Exporting Software Components—Finding A Role for Software in 35 U.S.C. § 271(f) Extraterritorial Patent Infringement*

TABLE OF CONTENTS

* J.D. Candidate 2005, University of San Diego School of Law; B.S.E.E. 1986, Drexel University.

This Comment is dedicated to my daughters Katherine and Julia, who keep me focused on what is really important in life. Special thanks are owed to several people for making this work possible. First and foremost, I want to thank my wife and best friend, Renee Tietsworth, for her constant support, encouragement, and inspiration. I also want to thank Tonya Cross, my friend and extraordinary comment advisor, for her motivation and thoughtful feedback, and Professor Jane Henning for her insightful teaching of intellectual property law and help as my faculty advisor. Thanks also to my colleagues at the *San Diego Law Review* for pushing me to do my best. Finally, I want to thank my parents for their unlimited love and support.

> [I]t is one of the surest indexes of a mature and developed jurisprudence not to make a fortress out of the dictionary; but to remember that statutes always have some purpose or object to accomplish, whose sympathetic and imaginative discovery is the surest guide to their meaning.
>
> Learned Hand[1]

I. INTRODUCTION

For over two hundred years the American patent system has fostered innovation by providing a limited monopoly on new inventions in exchange for full disclosure.[2] Recent innovations in areas unimagined by previous generations have strained the patent system's ability to accommodate new technologies within the existing legal framework. This Comment explores an example of this problem involving software and proposes a framework for integration of software into a patent system oriented around tangible, physical innovations.

On August 11, 2003, a small Illinois technology company, Eolas Technologies, was awarded the staggering sum of $520.6 million in damages against Microsoft.[3] This enormous damage award was granted by a jury in the federal district court for the Northern District of Illinois[4] for an increasingly litigated area[5]: patent infringement.[6] Eolas successfully

1. Cabell v. Markham, 148 F.2d 737, 739 (2d Cir. 1945), *aff'd*, 326 U.S. 404 (1945).

2. This quid pro quo forms the basis for the patent system; while individual patentability requirements—as well as the duration and scope of protection—have varied over the years, the basic requirement of disclosure in exchange for protection has remained. *See generally* FRED WARSHOFSKY, THE PATENT WARS (1994) (discussing historical changes in the scope of patent protection).

3. Joseph Menn, *Microsoft Loses UC Patent Case*, L.A. TIMES, Aug. 13, 2003, at C1. Despite the size of this award, things could have turned out worse for Microsoft had the jury granted the plaintiff's request for damages of $1.2 billion. John Markoff, *Jury Rules Against Microsoft in Patent Case*, N.Y. TIMES, Aug. 12, 2003, at C2.

4. Patent protection is an exclusive federal right and Congress limits jurisdiction over patent law issues to the federal district courts, providing that "[t]he district courts shall have original jurisdiction of any civil action . . . relating to patents . . . ." 28 U.S.C. § 1338(a) (2000). In addition, appeals from the federal district courts in patent cases are under the exclusive jurisdiction of the United States Court of Appeals for the Federal Circuit rather than the federal circuit courts. *See* 28 U.S.C. § 1295 (2000).

5. According to one source, the number of patent lawsuits rose by thirty percent from 1997 to 2002. Michelle Kessler, *The 521-Million-Dollar Man*, USA TODAY, Sept. 30, 2003, at 3B. The high-tech boom of the 1990s, which was heavily focused on Internet and software technologies, has led to "an onslaught of patent infringement

alleged that Microsoft had improperly incorporated its patented technology into the Windows operating system and Microsoft's Internet Explorer web browser.[7] The result of this ongoing[8] David versus Goliath[9] battle is the second largest patent infringement verdict in history[10] and the first significant patent victory against Microsoft in nearly ten years.[11]

---

cases." Amalia Deligiannis, *Browser Dispute Costs Microsoft $521 Million*, CORP. LEGAL TIMES, Oct. 2003, at 52. Patent industry consultant James J. Nawrocki, who has traced the history of patent litigation for the last twenty years, notes that "[w]e continue to see an increasing number of U.S. patent applications and grants, and a corresponding trend [increase] in patent litigation . . . . By gathering and updating this information on a regular basis, we hope to make legal professionals more aware of the surging tide of patent litigation and its serious financial implications to a company's bottom line." *Top Five Patent Damage Awards in First Half of 2002 Exceed $100M*, DAILY REC. OF ROCHESTER, Aug. 8, 2002, LEXIS, News Library, Drecro File.

