centers on whether the source code must be disclosed in the patent application. Source code consists of the instructions a programmer writes to control a computer's future operation.[100] To the lay person,

---

95 *Vas-Cath*, 935 F.2d at 1563 (*quoting* In re Wilder, 736 F.2d 1516, 1520 (Fed. Cir. 1984), *cert. denied*, 469 U.S. 1209 (1985)). The court also preemptively addressed conflicting language that "intertwined" description and enablement analysis in its prior opinions by noting with disapproval Kennecott Corp. v. Kyocera Int'l, 835 F.2d 1419, 1421 (Fed. Cir. 1987), *cert. denied*, 486 U.S. 1008 (1988). *But see* Kevin S. Rhoades, *The Section 112 "Description Requirement"—A Misbegotten Provision Confirmed*, 74 J. PAT. & TRADEMARK OFF. SOC'Y 869 (1992) (a critical treatment of the *Vas-Cath* decision).

96 Andrew J. Pollack, *Cold Fusion, Derided in U.S., Is Hot in Japan*, N.Y. TIMES, Nov. 17, 1992, at C1.

97 In re Vaeck, 947 F.2d 488, 492 (Fed. Cir. 1991).

98 Moleculon Research Corp. v. CBS, Inc., 793 F.2d 1261, 1268 (Fed. Cir. 1986), *cert. denied*, 479 U.S. 1030 (1987).

99 Ralston Purina Co. v. Far-Mar Co., 772 F.2d 1570, 1574 (Fed. Cir. 1985).

100 Source code is written in assembly language (a low-level language where the programmer needs to explicitly instruct the computer on the minutia of the program's operation—tends to be lengthy when written out, but blindingly fast when executed) or a high-level language that allow the programmer to be significantly more productive in creating systems, but run somewhat slower than assembly language programs due to the greater overhead. Numerous high-level languages exist that are tailored to the type of

source code can resemble, depending on the computer language chosen, anything from quasi-English structured language to Egyptian hieroglyphics. If the source code is not revealed, the specifications, flow-charts, and other information contained in the patent must be sufficient for a skilled programmer to code an implementation without "undue experimentation."[101]

As the case law stands, if it only takes a skilled programmer one to two days to program the invention, the patent is probably enabling while 1.5 to 2 years of work qualifies as "undue experimentation." Between those benchmarks, it's anyone's guess.[102]

The pre-Federal Circuit opinions were summarized in *Hirschfeld v. Banner*.[103] Judge Howard Markey, then Chief Judge

---

system being created such as C++ (Microsoft Windows and UNIX programs), COBOL (traditional IBM MVS databases), Natural (relational databases), FORTRAN (scientific computing), PASCAL (educational computing), and Lotus 1-2-3 macros (financial spreadsheet automation).

Both assembly and high-level language programs must be compiled into object (machine) code before they can be executed by the computer. Source code is the language of the programmer, object code is the binary 0's and 1's that is executable directly by a digital computer. *See generally* FINDING A BALANCE, *supra* note 25, at 7, 149 (exerts of the same program written in PASCAL, assembler, and machine language); ANDY JOHNSON-LAIRD, SOFTWARE DEVELOPMENT AND "REVERSE ENGINEERING" (1990); James Canfield, Note, *The Copyrightability of Object Code*, 59 NOTRE DAME L. REV. 412 (1984); S. Carran Daughtrey, Note, *Reverse Engineering of Software for Interoperability and Analysis*, 47 VAND. L. REV. 145 (1994). Packaged software is distributed almost exclusively in object code format. However, "Good engineers distinguish between *what* a component does (the abstraction seen by the user) and *how* it does it (the implementation inside the black box)." JON BENTLEY, PROGRAMMING PEARLS 134 (1986).

101 *See infra* notes 103-17 and accompanying text.

102 In the context of analyzing immunoassay methods for a class of antibodies, the Federal Circuit summarized the factors to consider in determining whether to invalidate patent claims for lack of an enabling disclosure:

> (1) the quantity of experimentation necessary,
> (2) the amount of direction or guidance presented,
> (3) the presence/absence of working examples,
> (4) the nature of the invention,
> (5) the state of the prior art,
> (6) the relative skill of those in the art,
> (7) the predictability or unpredictability of the art, and
> (8) the breadth of the claims.

In re Wands, 858 F.2d 731, 736 (Fed. Cir. 1988) (numbering in original).

103 462 F. Supp. 135 (D.D.C. 1978) (Markey, J., sitting by designation), *aff'd*, 615 F.2d 1368 (D.C. Cir. 1980) (unpublished), *cert. denied*, 450 U.S. 994 (1981).

Prior to *Hirschfeld*, the CCPA heard three cases on source code disclosure: In re Naquin, 398 F.2d 863, 866 (CCPA 1968) (Affidavits by expert's skilled in the art are sufficient to overcome an examiner's objection to non-enablement—"In the absence of a challenge to the affiant's qualifications or, at the very least, a contrary inference by the board of examiners from other evidence . . . ."); In re Brown, 477 F.2d 946, 951-52 (CCPA 1973) (Affidavits of expert is insufficient when opinion colored by information

of the Court of Customs and Patent Appeals (CCPA) and later first Chief Judge of the Federal Circuit, heard the case in the trial court by designation. The case involved an appeal from the PTO Board of Patent Interferences and Appeals (the Board) denying patentability for a software driven focusing system for a TV picture tube. The applicants submitted schematic block diagrams describing the software but not the source code. In response to a challenge by the patent examiner, the applicants obtained an affidavit from an expert saying that he had created an embodiment of the invention, albeit with some bugs, using just the patent application in four hours. The Board upheld the examiner's rejection of the application saying:

> In our view, the instant disclosure fails to disclose in full, clear and exact terms how such elements, if they exist, may be selected, interconnected, timed, programmed and controlled so as to obtain the system operations claimed. As such, we view appellants' disclosure as little more than an invitation to those skilled in the art to experiment extensively so as to reduce the system to practice.[104]

Judge Markey reversed. The factual evidence of enablement was enough to overcome PTO's presumption of correctness. The PTO failed to call any witnesses in the trial court and conducted a "limited cross-examination of [applicant's] witnesses . . . [which] produced no change and reflected no weaknesses in their testimony."[105]

The Federal Circuit addressed disclosure of source code in *White Consolidated Industries v. Vega Servo-Control*.[106] The patent was directed to a numerically controlled system for machine tools that eliminated the need to create a new part program for each tool.[107] To practice the invention, one needed access to a partic-

---

obtained outside of the application's disclosure.); In re Brandstadter, 484 F.2d 1395 (CCPA 1973) (Invention consisted of automated telephone messaging system on PBX. Bell Labs would not reveal any details of software in electronic messaging apparatus needed to practice invention because they considered it proprietary [sic]. *Id.* at 1403. Furthermore, affidavits simply asserted that one skilled in art could practice invention, but lacked any indication of time or effort needed. *Id.* at 1406-07.).

See also, Steven T. Naumann, *Compliance with 35 U.S.C. § 112 for Inventions Containing Computer Software: Is Disclosure of the Computer Code Required?*, 4 SOFTWARE L.J. 443 (1991).

104 *Hirschfeld*, 462 F.Supp. at 138.

105 *Id.* at 138.

106 713 F.2d 788 (Fed. Cir. 1983).

107 *Id.* at 789. *See generally supra* note 29.

ular compiler,[108] "SPLIT," which was held as a trade secret by the applicant, to translate the source code into object code. The court stated, "one may refer to an element of a claimed invention held as a trade secret by name only and yet satisfy 35 U.S.C. § 112 if equivalent elements are known, and known to be equivalents, and available to those skilled in the art . . . ."[109] Although SPLIT and other compilers were for sale on the market, only SPLIT had the features needed to implement the invention. Testimony was introduced that it would take 1.5 to 2 man years to create an equivalent.[110] In rejecting the patent, the court stated: "The *sine qua non* of a valid patent is a full, clear, enabling description of the invention . . . . Though the 'language translator' by itself is not the claimed invention, it is an integral part of the disclosure necessary to enable those skilled in the art to 'make and use the same.'"[111]

In 1980, the Federal Circuit revisited the issue of source code disclosure in *Northern Telecom v. Datapoint Corporation.*[112] Northern Telecom was the successor-in-interest to a company that had built a major business on its patent for a programmable batch data entry terminal.[113] The primary users of the system were clerks who would key transactions into the terminal. The terminal was programmed to do validity checking on the inputed data and also provide temporary storage for the data until it was transmitted to a host computer (typically an IBM mainframe). The terminal's advantage was that it reduced the processing done on the host computer. At the time of the invention, the typical IBM terminal was a "frame-buffer" model that would transmit updated information on the terminal screen's contents (whenever "enter" or certain other keys were hit) to the host computer for checking, verifying, and storing the data.

It was the hardware, not the software, that distinguished this invention. The terminal served to free the host computer from

---

108 *See supra* note 100.

109 *White*, 713 F.2d at 790.

110 Estimates of man years in budgeting time to develop computing systems is at best an art, at worst a dart game. The practice of CEO Charles Wang of Computer Associates, the largest independent developer of software for mainframes, is to reduce the number of employees working on a project in which he needs to speed up the development effort. *See generally* FREDERICK P. BROOKS, JR., THE MYTHICAL MAN-MONTH: ESSAYS IN SOFTWARE ENGINEERING (1975).