6. Patent infringement generally involves making or using another's patented innovation. *See* 35 U.S.C. § 271 (2000). Infringement is discussed in detail later in this Comment. *See infra* notes 84–97 and accompanying text.

7. Menn, *supra* note 3, at C1.

8. The case was originally filed by Eolas in 1999. Deligiannis, *supra* note 5, at 52. Microsoft has already asked for rehearing of some aspects of the case, including a charge of inequitable conduct, which involves failure to fully disclose information to the patent office during the patent application process (known as prosecution in the patent trade). *See* Eolas Techs., Inc. v. Microsoft Corp., No. 99 C 626, 2003 WL 22078029, at *1 (N.D. Ill. Sept. 3, 2003) (detailing claim of breach by inequitable conduct during patent prosecution). Based on Microsoft's statements of intent to appeal various aspects of the case, including the patent's validity, it is likely the case will end up at the Court of Appeals for the Federal Circuit which hears all patent appeals. *See supra* note 4.

9. This case pits technology giant Microsoft against Eolas Technologies, a company whose sole employee is its founder and chief executive Michael Doyle. Barbara Rose, *Inventor Wins Patent Suit Against Microsoft*, CHI. TRIB., Aug. 12, 2003, at C1, LEXIS, News Library, Chtrib File. Doyle is, however, joined in this case by the University of California, an assignee of the patent. The University joined Doyle in the lawsuit after Microsoft alleged it could not proceed alone because Doyle had been a professor at the University in 1994 when he made his patented invention. Nevertheless, Doyle has held exclusive rights to the invention since 1994. Deligiannis, *supra* note 5, at 52; *see also* Steven M. Cherry, *David and Goliath Versus Another David*, IEEE SPECTRUM, Dec. 2003, at 47 (summarizing the history of the case and suggesting that the final award could go as high as $1.2 billion).

10. Barbara Rose, *Microsoft Friend, Little Guy the Foe; Web Groups Join Giant in Fight to Overturn Patent*, CHI. TRIB., Oct. 5, 2003, at C1, LEXIS, News Library, Chtrib File. The largest patent damage award in U.S. history was $873,158,971 in a landmark 1991 case between Polaroid and Eastman Kodak. *Top Five Patent Damage Awards in First Half of 2002 Exceed $100M*, *supra* note 5. Polaroid was awarded these damages for Kodak's infringement of its instant camera technology. *See* Polaroid Corp. v. Eastman Kodak Co., 867 F.2d 1415, 1416 (Fed. Cir. 1989) (discussing Polaroid's suit against Kodak for infringement of ten patents related to its instant camera and film system).

11. Microsoft's most recent loss in a patent case goes back to its highly publicized

The basic elements of *Eolas* are straightforward.[12] Microsoft developed and sold, typically in conjunction with its popular Windows operating system, the Internet web browser Internet Explorer.[13] Eolas successfully claimed that Microsoft incorporated technology covered by its patent[14] into the web browser and then sold and exported the protected software without authorization, resulting in several forms of patent infringement liability.[15]

Just two weeks after the *Eolas* verdict, the same infringement issues surfaced again in a second case, also involving Microsoft, under the same statutory provisions.[16] In this second case, another software company, Imagexpo, L.L.C., contended that Microsoft infringed its patented NetMeeting software in a similar way.[17]

Aside from sensationalism due to the prominence of the defendant and the amount of damages involved, both of these cases raise a number of interesting legal issues[18] related to software patents as well as extraterritorial

---

1994 case against Stac Electronics in San Diego. A jury awarded Stac $120 million for Microsoft's infringement of its patented hard disk compression technology. As a consequence, Microsoft had to pull its highly successful NT operating system from the market to remove the infringing code. Maureen O'Gara, *Eolas Clips Microsoft for $521m*, CLIENT SERVER NEWS, Aug. 18, 2003, at 1, LEXIS, News Library, Allnws File.

12. The trial jury found that Microsoft had infringed a U.S. patent held by Michael Doyle that claims a means for sending software applications over the Internet to a user's computer. Markoff, *supra* note 3, at C2.

13. A web browser is an application program used to access and display information stored on other computers connected to the Internet. A browser displays information formatted in an appropriate language such as hypertext markup language (HTML) or other standard file formats such as PDF, JPEG, or DOC. Web browser plugins allow files in specialized formats to be displayed in the browser. *See* Webopedia Computer Dictionary, Plug-in, *at* http://www.webopedia.com/TERM/p/plug_in.html (last modified May 17, 2004); *What's a Plug-in?*, Netscape Brower Central, *at* http://channels.netscape.com/ns/browsers/plugins.jsp (last visited Feb. 9, 2005).

14. The patent at issue is described as a "[d]istributed hypermedia method for automatically invoking external application providing interaction and display of embedded objects within a hypermedia document." U.S. Patent No. 5,838,906 (issued Nov. 17, 1998).

15. Sale of a product that incorporates another's patented technology, without authorization, is perhaps the most blatant and obvious form of patent infringement as statutorily defined. *See* 35 U.S.C. § 271(a) (2000). Patent infringement can, however, occur in a number of other ways, including by extraterritorial manufacturing that would not otherwise be prohibited. *See, e.g.*, 35 U.S.C. § 271(f) (2000).

16. Imagexpo, L.L.C. v. Microsoft Corp., 299 F. Supp. 2d 550 (E.D. Va. 2003).

17. *Imagexpo*, 299 F. Supp. 2d at 552 (involving a similar claim that Microsoft supplied Golden Masters for overseas manufacture, implicating § 271(f) liability).

18. Beyond the specific legal issues in question, at least one commentator suggests that *Eolas* threatens the viability of the Internet itself, classifying it as "[t]he most important patent case in the history of the Internet." Steven M. Cherry, *Eolas vs Web*, IEEE SPECTRUM, Jan. 2004, at 85. Eolas's patent covers methods for invoking other software from a web browser. This includes automatic invocation of programs to display text files in forms such as portable document format (PDF), or media players to display movies, sounds, or graphics. Because these applications have become so pervasive, Tim

patent infringement.[19] Some of these issues are hesitantly addressed by both District Courts for the first time.[20] In imposing liability in *Eolas*, the Court first looked to Microsoft's manufacture and sale of the infringing software in the United States, a relatively straightforward application of direct patent infringement.[21] Going beyond domestic product sales, however, the *Eolas* court also allowed imposition of liability for international sales where the product was assembled abroad from components of the patented invention exported from the United States, including Microsoft's allegedly infringing Windows operating system.[22] Liability under these extraterritorial provisions, codified in 35 U.S.C. § 271(f), can be imposed even though the components themselves do not enjoy any patent protection.[23] This provision is very recent in patent law development, having been in place for only twenty years.[24] During this time, software

---

Berners-Lee, inventor of the Web and head of the World Wide Web Consortium, wrote the patent office requesting a reconsideration of Eolas's patent to "prevent substantial . . . damage to the operation of [the] World Wide Web." Cherry, *supra* note 9, at 47 (quoting Tim Berners-Lee's letter to the head of the U.S. Patent and Trademark Office). As a consequence of this reconsideration, Microsoft claims that the patent office may "come to its rescue" and cancel Eolas's patent, making the infringement issue moot. *Microsoft Says U.S. May Cancel Patent*, L.A. TIMES, Jan. 30, 2004, at C3.

19. Extraterritorial infringement involves activities outside the United States that impose liability under a United States patent, which traditionally provides protection only within national geographic boundaries. *See* 35 U.S.C. § 100(c) (2000) (defining the "United States" and "this country" as "the United States of America, its territories and possessions"). Extraterritorial infringement liability has been greatly expanded within the last twenty years and liability can now be imposed for exporting components of a patented invention or importing a product made abroad from a patented process, even if the product itself is not protected by a patent. *See, e.g.*, 35 U.S.C. § 271(f–g) (2000).

20. The District Court in Illinois, in a Memorandum Opinion and Order allowing imposition of liability for extraterritorial infringement activities by Microsoft, notes that the issue of software component export has not previously been addressed. Eolas Techs., Inc. v. Microsoft Corp., 274 F. Supp. 2d 972, 973 (N.D. Ill. 2003). The District Court in Virginia cited one earlier software case involving a process patent that rejected liability under § 271(f) for a process patent. *Imagexpo*, 299 F. Supp. 2d at 552 (citing Enpat, Inc. v. Microsoft Corp., 6 F. Supp. 2d 537, 539 (E.D. Va. 1998)).

21. Direct infringement involves, among other activities, making or selling of another's patented invention within the United States while the patent is still active. *See* 35 U.S.C. § 271(a).