111 *White*, 713 F.2d at 791.

112 908 F.2d 931 (Fed. Cir. 1990).

113 *Id.* at 933.

supervising routine data entry. The court stated: "The claimed invention of the '375 patent is not in the details of the program writing, but in the apparatus and method whose patentability is based on the claimed combination of components or steps."[114] The patent specification contained only a cursory reference to how the terminal's software was loaded and stored. Neither side in this patent infringement suit disputed that writing the software for the terminal would be "straightforward" and "obvious" to someone skilled in the art.[115] The court concluded that the programs were "routine" and would only take the "ordinary effort" of a "programmer of reasonable skill" to duplicate.[116] The court gave no indication of how long this "ordinary effort" would take. In terms of equity, the court was probably unwilling to hold the patent invalid on this peripheral issue.[117]

3. Best Mode

The demarcation between enablement and best mode language is not always clear.[118] The purpose of the best mode requirement is that the applicant "plays 'fair and square' with the patent system."[119] To illustrate, suppose you enter a new town and ask directions to Aunt Millie's. One person tells you go straight ahead to the third light, make a right, and its the fourth house on the left. Another person gives directions that have you meandering down every alley in town until, hours later, Aunt Millie greets you with a cold supper. Both sets of directions are en-

---

114 *Id.* at 941.

115 *Id.* at 941-42.

116 *Id.* at 943.

117 *See also* In re Hayes Microcomputer Prods., 982 F.2d 1527, 1533-37 (Fed. Cir. 1992) (similar discussion of enablement where narrowly claimed invention was a relatively simple technique); *see infra* notes 124-30 and accompanying text; *cf.* Amgen, Inc. v. Chugai Pharm. Co., 927 F.2d 1200, 1210-11 (1991) ("For many years, it has been customary for patent applicants to place microorganism samples in a public depository when such a sample is necessary to carry out a claimed invention. . . . When a biological sample is obtained from nature, the invention may be incapable of being practiced without access to that organism. . . . [when] the organism is created by insertion of genetic material into a cell obtained from generally available sources, then all that is required is a description of the best mode and an adequate description of the means of carrying out the invention, not deposit of the cells.")

118 Sometimes the sub-heading in a case might read "Best Mode," but the analysis will closely track that used in an enablement inquiry. For example, the In re Sherwood, 613 F.2d 809 (CCPA 1978) dcision's section on "Best Mode" discusses aspects of the record that today would more typically be part of an enablement inquiry.

119 Amgen, Inc. v. Chugai Pharm. Co., 927 F.2d 1200 (Fed Cir. 1991).

abling, only one is the best mode. The Federal Circuit has summarized the test as:

> A best mode analysis has two components. The first inquiry focuses on whether the inventor knew of a mode of practicing his invention at the time he filed his patent application which he considered to be better than any other. This determination is subjective, focusing on the inventor's state of mind at the time he filed his application. If he did have a best mode, the next question is whether he disclosed it and did so adequately to enable one of ordinary skill in the art to practice the best mode. This is an objective determination. There must be no concealment of a mode known by the inventor to be better than that which is disclosed.[120]

The CCPA considered best mode disclosure of source code in a Judge Markey opinion with language that more resembled an enablement inquiry than a best mode analysis. *In re Sherwood*[121] involved a specification for a computer program used in seismic wave prospecting for petroleum. Holding that the application before the PTO complied with the best mode requirement, the court relied on a affidavit stating that the specifications in the patent application contained a fully complete specification of the "human-to-human communication" a systems analyst might provide for a programmer charged with writing the program code.[122] In a juxtaposition of views, Judge Markey stated,

> In general, writing a computer program may be a task requiring the most sublime of the inventive faculty or it may require only the droning use of a clerical skill . . . . [T]he conversion of a complete thought [the specification] . . . into the language a machine understands is necessarily a mere clerical function to a skilled programmer."[123]

---

120 In re Hayes Microcomputer Prods., 982 F.2d 1527 (Fed. Cir. 1992). Determining whether the best mode requirement is satisfied is a question of fact. DeGeorge v. Bernier, 768 F.2d 1318, 1324 (Fed. Cir. 1985). *See generally* Christopher S. Marchese, *Promoting the Progress of the Useful Arts by Narrowing Best Mode Disclosure Requirements in Patent Law*, 54 U. PITT. L. REV. 589 (1993); Richard M. Mescher, Note, *Best Mode Disclosure—Genetic Engineers Get Their Trade Secret and Their Patent Too?*, 18 U. DAYTON L. REV. 177 (1992).

121 613 F.2d 809 (CCPA 1980).

122 *Id.* 816-17.

123 *Id.* at 816-17 and n.6. Having had years of experience writing so-called "complete" specifications and programming computers from the same, I shake my head at the unfamiliarity demonstrated of what is involved implementing systems in a complex environment—if only it was so easy!

The Federal Circuit first considered this issue in *In re Hayes Microcomputer Prods.*[124] Hayes was one of the early success stories in the PC industry. With the introduction of the IBM PC in 1981 and the liberalization of FCC rules governing what could be plugged into the switched network (the telephone jack), Hayes quickly came to dominate the PC modem market with its line of "Smartmodems." At the time, Hayes' larger competitors focused on traditional data processing markets and gave the upstart Hayes a few years lead in the high volume/lower priced PC market.[125] By the time the competing firms woke up to the missed opportunity, the market had standardized on Hayes' "AT" command set for controlling modem operations and escape sequence mechanism for switching between command and data transmission modes.[126]

Hayes was awarded an apparatus patent on an improved escape code sequence mechanism to switch between command and data modes.[127] In terms of enablement and best modes, the Federal Circuit held that the source code listing stored in ROM[128] did not have to be revealed.[129] The portion of code covering the

---

124 982 F.2d 1527 (Fed. Cir. 1992).

In an unpublished opinion, a Federal Circuit panel had previously upheld a district court's rejection of a best mode challenge to source code disclosure. Mendenhall v. Astec Industries Inc., 14 U.S.P.Q.2d (BNA) 1134 (E.D. Tenn. 1988), *aff'd*, 891 F.2d 299, 14 U.S.P.Q.2d (BNA) 1140 (Fed. Cir. 1989) (per curiam) (unpublished - the full text of the unpublished opinion can be found in U.S.P.Q.). The case involved a company that in-house came up with an automated weighing and dispensing system for an asphalt bin. It contracted out to a California company to write the computer program. The patent disclosed that the best mode for practicing the invention would be through a microprocessor control system (a thinly disguised way of saying software running on a PC which is a common claims drafting technique that does not narrow the invention). The details of implementing the invention were given in the language of industrial process controls. The district court stated, "There was nothing in the proof to indicate that once the patent was disclosed, anybody conversant with programming microprocessors could not have developed a similar system to perform its steps." *Id.* at 1140.

125 Matt Kramer, *When PC users want modems, they're apt to dial Hayes*, PC WK., July 10, 1984, at 71.

126 After Hayes had become the de facto standard, IBM tried selling an incompatible PC modem. IBM's product was never accepted by the marketplace. CARROLL, *supra* note 36, at 132.

127 The escape code was ingeniously simple. Three "+" signs would be surrounded by one second of "guardtime" (empty space) on each side to signal that data transmission should be interrupted. *Hayes*, 982 F.2d at 1532. Competing modem manufacturers had to mimic this function if they were to be compatible with other modems and popular communications software packages.

128 Read-only memory (a form of permanent memory in a computer that does not get erased when the power is turned off).

129 *Hayes*, 982 F.2d at 1533-39.

escape sequence would have been trivial.[130] However, the fortified beachhead the patent provided was enough to keep unlicensed competitors at bay.

D. Subject Matter: "Anything under the Sun" that Fits into one of Four Categories

The previous section laid out the patent system's disclosure requirements but begged the question of what can be patented. With thousands of issued patents having an integral software component,[131] it is remarkable that the question whether software *per se* can be patented is still hotly debated.[132] Section 101 of the patent statute provides:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.[133]

The black letter rule is that a patent is available for any novel,[134] useful,[135] and non-obvious[136] invention.

---

130 A lot more work would have gone into making a modem fully compatible with a Hayes. For example, the form and function of the "AT" command set would need to be incorporated.

131 FINDING A BALANCE, *supra* note 25, at 55; *see also* notes 205-06 and accompanying text.

132 *See supra* notes 17, 34 & 43.

133 35 U.S.C. § 101 (1988).

134 35 U.S.C. § 102 (1988). § 101 requires that an invention must be "new . . . subject to the conditions and requirements of this title." *Supra* note 133. "Specific conditions for patentability follow and § 102 covers in detail the conditions relating to novelty." Diamond v. Diehr, 450 U.S. 175, 189 (1981). § 102 has paragraphs (a) through (g) listing ways the right to a patent can be lost.

The patent statute today essentially remain unchanged from the last major revision completed in 1952. The primary author of that revision, Pasquale J. Federico, expounded on the novelty requirement saying:

> The novelty required is not novelty in an absolute sense, as the statute defines what is to be looked to in order to show that an invention is not new . . . . section 102 "may be said to describe the statutory novelty required for patentability, and includes, in effect, an amplification and definition of 'new' in section 101."

P.J. Federico, *Commentary on the New Patent Act*, 35 U.S.C.A. § 1 (1954 ed., West), *reprinted in* 75 J. PAT. & TRADEMARK OFF. SOC'Y 161, 178 (1993) (citation omitted).