22. *Eolas Techs., Inc.*, 274 F. Supp. 2d at 974. The jury was allowed to impose liability under 35 U.S.C. § 271(f) for export of a "golden master," a master copy of the Windows operating system that is subsequently loaded onto computer hard disks during final computer assembly abroad. *Id.* at 973.

23. Liability is imposed merely for supplying components of a patented invention, for overseas assembly, with knowledge that assembly would be an infringement if done in the United States. *See* 35 U.S.C. § 271(f).

24. Patent law has a long history in the United States, being inherited by the

has only been considered a potential component in one case, where it was ultimately rejected.[25] By raising the issue again, *Eolas* and *Imagexpo* suggest that software might now receive serious attention as a component for liability purposes under 35 U.S.C. § 271(f). This raises a number of concerns about the unique characteristics of software. These include whether software, due to its intangible nature, should be considered a component at all for purposes of extraterritorial infringement. If so, what constitutes a software component for liability purposes, and what activities constitute export or supply of these components for liability purposes?[26] As a consequence of the visibility of these cases, these questions are likely to garner much attention.[27] Moreover, the rapid increase in global product markets, growth in both domestic and overseas software development, and the increasing trend towards outsourced offshore manufacturing make extraterritorial infringement under these provisions much more likely.[28] In addition, the growth of the Internet and its use as a channel for software distribution and execution will undoubtedly raise concern about the boundaries of export channels for liability purposes.[29]

---

colonies from England. Moreover, many of the basic concepts, such as novelty and territorial limitations, have remained the same. *See generally* ROGER E. SCHECHTER & JOHN R. THOMAS, INTELLECTUAL PROPERTY: THE LAW OF COPYRIGHTS, PATENTS AND TRADEMARKS 284–90 (2003).

25. *Enpat*, 6 F. Supp. 2d at 539.

26. The statutory provisions of 35 U.S.C. § 271(f) impose liability for supplying a "component" or "components," without clarifying what constitutes a component. Liability is imposed when protected components are "supplied" or "cause[d] to be supplied." *See* 35 U.S.C. § 271(f).

27. For reasons discussed later in this Comment, these two cases may open the door for increased litigation over software component exports, depending on how software components are ultimately integrated into the extraterritorial infringement provisions of § 271(f).

28. Software development has been a rapidly growing technology field in the United States. Recently, however, many companies have been outsourcing this work to programmers in low cost countries such as India and China. A survey by the Institute for Electrical and Electronic Engineers (IEEE) noted that "companies are outsourcing everything from product design and tech support to employee benefits—to low wage countries like the Philippines, China, Mexico, Costa Rica, India and others." The survey's projections for export of computer programming jobs is particularly troublesome for the domestic job market, predicting a loss of 277,000 jobs by 2010 and 473,000 by 2015. Sharon Richardson, *Watching the Tide Wave White Collars Offshore*, IEEE NEWS & VIEWS, Dec. 2003, at 2. Any further impediments to domestic software development, such as imposing liability for mere export of a software component not protected outside the United States, might serve to accelerate this outsourcing, further impacting the domestic job market.

29. While no cases have yet arisen over Internet-based infringement under 35 U.S.C. § 271(f), one prominent software patent attorney and author suggests this type of Internet scenario as potential grounds for liability. *See* GREGORY A. STOBBS, SOFTWARE PATENTS 659–60 (2d ed. 2000).

This Comment examines these issues by exploring how software components and their export might be harmonized with extraterritorial infringement under 35 U.S.C. § 271(f). As the District Court observed in *Eolas*, fitting software into these newer provisions has been "long debated in this case" and is a matter of first impression, noting that "[t]here is no clear governing precedent."[30] As is common when the patent system attempts to integrate new types of innovations, such as software, merger into the existing paradigm becomes conceptually challenging.[31]

This Comment addresses these issues by first summarizing the basic requirements of patentability and how software, which was initially viewed as unpatentable (and therefore not subject to direct infringement and, by implication, other forms of infringement as well), gradually gained acceptance as patentable subject matter. It then examines patent infringement, its historical limitations, and recent expansion to include extraterritorial activities such as export of components used in overseas

30. Eolas Techs., Inc. v. Microsoft Corp., 274 F. Supp. 2d 972, 973 (N.D. Ill. 2003).

31. Commonly encountered problems include difficulties in basic judicial understanding of the new technology, interpretation of existing statutory provisions in light of its novel characteristics, and decisions regarding whether existing statutes should be interpreted narrowly to exclude the new technology or broadly to include it. Examples of the fundamental understanding problem abound in patent cases. For example, in *Eolas* the District Court made a basic error about the very type of software at issue, referring to what should be described as object code, Microsoft Windows itself, as "source code." *Id.* The distinction between source and object code, a fundamental one when analyzing software, is discussed in more detail later in this Comment. *See infra* notes 193–214 and accompanying text. In addition to the district courts, the Supreme Court also struggles with this problem. In a recent case involving a chemical process, the Court dealt with pH, a logarithmic parameter related to ion concentrations in a liquid and its associated acidity, and described a pH difference of 1.0 as a difference in acidity of a factor of ten. Warner-Jenkinson Co. v. Hilton Davis Chem. Co., 520 U.S. 17, 22 n.1 (1997). On remand, the Federal Circuit indirectly reprimanded the Supreme Court for misconstruing pH, noting that factor of ten represents a "ten-fold difference in hydrogen ion concentration, rather than literally indicating a ten-fold difference in 'acidity'" as the Court had written. Hilton Davis Chem. Co. v. Warner-Jenkinson Co., 114 F.3d 1161, 1164 n.1 (Fed. Cir. 1997). The issue of statutory subject matter interpretation has occurred frequently as revolutionary, as opposed to evolutionary, technologies are developed. Recent examples include struggles to assimilate biotech inventions such as genetically engineered organisms, into the patent system. *See, e.g.*, Diamond v. Chakrabarty, 447 U.S. 303 (1980) (overruling the patent office's assertion that biological organisms were not patentable subject matter). Breadth of statutory interpretation was recently raised in a case concerning a statutory provision of 35 U.S.C. § 271(g), which protects products manufactured abroad using a domestic process patent. *See* Bayer AG v. Housey Pharms., Inc., 340 F.3d 1367, 1376 (Fed. Cir. 2003) (interpreting statute narrowly rather than broadly to preclude protection for intangible data outputs from a patented process).

manufacturing. This is followed by an analysis of software components under § 271(f) and a proposal for classifying various types of software within the statutory extraterritorial infringement provisions. Various means of exporting software are then discussed and a proposal is made for determining whether the export violates the statutory infringement provisions. Finally, several policy concerns are raised suggesting that liability for software component export under § 271(f) be limited. In conclusion, it is argued that software should enjoy some protection under § 271(f), but that the scope of protection should be narrowly construed.

II. PATENTABILITY BASICS

A. Fundamental Requirements

To understand the distinction created by § 271(f) between patented inventions and their components, it is helpful to review the basics of patentability. In the United States, patents enjoy a high pedigree, being expressly authorized by the U.S. Constitution.[32] As implemented, this grant represents a quid pro quo between monopoly and incentive to innovate.[33] This balancing is typically done by limiting the duration of

32. Article I gives Congress the power to grant inventors a monopoly on their work as an incentive for innovation. U.S. CONST. art. I, § 8, cl. 8. This grant, which covers both patents and copyrights, is designed 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." *Id.* The patent grant is unique in Article I Congressional powers in that it specifies not only the power granted but also the means by which the power is to be implemented. The power granted by the Constitution is discretionary but has been used and statutorily codified almost since adoption of the Constitution, with the first patent act appearing in 1790, barely a year after adoption of the Constitution itself. STOBBS, *supra* note 29, at 13.

33. This quid pro quo represents a delicate Congressional balancing act. If Congress chooses to grant a patent monopoly to an inventor, which is generally disfavored for economic reasons, it must balance the disadvantages against the economic incentive to innovate provided by monopoly protection. The Supreme Court, in the case that motivated the enactment of § 271(f), noted that "monopolies—even those conferred by patents—are not viewed with favor." Deepsouth Packing Co. v. Laitram Corp., 406 U.S. 518, 525 (1972) (quoting Laitram Corp. v. Deepsouth Packing Co., 310 F. Supp. 926, 929 (E.D. La. 1970)). In an earlier landmark patent case, Supreme Court Justice Clark noted the historical disfavor of monopolies, pointing out that a "monopoly on tea . . . sparked the [American] Revolution . . . ." Graham v. John Deere Co., 383 U.S. 1, 7 (1966). Thomas Jefferson was adamantly opposed to the Constitutional grant of monopoly power, writing that "the benefit even of limited monopolies is too doubtful to be opposed to that of their general suppression." *Id.* at 8 (quoting 5 THE WRITINGS OF THOMAS JEFFERSON 47 (Paul Leicester Ford ed., 1895)). To balance this disfavor, the Constitution explicitly requires the monopoly be for a limited time and only as an incentive to promote Science and Useful Arts. The balance between monopoly protection and duration has not frequently come into question in the patent domain, which generally limits the monopoly to twenty years or less. 35 U.S.C. § 154(a)(2) (2000).