135 35 U.S.C. § 101 (1988). "All that the law requires is that the invention should not be frivolous, or injurious to the well-being of society. The word *useful* therefore is incorporated into the act in contradistinction to mischievous or immoral." In re Nelson, 280 F.2d 172, 178-79 (CCPA 1960) (citation omitted).

If the invention does not fit within one the section 101's four categories (a process, machine, manufacture, or composition), the applicant will not get a utility patent.[137] Practitioners typically apply for software patents as either a process (a.k.a. method) or machine (a.k.a. apparatus).[138] Patents also may be sought for new useful combinations (a.k.a. means-plus-function claims which are very popular with software patents) as long as the parts of the combination fall within one of the four subject-matter categories. For software applications, this typically means that an element of hardware (i.e. a memory chip) is included in the specifications. The statutory basis for means-plus-function claims is:

> An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding

---

136 35 U.S.C. § 103 (1988). "Obviousness is a legal determination . . . . [whether] the subject matter as a whole would have been obvious to a person having ordinary skill in the art at the time the invention was made." In re Hayes Microcomputer Prods., 982 F.2d 1527, 1539-40 (Fed. Cir. 1992). "When determining the patentability of a claimed invention which combines two known elements, 'the question is whether there is something in the prior art as a whole to suggest the desirability, and thus the obviousness, of making the combination.'" In re Beattie, 974 F.2d 1309, 1311 (Fed. Cir. 1992) (citation omitted).

137 The two other types of patents are plant (35 U.S.C. § 161 (1988)) and design patents (35 U.S.C. § 171 (1988)). Plant patents are available for new asexually reproduced plant life. Design patents cover the ornamental characteristics of an article. It is uncertain whether design patents cover icons and other display elements in computer program displays. The PTO issued a few design patents to Xerox in the 1980s for software icons, but imposed a moratorium in 1989. *See* Lance L. Vietzke, *Design Patents for Icons: What is the Article of Manufacture?*, COMPUTER LAW., June 1993, at 15, 21 n. 45; *see generally* Robert Barr & Susan Hollander, *Design Patents Revisited: Icons as Statutory Subject Matter*, COMPUTER LAW., June 1992, at 13; David L. Hayes, *What's Left of 'Look and Feel': A Current Analysis (Part III)*, COMPUTER LAW., July 1993, at 13, 15-17.

An example of non-statutory subject matter is provided in this account:

> It's 1913, and we're at the Notre Dame versus Army football game . . . . Changing football forever, the Notre Dame quarterback throws the first-ever forward pass, winning the game. A week later, Notre Dame is facing another team, say Purdue . . . . So early in the first quarter, the Purdue quarterback throws a forward pass. The Notre Dame coach calls a time-out and sends young Knute Rockne jogging over to the Purdue bench.
>
> "Coach says that'll be five dollars," mumbles an embarrassed Knute, kicking at the dirt with his toe.
>
> "Say what, son?"
>
> "Coach says the forward pass is Notre Dame property, and if you're going to throw one, you'll have to pay us five dollars. I can take a check."

ROBERT X. CRINGELY, ACCIDENTAL EMPIRES 73-74 (1992).

138 *See* Firestone, *supra* note 10, at ch. 5, 33-40.

structure, material, or acts described in the specification and equivalents thereof.[139]

The determination of whether software-related inventions are statutory subject matter and patentable has been subject to varying arbitrary rules for more than two decades. Dozens of articles have already traced, dissected, and analyzed the history of the algorithm test[140] from *Benson* to the latest Federal Circuit and PTO decisions. This Note avoids a detailed treatment and rather focuses on the three Supreme Court cases that have addressed software patentability and some recent Federal Circuit cases that have made substantive changes in applying the test. This Note's first recommendation for improvement is to scrap the mathematical algorithm test.

III. *BENSON* TO *DIEHR*: CLOSING AND RE-OPENING THE WINDOW

A. *Gottschalk v. Benson*

Justice Douglas' opinion in *Gottschalk v. Benson* virtually foreclosed the patentability of computer programs.[141] Prior to reaching the Court, the CCPA had overturned the PTO's Board's decision to reject Benson's claims as non-statutory subject matter. The Supreme Court reversed holding that an algorithm solving a mathematical problem is not patentable. The two claims at issue covered a specific method for converting binary-coded decimal

---

139 35 U.S.C. § 112 ¶ 6 (1988). *See also infra* notes 180-89 and accompanying text.

140 *See especially* Donald S. Chisum, *The Patentability of Algorithms*, 47 U. PITT. L. REV. 959 (1986); Irah Donner, *Patenting Mathematical Algorithms that 'Embrace' Mother Nature*, COMPUTER LAW., May 1992, at 1; Irah H. Donner & J. Randall Beckers, *Throwing Out Baby* Benson *With the Bath Water: Proposing a New Test for Determining Statutory Subject Matter*, COMPUTER LAW., Jan. 1993, at 8; Steven W. Lundberg & John C. Reich, *Identifying Mathematical Algorithms in Patent Claims*, COMPUTER LAW., Sept. 1993, at 1; Alan D. Minsk, *The Patentability of Algorithms: A Review and Critical Analysis of the Current Doctrine*, 8 SANTA CLARA COMPUTER & HIGH TECH. L.J. 251 (1992); Alan D. Minsk, *The Patentability of Algorithms: An Update on the Current Doctrine*, 9 SANTA CLARA COMPUTER & HIGH TECH. L.J. 233 (1993); David C. Radulescu, *The Status of the Patentability of Subject Matter Containing "Mathematical Algorithms" after* Grams *and* Iwahashi; 74 J. PAT. & TRADEMARK OFF. SOC'Y 96 (Part I), 153 (Part II) (1992); Pamela Samuelson, Benson *Revisited: The Case Against Patent Protection for Algorithms and other Computer Program-Related Inventions*, 39 EMORY L.J. 1025 (1990); Richard H. Stern, *Tales from the Algorithm War:* Benson *to* Iwahashi, *It's Deja Vu All Over Again*, 18 AM. INTELL. PROP. L. ASS'N Q.J. 371 (1991); Jur Strobos, *Stalking the Elusive Patentable Software: Are there still* Diehr *or was it just a* Flook*?*, 6 HARV. J. OF LAW & TECH. 363 (1993); *see generally* 1 BENDER, *supra* note 14, at § 3A.03; 1 CHISUM, *supra* note 12, at §1.06(6)(i) n.191 (Supp. 1993) (provides an extensive bibliography of articles on the subject).

141 441 F.2d 682 (CCPA 1971), *rev'd*, 409 U.S. 63 (1972).

(BCD). numbers into pure binary numbers. Although Bell Labs intended to use the equations primarily in telephone switching systems, no such limitations were included in the application.[142] One claim, however, recited the physical "shift register" to be used in performing the translation. The court did not make a distinction between the claims writing:

> It is said that the decision precludes a patent for any program servicing a computer. We do not so hold . . . . What we come down to in a nutshell is the following.
>
> It is conceded that one may not patent an idea. But in practical effect that would be the result if the formula for converting BCD numerals to pure binary numerals were patented in this case . . . . the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself,[143]

The root of the current debate on the bounds of software patentability stems from the loose definition of algorithm given earlier in the opinion. The Court defined an algorithm as:

> A procedure for solving a given type of mathematical problem is known as an "algorithm." The procedures set forth in the present claims are of that kind; that is to say, they are a generalized formulation for programs to solve mathematical problems of converting one form of numerical representation to another.[144]

Courts have been grappling with this definition ever since. $E=mc^2$ is not, and should not, be patentable. However, a streamlined method for handling data translation should be protected. It is exactly for such a generalized "algorithm" that Stac won $120 million from Microsoft for infringing on Stac's compression technology. Today, there are innumerable translation problems of converting and compressing analog signals (the human voice) and real-time images (pictures and video) into digital packets for transmission over the incipient information superhighways.[145] There was an important distinction that was missed between inventing a new method of translation and routine computational problem

---

142 Stern, *supra* note 140, at n.5 and accompanying text.

143 *Benson*, 409 U.S. at 71-72.

144 *Id.* at 65.

145 *See, e.g.*, Steve Ditlea, *Digital Compression: Squeezing Video to Fit*, HEMISPHERES, Jan. 1994, at 105.

solving.[146] If Bell Labs' algorithm in *Benson* was not inventive, it should have been rejected on the basis of sections 102 (novelty) or 103 (obviousness). Concerns that the "practical effect" was that the patentee was trying to "wholly pre-empt" all uses of the algorithm could have been addressed through both the overbreadth doctrine[147] and enablement requirement.[148]

The *Benson* court did not need to create a new test for non-statutory subject matter. With citations to the Supreme Court's prior decisions, the opinion began with a summary statement: "Phenomena of nature, though just discovered, mental processes, and abstract intellectual concepts are not patentable, as they are the basic tools of scientific and technological work."[149] This was later restated in *Diamond v. Diehr*: "Excluded from such patent protection are laws of nature, natural phenomena, and abstract ideas. 'An idea of itself is not patentable.'"[150] The *Benson* algorithms, designed using applied mathematics, do not fall into any of these categories.