the patent as well as its geographic boundaries.[34] Congress may also limit the scope of what is patentable.[35] Assessing whether new technologies fit within the scope of patent protection has been a recurring problem, particularly with the revolutionary technological innovations of the past half century.[36] As discussed in this Comment, fitting software into established patentable subject areas has proven to be conceptually difficult.[37]

B. Patentable Subject Matter

Patentable subject matter was codified in the 1952 Patent Act[38] based

34. The current duration for U.S. patents is twenty years from the date of issuance. *See* 35 U.S.C. § 154(a)(2). The geographic boundary limitation traditionally provided protection only within U.S. territorial boundaries. Dowagiac Mfg. Co. v. Minn. Moline Plow Co., 235 U.S. 641, 650 (1915) (nothing that infringement cannot be based on acts done wholly in foreign countries). The recent expansion of this limitation through provisions in 35 U.S.C. § 271 has led to problems as discussed in detail in this Comment.

35. Under current judicial interpretation of the patent statutes, subject matter should be construed broadly because Congress has given it "wide scope." *Chakrabarty*, 447 U.S. at 308 (explaining the Court's broad judicial construction of the various Patent Acts). Nevertheless, statutory interpretation, particularly regarding newer technologies, has led to denial or questionability of patent protection for particular subject matter including software, *see infra* note 37, genetically engineered organisms, *Chakrabarty*, 447 U.S. at 306 (noting the Patent Examiner and the Patent Office Board of Appeals' rejection of patentability of a micro-organism), and under recently overruled doctrines, business methods. State St. Bank & Trust Co. v. Signature Fin. Group, Inc., 149 F.3d 1368, 1375–80 (Fed. Cir. 1998) (reversing a District Court judgment based on the now invalid "business method exception" that business methods were not statutory subject matter).

36. While revolutionary technology alone may raise concerns about patentability, economic impact may push these concerns to the forefront. In the past fifty years, the U.S. economy has grown rapidly, driven heavily by intellectual property such as patents. In 1947, intellectual property comprised just under ten percent of all U.S. exports. By the early 1990s that figure grew to well over fifty percent of all exports. Many of these exports include nontraditional technologies such as integrated circuits, software, and, more recently, biotechnology. WARSHOFSKY, *supra* note 2, at 5–10.

37. In some cases, the problem lies in the intangible nature of certain types of innovations such as software. *See, e.g.*, Gottschalk v. Benson, 409 U.S. 63, 72 (1972) (noting uncertainty regarding the patentability of software and suggesting that patentability of software is a "policy matter" to which the court is not "competent to speak"). In other cases, particular subject matter, such as living organisms, might initially be viewed as nonstatutory subject matter. *See, e.g.*, *Chakrabarty*, 447 U.S. at 306 (reversing the patent office's rejection based on the Patent Board of Appeals' conclusion that "[35 U.S.C.] § 101 was not intended to cover living things such as these laboratory created micro-organisms").

38. Act of July 19, 1952, Pub. L. No. 593, 66 Stat. 797 (codified as amended in 35 U.S.C. §§ 1–376).

on previous statutory and common law definitions.[39] Included among the protectable categories of invention is any "process, machine, manufacture, or composition of matter, or any new and useful improvement thereof."[40]

Current patentable subject matter is construed broadly and may include nontraditional inventions such as software and "business methods."[41] As one recent Supreme Court decision has held, patentable subject matter now includes "anything under the sun that is made by man."[42] Nevertheless, the Supreme Court has not clearly extended software patent protection beyond software used to control or manipulate physical entities or processes.[43] The Federal Circuit has, however, attempted to broaden this protection to include any form of software.[44]

---

39. Several patent acts preceded the 1952 Act, including acts of 1790, 1793, and 1836. The 1952 Patent Act represented the first codification under the United States Code, which was first established in 1926. The 1952 Patent Act codification primarily retained statutory acts passed from 1874 onward as well as much of the contemporaneous case law. STOBBS, *supra* note 29, at 15–24.

40. 35 U.S.C. § 101 (2000). In addition to meeting the statutory subject matter categories of § 101, the invention must also be useful or have utility; however, utility is interpreted broadly. As a consequence, utility does not currently present a high bar to patentability. Certain subject matters, such as perpetual motion machines, are not patentable because they lack credible utility; i.e., they do not actually work. Historically, certain types of morally undesirable inventions, such as gambling machines, were also considered to lack beneficial utility and were, therefore, unpatentable. *See, e.g.*, Juicy Whip, Inc. v. Orange Bang, Inc., 185 F.3d 1364, 1366 (Fed. Cir. 1999) (referencing previously ineligible subject matter lacking beneficial utility, including inventions used for gambling, poisoning people, promoting debauchery, or assassination). In addition to utility, the invention must be new or novel. 35 U.S.C. § 102 (2000). The novelty requirement is not absolute, but any previous printed publication from anywhere in the world, or previous use or sale more than one year prior to the date of a patent application, can bar issuance or invalidate an already issued patent. *Id.* This requirement is probably one of the most litigated in patent law as accused infringers have a very strong incentive to search for any possible prior version of the invention, known in the patent trade as prior art, given the typically limited search required by the patent office. Finally, a patented invention must be nonobvious. 35 U.S.C. § 103 (2000). This requirement is the most amorphous, making nonobviousness a subjective matter based on what a person having ordinary skill in the respective field would have deemed obvious at the time of invention. *Id.*

41. One commentator suggests the common thread joining all statutory subject matter is human control. STOBBS, *supra* note 29, at 205. Under this view, if humans have the power to change something, or make it exist or cease to exist, it is patentable subject matter. *Id.*

42. *Chakrabarty*, 447 U.S. at 309 (quoting S. REP. NO. 1979, at 5 (1952); H.R. REP. NO. 1923, at 6 (1952)).

43. The Supreme Court last visited the software patent issue in 1981, validating use of software in connection with machinery for rubber molding. *See* Diamond v. Diehr, 450 U.S. 175, 184–85 (1981) (holding that a rubber molding process is patentable even if it uses software in its implementation). In this case, the software was part of a process patent "which includes in several of its steps the use of a mathematical formula and a programmed digital computer . . . ." *Id.* at 177.

44. *See, e.g.*, State St. Bank & Trust Co. v. Signature Fin. Group, Inc., 149 F.3d 1368, 1373–77 (Fed. Cir. 1998) (providing protection for software implementing numerical algorithms that produce intangible results expressed in the form of numbers as

C. Patent Claims

In addition to meeting the subject matter requirements, each patent must describe the invention in a written "specification."[45] The specification must be concluded with one or more elements known as claims.[46] The claims are the most significant and important feature of any patent because they define the patent's scope.[47] When an inventor seeks to enforce his patent against a potential infringer, it is the scope of the claims, rather than anything else in the patent, that determines the extent of what is protected.[48]

A related role of patent claims is to classify the invention into one of two general subject categories, an apparatus or a process.[49] These two

---

well as "business methods," which are novel processes or methods of doing business).

45. 35 U.S.C. § 112 (2000). This requirement serves several purposes including demonstrating that the inventor had possession, or the ability and knowledge to actually make the invention, and has communicated this in the patent application. *See* Gould v. Hellwarth, 472 F.2d 1383, 1387–88 (C.C.P.A. 1973) (denying patent because application failed to enable a person skilled in the art to make the invention). The invention must be described sufficiently to enable another "skilled in the art to which it pertains" to make and use the invention. 35 U.S.C. § 112. This requirement serves the basic patent function of providing disclosure to enable the public to both improve upon the invention and practice it once the patent has expired without resorting to excessive experimentation. *See, e.g.*, Atlas Powder Co. v. E.I. du Pont De Nemours & Co., 750 F.2d 1569, 1576 (Fed. Cir. 1984) (discussing enablement and noting that a patent will not be held to be nonenabling because of a need for experimentation as long as the experimentation required is not undue). If the inventor envisions a particular "best mode," or best way of making or using the invention, this must also be described in the patent. 35 U.S.C. § 112. The best mode requirement serves to protect the public's interest in a full disclosure of the invention, as envisioned by the inventor, at the time the application is filed and preclude monopoly protection when the inventor's knowledge has not been fully disclosed. *See* Chemcast Corp. v. Arco Indus. Inc., 913 F.2d 923, 926 (Fed. Cir. 1990).