A *realpolitik* explanation offered for the outcome in *Benson* would rest on the opposition lined up against software patents by 1972. During the early 1960s, the Patent Office was faced with a limited budget, antiquated procedures, a four year pendency on applications, and a desire to reject software as "creations in the area of thought."[151] A presidential commission was formed to study the issue.[152] A portion of the 1966 report, quoted verbatim in the *Benson* opinion, stated:

> The Patent Office now cannot examine applications for programs because of a lack of a classification technique and the requisite search files. Even if these were available, reliable searches would not be feasible or economic because of the tremendous volume of prior art being generated. Without this search, the patenting of programs would be tantamount to

146 An in-depth exploration of this distinction is provided in Chisum, *The Patentability of Algorithms*, *supra* note 140, at 972-92.

147 The *Benson* decision discussed some of the leading cases dealing with the overbreadth doctrine but did not employ the doctrine in crafting the holding of the case. *Benson*, 409 U.S. at 68-70. O'Reilly v. Morse, 56 U.S. (15 How.) 62 (1854) (Samuel Morse trying to claim future uses of electromagnetism at distances not enabled by his work). *See* Strobos, *supra* note 140, at 365 n.8.

148 *See supra* notes 97-117 and accompanying text.

149 *Benson*, 409 U.S. at 67.

150 *Diehr*, 450 U.S. at 185.

151 FINDING A BALANCE, *supra* note 25, at 45.

152 Samuelson, *supra* note 140, at 1038-39 nn.39-41.

> mere registration and the presumption of validity would be all but nonexistent.
>
> It is noted that the creation of programs has undergone substantial and satisfactory growth in the absence of patent protection and the copyright protection for programs is presently available.[153]

Political pressure from software developers prevented the recommendation from being enacted into legislation.[154] The Patent Office tried to impose guidelines excluding software, but the CCPA struck them down.[155] Four major hardware manufactures, endorsers of the original report, then submitted amici curiae briefs in *Benson* opposing software patents.[156] They succeeded. From seventy software-related patents issued in 1972, the number dropped to five (heavily disguised) patents issued in 1976.[157]

B. Parker v. Flook

The Supreme Court next addressed section 101 in *Parker v. Flook*.[158] The invention was a system that polled various sensors and computed alarm limits for catalytic converters. The application revealed the algorithm used in determining the limits but did not teach the chemical processes at work or even hint at the proper values for user variables to insure safe operation.[159] For a divided court, Justice Stevens wrote:

> Respondent's application simply provides *a new and presumably better method* for calculating alarm limit values. If we assume that method was also known, as we must under the reasoning in Morse, then respondent's claim is, in effect, comparable to a claim that the formula $2\pi r$ can be usefully applied in determining the circumference of a wheel . . . . Very simply, our hold-

153 REPORT OF THE PRESIDENT'S COMMISSION ON THE PATENT SYSTEM, "TO PROMOTE THE PROGRESS OF . . . USEFUL ARTS" IN AN AGE OF EXPLODING TECHNOLOGY 13 (1966). *Benson*, 409 U.S. at 72. The same complaints are still heard today.

154 FINDING A BALANCE, *supra* note 25, at 45-46.

155 *Id.* at 45-47.

156 Samuelson, *supra* note 140, at 1053 n.90.

157 FINDING A BALANCE, *supra* note 25, at 55.

158 437 U.S. 584 (1978). Another case had come up for review on § 101, but the Supreme Court reversed the CCPA solely on the basis of § 103 (obviousness) and avoided discussing § 101. In re Johnston, 502 F.2d 765 (Fed. Cir. 1974), *rev'd sub nom.*, Dann v. Johnston, 425 U.S. 219 (1976).

159 *Id.* at 585-87.

> ing today is that a claim for an improved method of calculation, even when tied to a specific end use, is unpatentable subject matter under § 101.[160]

This decision represents the outer limits to which the mathematical algorithm test has been taken. The court acknowledged that, "He [the patentee] does not seek to 'wholly preempt the mathematical formula.'"[161] Yet, the Court equated the geometric law that the circumference of a circle is approximated by $2\pi r$ with a new formula for automating the laborious process of collecting data and processing alarm limits. Both examples are mathematical in that numerical values are computed. However, $2\pi r$ is an algorithm describing a law of nature while a collection and processing algorithm is a demonstration of the human creative faculty.[162] The same process performed using mechanical and analog equipment would easily qualify as statutory subject matter.

The case could have been decided based on the inventor's failure to meet the disclosure requirement by concealing information needed to practice the invention.[163] In addition, if the implementation on a computer was a straight porting of manual procedures, the claims could also have been denied through the rubric of sections 102 (novelty) and 103 (obviousness). There was no need to assume *arguendo* that the invention was "a new and presumably better method."[164] Judge Stevens' opinion foreshadowed his dissent in *Diamond v. Diehr* when he recommended "(1) an unequivocal holding that no program-related invention is a patentable process under § 101 . . . . and (2) an unequivocal explanation that the term 'algorithm' . . . is synonymous with the term 'computer program.'"[165]

C. Diamond v. Diehr

The Court returned to the issue in *Diamond v. Diehr* after a series of CCPA decisions that demonstrated an "unenthusiastic reception" to the earlier rulings.[166] *Diehr* was argued less than

---

160 *Id.* at 594-95 & n.18 (emphasis added) (6-3 decision).

161 *Id.* at 589.

162 *See infra* notes 170, 175 and accompanying text.

163 These omissions are discussed at greater length in Diamond v. Diehr, 450 U.S. 175, 186-88 (1981).

164 *Flook*, 437 U.S. at 594.

165 Diamond v. Diehr, 450 U.S. 175, 219 (Stevens, J., dissenting).

166 *Id.* at 211 n.34 (5-4 decision) (Stevens. J., dissenting) (Justice Stevens provides an extensive review of the historical debate and a detailed survey of the CCPA rulings.).

four months after the *Diamond v. Chakrabarty* opinion was announced.[167] *Chakrabarty* was the landmark case in which the Court held that live, human-made micro-organisms were patentable subject matter. Writing for the *Diehr* court, Justice Rehnquist quoted from the language of *Chakrabarty* and the history of the Patent Act in support of the proposition that statutory subject-matter was intended to "include anything under the sun that is made by man."[168] In creating new rules governing what is to be covered by the patent laws, the court stated:

> In cases of statutory construction, we begin with the language of the statute. Unless otherwise defined, "words will be interpreted as taking their ordinary, contemporary, common meaning," and in dealing with the patent laws, we have more than once cautioned that "courts 'should not read into the patent law limitations and conditions which the legislature has not expressed.'"[169]

In *Diehr*, the Court upheld a CCPA decision granting a method patent where the invention consisted of using a well-known formula in a computer program to control the manufacture of synthetic rubber. The software determined how long raw materials should be left heating in a mold and then signaled the machinery to open the press and remove the cured rubber. Except for the use of a computer program to control the machinery, all steps in the process were contained in the prior art. The *Diehr* Court signaled that its prior opinions in *Benson* and *Flook* were to "stand for no more than these long-established principles" that "[e]xcluded from such patent protection are laws of nature, natural phenomena, and abstract ideas."[170]

---

167 447 U.S. 303 (1980) (opinion rendered on June 16, 1980; oral arguments in *Diehr* were held on October 14, 1980).

168 *Diehr*, 450 U.S. at 182. *Chakrabarty*, 447 U.S. at 309. S. Rep. No. 1979, 82d Cong., 2d Sess., 5 (1952); H.R. Rep. No. 1923, 82d Cong., 2d Sess., 6 (1952).

169 *Diehr*, 450 U.S. at 182 (citations omitted).

170 *Id.* at 185. "Although commentators have differed in their interpretations of *Benson*, *Flook*, and *Diehr*, it appears to be generally agreed that these decisions represent evolving views of the Court, and that the reasoning in *Diehr* not only elaborated on, but in part superseded, that of *Benson* and *Flook*." Arrhythmia Research Tech. v. Corazonix Corp., 958 F.2d 1053, 1057 n.4 (1992).

IV. THE CURRENT LANDSCAPE AT THE FEDERAL CIRCUIT AND PTO

A. Federal Circuit Decisions

The "unenthusiastic reception" of the CCPA to the Supreme Court's rulings was demonstrated in a series of decisions in which it allowed software to be patented.[171] Eventually, the CCPA's "interpretation" of the Supreme Court's trilogy was cast as a two-part test later summarized by the Federal Circuit:

> This analysis has been designated the *Freeman-Walter-Abele* test for statutory subject matter. It is first determined whether a mathematical algorithm is recited directly or indirectly in the claim. If so, it is next determined whether the claimed invention as a whole is no more than the algorithm itself; that is, whether the claim is directed to a mathematical algorithm that is not applied to or limited by physical elements or process steps. Such claims are nonstatutory. However, when the mathematical algorithm is applied in one or more steps of an otherwise statutory process claim, or one or more elements of an otherwise statutory apparatus claim, the requirements of section 101 are met.[172]

The first step requires a determination of what is a "mathematical algorithm." As mentioned above, *Benson* used the term "mathematical formula."[173] In *Diehr*, the Court directed that the interpretation be "limited to the more narrow definition employed by the Court" without explicitly saying what it was other than a "procedure for solving a given type of mathematical problem."[174] The current favored definition limits the term to computing one or more sets of numbers from another.[175] The importance of this

---

171 *See supra* note 166 and accompanying text.

172 *Arrhythmia*, 958 F.2d at 1058. The three cases in which the test was derived from are: In re Freeman, 573 F.2d 1237 (CCPA 1978); In re Walter, 618 F.2d 758 (CCPA 1980); and In re Abele, 684 F.2d 902 (CCPA 1982).

173 *See supra* note 143 and accompanying text.

174 *Diehr*, 450 U.S. at 186 n.9.

175 This latest definition was devised by a PTO Board as providing:

> [W]e believe a claim should be considered as reciting a mathematical algorithm only if it essentially recites, directly or indirectly, a method of computing one or more numbers from a different set of numbers by performing a series of mathematical computation. Consequently, a claim which essentially recites another type of method does not recite a mathematical algorithm, even though it incidentally requires, either directly or indirectly, the performance of some mathematical computations. In our view, this approach correctly places the emphasis on what the claimed method steps do rather than how the steps are performed.

limitation will diminish as computer science migrates to more abstract concepts such as neural nets, and software development concentrates more on multimedia titles and user interfaces.

Even so, the *Freeman-Walter-Abele* test does not provide a meaningful barrier to the patentability of software applications. In 1989, the Federal Circuit decided that the inclusion of a ROM (read-only memory) chip as an element of a means-plus-function claim satisfied the second part of *Freeman-Walter-Abele* inquiry.[176] The *In re Iwahashi* claims were for software that used a simplified method for computing auto-correlation coefficients used in pattern recognition. It is a simple fact that software needs hardware memory chips on which to run. Presumably, through operation of the doctrine of equivalents, the scope of protection would extend to the software's operation on other platforms. In effect, "the Federal Circuit has effectively thrown the patent system's doors wide open to algorithms—if the correct formalities are observed."[177]

The new openness was highlighted in *Arrhythmia Research Technology v. Corazonix Corporation.*[178] The *Arrhythmia* claims were for a medical diagnostic system that would "number-crunch" values taken from monitoring equipment in order to predict the likelihood of a heart attack from ventricular tachycardia. Whereas *Iwahashi* involved means-plus-function apparatus claims, the *Arrhythmia* court upheld process claims as the invention consisted of "physical process steps that transform one physical, electrical signal into another."[179]

B. PTO Reaction

The Federal Circuit decisions have met resistance from the PTO. Just before *Iwahashi* was decided, the solicitor's office at the PTO prepared and published an analysis of the PTO's interpreta-

---

Ex Parte Logan, 20 USPQ2d 1465, 1468 (Bd. Pat. App. and Interf. 1991).

As computers perform all operations using computations, this is not taken literally. It focuses more on the task that the programmer is implementing rather than what happens after the program is compiled. In re Phillips, 608 F.2d 879, 882 (CCPA 1979). *See also* Lundberg & Reich, *supra* note 140, at 6 (An appendix chart, covering 1938-92, lists how courts have analyzed the presence or absence of a mathematical algorithm in a claim.).

176 In re Iwahashi, 888 F.2d 1370, 1375 (Fed. Cir. 1989).

177 Stern, *supra* note 140, at 372. *But see* In re Grams, 888 F.2d 835, 839 n.4 (Fed. Cir. 1989) (focuses on physical transformation of matter for statutory process).

178 958 F.2d 1053 (Fed. Cir. 1992).

179 *Id.* at 1059.

tion of the case authority on algorithms.[180] The PTO report had as a key test whether the software works a "transformation of something physical into a different form . . . . It is manifest that the statutory nature of the subject matter does not depend on the labels 'signals' or 'data.'"[181] In response to *Iwahashi*, the PTO issued another notice stating that its position on algorithms was "unaffected" and rejected the Federal Circuit's instruction to look at the specification in interpreting means-plus-function claims as "dicta."[182] The Federal Circuit reaffirmed *Iwahashi* in *In re Bond*.[183] The PTO followed with an official announcement that it would follow its own interpretation until either the Federal Circuit ruled en banc or the law was changed.[184]

The PTO's practice was to evaluate means-plus-function claims for section 101 purposes to cover every imaginable means (without looking at the specifications) unless the applicant could prove that the claims were drafted to a specific apparatus (i.e. not a general purpose computer).[185] For software patents, this often worked to transform means-plus-function language into pure method claims that were rejected as mathematical algorithms failing the second part of the *Freeman-Walter-Abele* inquiry. Field of use limitations, data-gathering steps, or the transformation of matter in post-solution activity would not save an application.[186] Twice PTO Board panels tried to render opinions contrary to the PTO official position. In both cases, the PTO commissioner withdrew the opinions

180 Lee E. Barrett, *Patentable Subject Matter: Mathematical Algorithms and Computer Programs*, 1106 Off. Gaz. Pat. & Trademark Office 5 (Sept. 5, 1989), *reprinted in* 38 Pat. Trademark & Copyright J. (BNA) No. 948, at 563 (Sept. 21, 1989).

181 *Id.* at 9.

182 James E. Denny, *Notice interpreting* In re Iwahashi, 1112 Off. Gaz. Pat. & Trademark Office 16 (Mar. 13, 1990), *reprinted in* 39 Pat. Trademark & Copyright J. (BNA) No. 399 (Mar. 15, 1990).

183 910 F.2d 831 (Fed. Cir. 1990).

184 *Notice of Application of 35 U.S.C. § 112 ¶ 6*, 1134 Off. Gaz. Pat. & Trademark Office 633 (Jan. 7, 1992). The Federal Circuit has a self-imposed policy of only overruling prior CCPA or Federal Circuit opinions in en banc hearings. As *Iwahashi* and *Bond* were three judge panel opinions, the PTO considered as good law the earlier case of In re Lundberg, 244 F.2d 543 (CCPA 1957). *See* Wesley W. Whitmyer, Jr., *The Patent and Trademark Office's Refusal to Follow* In re Bond, 74 J. PAT. & TRADEMARK OFF. SOC'Y 397 (1992).

185 *Notice of Application*, *supra* note 184.

186 *See supra* note 180, at 8-9.

before publication and appointed packed "super boards" to render the desired outcome.[187]

To avoid the costly and lengthy process of appealing all the way to the Federal Circuit, practitioners loaded patent claims with irrelevant references to hardware. Numerous how-to articles explained "tricks of the trade" and other substantively meaningless ways to get around PTO objections.[188] This year the Federal Circuit, sitting en banc, finally cleared up the heated dispute. The court stated:

> The plain and unambiguous meaning of paragraph six is that one construing means-plus-function language in a claim must look to the specification and interpret that language in light of the corresponding structure, material, or acts described therein, and equivalents thereof, to the extent that the specification provides such disclosure . . . . we hold that paragraph six applies regardless of the context in which the interpretation of means-plus-function language arises, i.e., whether as part of a

187 PTO Board of Appeals typically are three person panels of examiners-in-chief. Ex parte Akamatsu, 22 U.S.P.Q.2d 1915 (Bd. Pat. App. and Interf. 1992) (original panel dismissed and opinion discarded; newly appointed five person panel reached opposite decision); Ex parte Alappat, 23 U.S.P.Q.2d 1340 (Bd. Pat. App. and Interf. 1992) (panel enlarged to eight; original opinion became dissent in 5-3 ruling). *See* Daniel J. Kluth, *Mathematical Algorithms on Appeal: Recent Decisions by the Board of Patent Appeals*, COMPUTER LAW., Feb. 1993, at 12.

188 *See, e.g.*, DAVID PRESSMAN, PATENT IT YOURSELF 9/14 (1992); Stephen A. Becker, *Drafting Patent Applications on Computer-Implemented Inventions*, 4 HARV. J. OF LAW & TECH. 237 (1991); David S. Benyacar, *Mathematical Algorithm Patentability: Understanding the Confusion*, 19 RUTGERS COMPUTER & TECH. L.J. 129, 195 (1993); Michael A. Glenn. *Software, Patents, Alappat, and All That*, NEW MATTER, Summer 1993, at 23, 27 (publication of the State Bar of California); Stuart P. Meyer, *Obtaining and Enforcing Patents for Software-Related Inventions: Avoiding the Pitfalls*, 5 SOFTWARE L.J. 715 (1992); Rick D. Nydegger, *Practical and Legal Consideration in Drafting a U.S. Patent Application for Computer-Related Inventions*, 18 RUTGERS COMPUTER & TECH. L.J. 109, 113 (1992); Anthony L. Miele, *Drafting Claims for Patent Protection of Software/Computer-Related Inventions in the U.S. in order to Maximize the Scope of Protection*, 1 U. BALT. INTELL. PROP. L.J. 41 (1992); Stern, *supra* note 140, at App. A; Robert Greene Sterne, et al., *Preparing and Prosecuting Electronic and Computer Related Patent Applications: Avoiding and Overcoming Statutory Subject Matter Rejections*, 33 IDEA 297 (1993).

> patentability determination in the PTO or as part of a validity or infringement determination in a court.[189]

This decision, along with the appointment of PTO Commissioner Lehman, hopefully will return consistency to the practice of patent law before the Federal Circuit and PTO.

V. RECOMMENDATIONS

A. Eliminate Non-statutory Tests for Determining Subject Matter

At a minimum, no meaningful test can be constructed around whether a given piece of software is an algorithm or not. The loose use of "algorithm" in *Benson* left open the question whether any computer software could be patented. Why? Programmers love to describe what they do as writing algorithms. It is much like: attorneys don't fight, they litigate.[190]

An algorithm is a step-by-step method of solving a problem, and computer software is logic that handles input to generate output. Any program that performs data manipulation can be said to contain an algorithm. Soon you have a rule that does not exclude any logical possibilities, a tautology. To achieve intellectual rigidity, some observers borrow the etymology of the word from Donald Knuth's reference to the word "algorism" in classic Islamic mathematics.[191] That is like trying to use *The Canterbury Tales* to convey to the MTV generation the definition of "bad." The courts realized the futility of the task and started the march to definitional clarity with *Benson's* "procedure for solving a given type of mathematical problem . . . ."[192]

---

189 In re Donaldson, 29 U.S.P.Q.2d (BNA) 1845 (Fed. Cir. 1994). *See generally* Edward J. Webman, *The Controversy Over the Application of 35 U.S.C. § 112 ¶ 6 to Patentability Determinations*, 76 J. PAT. & TRADEMARK OFF. SOC'Y 47 (1994) (A patent examiner summarizes the contrary position taken by PTO's counsel at oral argument).

The *Alappat* case was also heard on appeal en banc by the Federal Circuit, but a decision has not yet been rendered.

190 In nearly a decade of experience programming all types of computers, it was my common experience to hear individuals loosely use "algorithm" to cover all but the most simple programming tasks. PC programmers do not use the phrase much—call it a healthy aversion to pretentious language.

191 Professor Donald E. Knuth of Stanford University is the author of the seminal work in theoretical computer science. He begins his classic with a reference to a Persian textbook author who used the word "algorism" in the year 825. DONALD KNUTH, 1 THE ART OF COMPUTER PROGRAMMING: FUNDAMENTAL ALGORITHMS 1 (2d ed. 1973). *See generally* CLAPES, *supra* note 2, at 106-07.

192 Gottschalk v. Benson, 409 U.S. 63, 65 (1972).

By itself, verbosity does not clarify. One of the leading and most prolific commentators in this field has said: "Defining the term 'mathematical algorithm' is a difficult task which does not appear to have yet been accomplished, at least to the satisfaction of many . . . . [A]ny further attempt in defining the term would be masochistic."[193] One computer industry professional has observed:

> It is amusing to hear some lawyers at PTO hearings spout knowledgeably about how software is *not* equivalent to mathematical algorithms and therefore should be freely patentable. Anyone who has the most basic familiarity with the work of Alan Turing or John McCarthy would get a good laugh out of this assertion. (Turing machines are an abstract formal system invented a decade before the first electronic computer; every computer program is mathematically equivalent to an instance of a Turing machine. John McCarthy invented the Lisp language in 1956 purely as a mathematical notation; it was a surprise to McCarthy when, several years later, one of his graduate students implemented a Lisp interpreter on a computer.)[194]

The *Iwahashi* court tried to reconcile the various definitions being used,[195] but was to see its attempt augmented with new twists by the Board in *Logan*.[196] In his *Arrhythmia* concurrence, Judge Rader decisively cut through the convoluted attempts to form a standard and concluded:

> When determining whether claims disclosing computer art or any other art describe patentable subject matter, this court must follow the terms of the statute. The Supreme Court has focused this court's inquiry on the statute, not on special rules for computer art or mathematical art or any other art . . . . To me, the Supreme Court's most recent message is clear: when

193 Irah H. Donner & Randall Beckers, *Throwing Out Baby* Benson *With the Bath Water: Proposing a New Test for Determining Statutory Subject Matter*, COMPUTER LAW., Jan. 1993, at 8, 14 n.7.

194 Ray Valdés, *Software Patents: You Make the Choice*, DR. DOBB'S DEVELOPER UPDATE, Apr. 1994, at 2, 4 (emphasis in original). Increasingly, the distinction between software and hardware is meaningless. "There is full functional equivalence among hardware, software, and firmware." 1 BENDER, *supra* note 14, at § 3.02[3]. *See generally* KENNETH L. SHORT, MICROPROCESSORS AND PROGRAMMED LOGIC 146-76 (1981); Tom Ochs, *A Clear Look Through Bleary Eyes at Two Books on Algorithms*, DR. DOBB'S JOURNAL OF SOFTWARE TOOLS, Apr. 1994, at 133.

195 In re Iwahashi, 888 F.2d 1370, 1374 (Fed. Cir. 1989).

196 Ex parte Logan, 20 U.S.P.Q.2d (BNA) 1465 (Bd. Pat. App. and Interf. 1991).

> all else fails (and the algorithm rule clearly has), consult the statute.[197]

After the *Iwahashi* decision, it appears that the artificial barriers placed in front of software patentability no longer exist provided the drafter describes some token hardware in the application (i.e. a means for storing the program in memory).[198] What's left of the mathematical algorithm test is that laws of nature, natural phenomena, and abstract ideas are not patentable—and they never were.

It is time we took the Supreme Court's direction in the *Diehr* and *Chakrabarty* decisions not to "read into the patent laws limitations and conditions which the legislature has not expressed."[199] $E=mc^2$ is not a method and certainly not an apparatus. It simply describes how nature works.[200] Inventions that implement the equation such as nuclear power stations or programs positioning magnets along a super-collider should be patentable as long as they use the equation in a novel way.[201] The phrase that "anything under the sun that is made by man" is patentable, quoted in most cases discussed in this Note, must be paid more than lip-service.[202] Patents should be awarded in new technological arts without regard to how unanticipated the advances might have been in 1954.

---

197 Arrhythmia Research Tech. v. Corazonix Corp., 958 F.2d 1053, 1066 (1992) (Rader, J., concurring). The former head of the U.S. Justice Department's Intellectual Property Section, who was one of the draftsmen of the Johnson Administration legislation and counsel in the *Benson* and *Flook*, cases has stated:

> I now wonder whether that effort was misguided, indeed ill-conceived as a matter of policy. First of all, it has turned out to be a King-Canute-and-the-tide process. Second, we have ended up with the courts often treating computer program copyrights as if they were patents, which makes even less sense than issuing patents via the patent office and after examination for novelty and technical merit.

Stern, *supra* note 140, sec. VI.

198 *See supra* notes 176-77 and accompanying text.

199 *See supra* note 169 and accompanying text.

200 Chief Justice Burger used the example in *Chakrabarty*. "Einstein could not patent his celebrated law $E=mc^2$ . . . . Such discoveries are 'manifestations of . . . nature, free to all men and reserved exclusively to none.'" Diamond v. Chakrabarty, 447 U.S. 303, 309 (1980) (citation omitted).

201 *See also* Benyacar, *supra* note 188, at 131 (describes Karmarkar's algorithm, a "breakthrough" method at solving linear programming problems more efficiently, that was developed at Bell Labs).

202 The phrase is used so pervasively in cases discussing § 101 that its point is getting dulled through mere repetition. *See, e.g.*, Diamond v. Diehr, 450 U.S. 175, 182 (1981); Arrhythmia Research Tech. v. Corazonix Corp., 958 F.2d 1053, 1056 (1992).

The PTO considers the term "software patents" to be a misnomer since patents protect processes implemented using software or systems capable of performing certain functions.[203] In practice, the program code is often the measure of the process or the system when run on a personal computer.[204] In the PTO patent classification scheme, classes 364 (Data Processing Systems) and 395 (Information Processing Systems) are where most software-related patents are listed. Recent statistics for these two classes indicate that the process is becoming more open.[205]

	PATENTS APPLIED FOR	PATENTS ISSUED
1987	3,270	1,174
1988	3,829	1,908
1989	5,444	2,858
1990	6,555	2,591
1991	6,600	2,647
1992	7,552	2,830
1993	8,391	3,613

In addition, it is estimated that there are approximately 10,000 applications backlogged in the PTO examining group that reviews these patents.[206]

---

203 FINDING A BALANCE, *supra* note 25, at 6 n.19, 132. The preferred alternative phrase is software-related patents.

204 Consider this story from a former patent examiner and current law student:

> I was once confronted with a computer program claim directed to a mathematical algorithm that made no reference to any hardware elements. The claim recited a method of interpreting data from a bar-code reader in a way that would minimize errors . . . . I directed the inventor to amend the claim to change all occurrences of "method" to "computer-implemented method." With these minor changes, the claim was allowed.

Kenton R. Mullins, *An Interpretive Model for Meeting the § 101 Requirement with § 112 Considerations*, COMPUTER LAW., Oct. 1993, at 23. The practicing bar reports that § 101 rejections are not applied in a consistent manner. *See supra* note 188.

205 Figures are for fiscal years Oct. 1 to Sept. 30. The full names of the two classes are: Class 364-Electrical/Computers and Data Processing Systems, and Class 395-Information Processing Systems Organization. Class 395 was created in 1991 and replaced certain art areas from 364. *Software-Related Patent Activity*, SPI REPORTER, Fall 1993, at 5 (additional statistics were provided to the author by the SPI). Some additional software-based inventions can be found in: Class 371-Error Detection, Correction and Fault Recovery, and Class 340-Displays and Communications.

206 Steve Hamm, *Patented Problems: U.S. Patent and Trademark Office Responding to Criticism*, PC WK., Jan. 24, 1994. Where the goal is to process the average application in eighteen months, many software applications are taking four to five years. *Id.* Since 1990, IBM alone has been awarded over 500 software-related patents that cover 3.5 percent of

B. Require Source Code to be Appendixed to Patent Applications

The practice in the software arts is to provide source code when publishing an article or other work on a new algorithm or programming technique. Standard industry publications such as *Dr. Dobb's Journal of Software Tools, Microsoft Systems Journal,* and *PC Magazine* provide complete source code listings for all but the most lengthy examples in the text of the periodical. These publications also give readers the ability to download the full-text of all examples from on-line services.[207] The *sine qua non* of getting published is demonstrating that what the author is propounding actually works. Programming is not like medical research found in the *New England Journal of Medicine.* Readers expect to be able to replicate and test the examples since distributing the code is a trivial task.

A software patent without source code is like a law review piece filled with case names but missing citations to case reporters. A person of ordinary skill in legal research might be able to track down the full-text of all the opinions. *Marbury v. Madison*[208] would be found quicker than a state trial court opinion. But, would anyone think that such a practice was enabling or the best mode? As it is now, the disclosure requirements can be met using such devices as specifications, flowcharts,[209] and pseudo-code.[210]

---

its computer programs. Aharonian, *supra* note 15 (testimony of Victor Siber, IBM Senior Counsel, at San Jose hearings); Whitmer, *supra* note 15, at 20. *See also supra* note 13.

207 Whether a publication provides source code depends to a great deal on where it falls on the spectrum from a practitioner's aide to theoretical musings. For example, different publications in the IEEE TRANSACTIONS series will give varying levels of source code disclosure. Patents are not theoretical. They must show actual or constructive reduction to practice. Hazeltine Corp. v. United States, 820 F.2d 1190 (Fed. Cir. 1987) (filing a patent application as constructive reduction to practice); Newkirk v. Lulejian, 825 F.2d 1581 (Fed. Cir. 1987) (requirements for actual reduction to practice require a demonstration that embodiment actually worked for its intended purpose).

208 5 U.S. (1 Cranch) 137 (1803).

209 Flowcharts are a way of using inter-connected geometric shapes to represent program flow. One can even find ISO and ANSI standards on their proper usage. *See* International Organization for Standardization, International Standard 1028—Information Processing-Flowchart Symbols; American National Standard, Flowchart Symbols and Their Usage ANSI X3.5-1970. In practice, flowcharts are little used outside of bureaucratic organizations. Even then, they are often done after the work is done to satisfy some documentation requirement. *Contra* Paul Winsberg & Daniel Richards, *Data modelling isn't Dead,* COMPUTERWORLD, Apr. 4, 1994, at 85.

The Department of Defense requires an extensive multi-step structured design methodology that utilizes flowcharts. *See* Richard Armstrong Beutel, *Software Engineering Practices and the Idea/Expression Dichotomy: Can Structured Design Methodologies Define the Scope of Soft-*

Professor Randall Davis of MIT summed it up at the National Research Counsel in 1990:

> There is almost no way to visualize software. Sure, we have flow charts, we have data-flow diagrams, we have control flow diagrams, and everybody knows how basically useless those are. Flow charts are documentation you write afterward—because management requires them, not because they are a useful tool.[211]

A patent is most similar to a real property deed specifying the metes and bounds for a parcel of land. Both documents are not easily understood but succeed if they secure the owners' interests in the specified claims. If the goal is to inform the world of an invention, software professionals have avenues more timely and less expensive than pursuing a patent application. In fostering the trade-off between the interests of inventors and the public, the source code is the best way to explain an algorithm.[212]

Under this proposal, a computer system's complete source code would not have to be appendixed to the patent. The applicant would only have to include the source code directly relevant to enabling the claim language. In cases where claims are broadly written (as in a means-plus-function apparatus claim that covers

---

ware Copyright?, 32 JURIMETRICS 1, 8 (1991) (Discusses DoD-STD-2167A Software Development Methodology). *Contra* CRINGELY, *supra* note 137, at 27-28 (Gifted programmers envision program flow just as master-level chess player can "see" how a game is going to develop.); Robert Kelley & Janet Caplan, *How Bell Labs Creates Star Performers*, HARV. BUS. REV., July/Aug. 1993, at 128.

210 Pseudo-code of P-code is an cross between source code and standard English. It is used in the attempt to explain an algorithm in a manner not dependent on knowledge of a specific computer programming language. The attempt often is not worth the effort. In the 1970's p-code was thinly disguised FORTRAN; today it is C code.

211 This statement was made at a forum on the state of software development. *quoted in* CLAPES, *supra* note 2, at 10.

212 In my days of supervising programmers, I preferred to see the raw source code stripped of any documentation. As long as the variable names made sense and the code was properly indented, I could plow through source code far more quickly than the text of a newspaper article. *See* BRIAN W. KERNIGHAN & P.J. PLAUGER, THE ELEMENTS OF PROGRAMMING STYLE 135-37 (1974) ("Summary of Rules . . . Don't comment bad code—rewrite it. Use variable names that mean something. Use statement labels that mean something. Format a program to help the reader understand it. Document your data layouts. Don't over-comment.").

the automation of an entire industry), a nearly complete program listing would be required.[213]

Occasionally, the disclosure might reveal what is called "spaghetti code"—that is source code so convoluted and unclear that it makes no sense. This problem is not unique to programming. We all can recall one-or-two instances of stilted and awkward legal writing that could be clarified. The intentional rearrangement of a program into "spaghetti code" would make the patent invalid by violating the best mode requirement.

A rule change requiring the attachment of source code could easily be made. The PTO already has rules governing the permissive submission of program code.[214] This rule-change should only apply prospectively.

C. *Computer Assisted Methods of Doing Business are not Per Se Unpatentable*

Computers automate many existing business practices. An uncomfortable fact is that the cost of developing and implementing these systems is routinely measured against anticipated labor savings.[215] Thousands of new systems begin operation each week

---

213 As with any general rule, exceptions would need to be carved out. Applicants should have the ability to black out sections that cover proprietary information and trade secrets irrelevant to the patent's claims. For instance, numerical values used in computing mark-ups could be redacted from a patent for an inventory control system. *See, e.g.*, 2 BENDER, *supra* note 14, at § 4B.03 (description of inventory control system). In addition, the source code would not limit the invention. *See supra* note 84.

214 37 C.F.R. § 1.96 Submission of computer program listings (1993). If the submission is 10 pages or less, it must be contained in either the drawings or as part of the specification. For longer programs, the print-out must be appended on microfiche that will not become part of the printed patent. Such appendices are available for purchase with the file wrapper once the patent issues or the application becomes otherwise public. According to the rule:

> The program listing may be either in machine or machine-independent (object or source) language which will cause a computer to perform a desired procedure or task such as solve a problem, regulate the flow of work in a task such as solve a problem, regulate the flow of work in a computer, or control or monitor events.

Id. Source code submitted would still be protected by the copyright laws but would lose any trade secret protection it enjoyed.

The current rule's allowance for object (machine) code submissions serves no purpose. Object code is the executable module created after a program is compiled. *See supra* note 100. In order to be understandable by a person, object code must be reverse engineered into source code through the laborious process of decompilation. *See* Andy Johnson-Laird, *Technical Demonstration of "Decompilation,"* 16 COMPUTER L. REP. 469 (1992).

215 SHOSHANA ZUBOFF, IN THE AGE OF THE SMART MACHINE 124-73 (1988).

with many not living up to their advanced billing.[216] In most large companies, a multi-year backlog of requested systems waits for someone to work on them.[217]

Although trade journals regularly feature "first-of-its-kind" systems, they are not necessarily patentable.[218] Every other year, consultants seem to uncover a newest, greatest tool to boost programmer productivity. The mere fact that a company was the first to use 4GL's,[219] RAD,[220] CASE,[221] or OOP[222] in its line-of-business should not be the basis for granting a patent. As noted, an invention must be novel, useful, and non-obvious.[223] With the proliferation of tools, it is not easy to distinguish the new from the merely repackaged.

For example, consider automation in the brokerage industry. A system that merely computerized the manual stock record function would not be patentable. Extensive documentation exists as prior art on how to accomplish the task.[224] However, a system that significantly cut down the number of steps and expense involved in processing stock records might be patentable. Contrast that with emerging systems, based on artificial intelligence, used to

---

216 *See, e.g.*, Gene Hall, et al., *How to Make Reengineering Really Work*, HARV. BUS. REV., Nov./Dec. 1993, at 119; Cornelius H. Sullivan, Jr., *Systems Planning in the Information Age*, *in* THE STRATEGIC USE OF INFORMATION TECHNOLOGY 118 (Stuart E. Madnick ed., 1987).

217 Evelyn Richards, *Society's Demands Push Software to Upper Limits; More Computer Crises Likely*, WASH. POST, Dec. 9, 1990, at A1.

218 Some trade journals that regularly include articles on new systems include: COMPUTERWORLD, DATAMATION, and WALL STREET COMPUTER REVIEW.

219 Fourth-generation language. Term used most frequently to refer to relational database languages that brought relief from the torture of coding in COBOL.

220 Rapid application development. *See* Robert A. DelRossi, *RAD: Rapid Application development tools put wheels under your feet but don't let you walk on air*, INFOWORLD, Feb. 14, 1994, at 62.

221 Computer-aided software engineering. *See generally Case findings*, COMPUTERWORLD, Apr. 11, 1994, at 76 (survey of CASE tool usage in corporate information service departments).

222 Object-orientated programming. It appears that OOP will be the dominant programming environment for the next few years. *See* David M. Barkan, *Software Litigation in the Year 2000: The Effect of Object-Orientated Design Methodologies on Traditional Software Jurisprudence*, 7 HIGH TECH. L.J. 315 (1992) (Explores the challenge of adapting intellectual property laws to handle software developed using OOP). Sheryl Canter, *C++ New Package, New Power*, PC MAG., Mar. 29, 1994, at 185 (Reviews the popular C++ programming environments which have OOP extensions.).

223 *See supra* notes 134-36 and accompanying text.

224 DAVID W. WEISS, AFTER THE TRADE IS MADE: PROCESSING SECURITIES TRANSACTIONS 95 (1986). In addition, software automating this task has been around since the 1960's.

create financial derivatives and diagnose medical ailments.[225] As pioneering inventions, a broader range of patentability should be given to these new technologies.[226]

Patents for stock record or financial derivative systems might be more difficult to obtain than one for a medical diagnostic system. Why? The *Notice of Public Hearings* for the PTO's recent hearings included the statement:

> There are three general categories of exclusions to patent eligibility that are particularly relevant to software-related inventions . . . . Second, methods of doing business are excluded from protection. While no cases have directly applied this exclusion to deny patent protection for software-related inventions, the exclusion is relevant for questioning the patent eligibility of processes that are modeled upon existing business processes but are implemented through a software-based system.[227]

The problem with this statement is that it is doubtful that a method of doing business exclusion even exists. Older cases decided by Article III courts include dicta referring to the doctrine, but no cases were decided on the issue.[228] On the two occasions courts have been asked to apply this so-called exclusion to software, they have summarily rejected the argument.[229]

Given the direction the law has taken with the algorithm test in *Diehr* and *Arrhythmia,* it is unlikely that a method of doing busi-

---

225 Barnaby J. Feder, *Sophisticated Software Set for Exotic Financial Trades,* N.Y. TIMES, Mar. 30, 1994, at D1.

226 *See* Laitram Corp. v. Cambridge Wire Cloth Co., 863 F.2d 855 (Fed. Cir. 1988) (scope of protection for pioneering inventions); Slimfold Mfg. Co. v. Kinkead Indus., 932 F.2d 1453 (Fed. Cir. 1991) (narrow scope of equivalents for invention in a crowded field).

227 *See supra* note 14 (the first exclusion was the mathematical algorithms test; the third was for printed matter).

228 E. Robert Yoches & Howard G. Pollack, *Is the "Method of Doing Business" Rejection Bankrupt?,* 3 FED. CIR. B.J. 73 (1993) (Provides a thorough review of case law and reveals an absence of any Article III court decision that used so-called exclusion as basis for decision.).

229 In re Johnston, 502 F.2d 765, 771 (CCPA 1974), *rev'd on other grounds,* Dann v. Johnston, 425 U.S. 219 (1976); Paine, Webber, Jackson & Curtis, Inc. v. Merrill Lynch, Pierce, Fenner & Smith, Inc., 564 F.Supp. 1358, 1369 (D. Del. 1983). *See also* CLAPES, *supra* note 2, at 110-12.

On one occasion, a PTO Board of Examiners panel used the method of doing business exclusion to deny a software patent using dicta from two CCPA cases. Ex parte Murray, 9 U.S.P.Q.2d (BNA) 1819 (Bd. Patent App. and Interf. 1988). Reference to doctrine is also found in a paragraph commenting on MPEP § 706.03(a)-Nonstatutory Subject Matter. The PTO should delete this paragraph.

ness rejection would be upheld on appeal. However, a legitimate objection to processing applications for business systems is the difficulty encountered when researching the prior art. Patent agents and attorneys have a good faith duty to disclose the relevant prior art in patent applications.[230]

D. Finding a Needle in a Haystack: Providing Easier Access to the Prior Art

The legacy of the historical hostility to software patents is that prior art cannot easily be found. For the majority of software inventions, patent protection was never sought. Where software patents have been awarded, the high-art of disguising the software component has made these issued patents invisible to all but the most skilled searchers.[231] Since 1991, the PTO has begun to address this problem by instituting a classification category specifically for software-related inventions and hiring computer science graduates into the examining corps.[232]

---

230 MPEP § 2001 Duty of Disclosure, Candor, and Good Faith; MPEP § 2202 Citation of Prior Art.

231 That other fields of invention face similar difficulties is not an acceptable excuse. Credit is due to David R. Syrowik, Chairperson of the Michigan A.B.A. Computer Law Section's Proprietary Rights Committee, who took on the task of preparing surveys of issued software patents long before commercial publishers stepped in to fill the void. *See, e.g.*, PROPRIETARY RIGHTS COMMITTEE, STATE BAR OF MICHIGAN, SURVEY OF U.S. SOFTWARE PATENTS: POST-Diehr THROUGH DECEMBER 1990 (1991); FOREIGN SOFTWARE PATENTS (1990); THE ANATOMY OF A SOFTWARE PATENT (1989); SURVEY OF UNITED STATES SOFTWARE PATENTS ISSUED FROM JULY 1987 THROUGH DECEMBER 1987 (1988). Portions of these reports can be found in 1 BENDER, *supra* note 14, at § 3A.13, App. 3A[4]-[6].

232 *See supra* note 205. Commissioner Lehman has stated:

> The patent office has not been very well prepared to deal with the issue of software patents. We did not have patent examiners skilled in this area. There is a unique aspect to the software industry, compared to other high-tech industries such as biotech. In other sectors, there is a strong database of prior art, which is absolutely essential to patent examination. We still don't have a database of prior art for software, and the industry has historically relied on trade secrecy. The combination of all these factors has resulted in a mess. It is pretty clear that we issued patents that will not he held valid in court. We want to get to the bottom of it and fix this.

Quoted in Software Patents: Into the Breach Again, DR. DOBB'S DEVELOPER UPDATE, Apr. 1994, at 1.

For patents that do get litigated, practitioners might want to investigate the standing order "Additional Disclosures Required for Patent Cases" that Judge Fern Smith drafted as a result of hearing Atari Games Corp. v. Nintendo of Am., 1993 U.S. Dist. LEXIS 8183 (N.D. Cal. Apr. 15, 1993).

One point of interest is that the Patent Information Clearinghouse in Silicon Valley has over 50,000 visitors a year.[233] A new initiative receiving broad support is the work of the Software Patent Institute (SPI).[234] Funded by most of the major software publishers, the SPI seeks to create a database of software techniques with an emphasis on those that have not been patented and are not otherwise available.[235] They will also offer educational services directed to patent examiners and offer some short courses to the public.[236] Also, computer books have been a popular category in the trade press for over a decade. A number of commercial publishers have started to gather some of this material and have offered compilations available on CD-ROM and other media.[237]

V. CONCLUSION

In the twenty-two years since the *Benson* decision, both the computer industry and patent law have come a long way. Next year will mark the twentieth anniversary of the Altair, the world's first PC.[238] Software now dwarfs hardware in terms of public attention, sales, and employment.

In the legal field, *Benson* has been so limited that a decision explicitly overruling it would not work a great change in the substantive law. In addition, the Federal Circuit has created a forum where difficult scientific cases are knowledgeably treated with considerable insight. The last organization trying to keep time still has

---

233 The clearinghouse is a branch of the public library system of Sunnyvale. Leland Joachim, *Sunnyvale's Patent Library Services May be Enhanced*, SAN JOSE MERCURY NEWS, Mar. 16, 1994, at ext.2 p.1.

234 *See, e.g.*, T.C. Doyle, *Support keeps growing for Software Patent Institute—Organization could be good resource to settle disputes*, COMPUTER RESELLER NEWS, Feb. 7, 1994, at 99.

235 *See generally* SPI REP., Fall 1993. The SPI has been formed within the Industrial Technology Institute under Professor Bernard A. Galler's direction at the University of Michigan. Their database will focus on software manuals, IBM Technical Disclosure Bulletins, and documents not otherwise available from a public source. Submissions from the public will also be accepted in the form of defensive disclosures. No source code will be available. A purpose for the SPI database will be to serve as a repository for prior art that has been "described in a printed publication in this or a foreign country." Prior publication is a defense in an infringement action available under 35 U.S.C. § 102 (a) & (b).

236 *Id.*

237 For example, Source Translation & Optimization has an extensive database of source code, and Rapid Patent offers a CD-ROM compilation of software-based patents. One estimate places general computer book sales running at two to three million copies a year. L.R. Shannon, *The 6-Mile Shelf of Computer Books*, N.Y. TIMES, Apr. 19, 1994, at C9.

238 FREIBERGER & SWAINE, *supra* note 46, at 27-53.

been the PTO. This too is changing. PTO Commissioner Lehman's actions so far have demonstrated an openness to public and industry interests.

The algorithm test for statutory subject matter has failed and remains merely as a barrier that patent applicants easily maneuver around. Alternative non-statutory tests, such as the method of doing business exception, should not be allowed to burrow their way into the law. The proper focus of subject matter inquires is on the Supreme Court's mandate that protected inventions include "anything under the sun that is made by man."[239] Public disclosure of patented inventions can be easily fostered through source code submissions and more accessible methods of searching the prior art. It is time for the patent system to fully embrace computer software and abandon special case rules limiting protection. Software patent applications should no longer automatically "ABEND" on receipt at the PTO.[240]

Thomas P. Burke

239 *See supra* notes 168, 197 & 202.

240 "ABEND" stands for abnormal end. A computer program that crashes or fails to run to its expected end is said to abend. A little debugging will usually fix the problem.