46. Claims are required to "particularly point[] out and distinctly claim[] the subject matter which the applicant regards as his invention." 35 U.S.C. § 112.

47. Claims are commonly analogized to the metes and bounds of real property as a means of staking out the boundaries of the invention. *See, e.g.*, Warner-Jenkinson Co. v. Hilton Davis Chem. Co., 520 U.S. 17, 29 (1997). Metes and bounds refers to the "territorial limits of real property" as used in the real property context. They are commonly described in "deeds and surveys to establish the boundary lines of land." BLACK'S LAW DICTIONARY 1005 (7th ed. 1999).

48. Determination of claim scope and meaning is a primary requirement in any patent litigation. Before a potentially infringing device or process can be analyzed, the scope of the patent must be determined by examining the claims. This is a function for a judge, not a jury. *See* Markman v. Westview Instruments, Inc., 517 U.S. 370, 384–90 (1996) (creating the so called Markman hearing which requires a judge, rather than a jury, to construct patent claims).

49. In addition, a patent claim can cover a composition of matter. 35 U.S.C. § 101

classifications derive from the general subject matter categories[50] and have a significant impact on the infringement analysis.[51]

One way to claim an invention is as a process or method.[52] A process patent claim typically recites a series of sequential steps or actions required to complete some action producing a useful result. Because this sequential approach models the operation of processor instruction execution in a computer, software patents are frequently written in this process or method form.[53] An apparatus claim describes a device composed of a series of components that, when taken as a whole, constitute the patented invention.[54] Software can also be claimed in this way as a component or element of a device or apparatus.[55] Traditionally, software was claimed primarily in process form; however, recent patents often claim software in both forms, providing broad protection with the advantages of both claiming methods.[56] As discussed in detail later in this Comment, the method of claiming should play a key role in assessing whether a software component should enjoy protection under § 271(f). Only software claimed in apparatus form should receive protection.[57]

(2000). Because software does not fall within composition of matter claims (software is not matter), this type of claim is not discussed here.

50. Process means a "process, art or method, and includes a new use of a known process, machine, manufacture, composition of matter, or material." 35 U.S.C. § 100(b) (2000). In the patent context, a process or method generally refers to a series of steps for doing something, as opposed to a machine or device. A patent may, however, include claims of both types, with one or more covering a method of doing something and other claims describing a particular device for performing the process. *See, e.g.*, U.S. Patent No. 6,609,431 (issued Aug. 26, 2003) (teaching a flow meter patented by the author for measurement of viscous fluids that includes both process and apparatus claims).

51. *See infra* text accompanying notes 211–22.

52. 35 U.S.C. § 101.

53. STOBBS, *supra* note 29, at 521–633 (providing examples of typical software patents and noting that process claims are the most common form of software patent protection).

54. *See* Great Atl. & Pac. Tea Co. v. Supermarket Equip. Corp., 340 U.S. 147, 152 (1950). Apparatus claims would fall under the statutory categories of "machine [or] manufacture" or an improvement of either of these. *See* 35 U.S.C. § 101.

55. An example is the author's patent for a flowmeter which includes mechanical, electronic, and software components, which, when taken as a whole, comprise the invention. *See* U.S. Patent No. 6,609,431 (issued Aug. 26, 2003).

56. *Id.*

57. This proposed limitation is based on an earlier district court case that limited infringement liability under § 271(f) when the allegedly infringed patent only claimed the invention in process form. *See* Enpat, Inc. v. Microsoft Corp., 6 F. Supp. 2d 537, 538 (E.D. Va. 1998). For discussion of this proposed limitation, see *infra* notes 213–22 and accompanying text.

D. Development of Software Patentability

The fundamental question of whether software should be considered a component for liability purposes under 35 U.S.C. § 271(f) must rest, in part, on whether software itself is patentable subject matter. While the individual components of a particular invention may not enjoy patent protection under § 271(f),[58] it would be inconsistent to impose liability for a component that could not, rather than does not, enjoy patent protection.[59] Therefore, this Comment starts with the basic assumption that to be considered a component for liability purposes under § 271(f), the component itself must fall within the statutory subject matter categories.[60]

Digital computers and the software required to make them operable[61] first emerged in the 1940s;[62] however, acceptance of software as patentable subject matter was not firmly established until the mid

58. Section 271(f) does not specifically define "component." However, the context implies that a component must be some subset of the patented invention, and therefore not necessarily protected, itself, if it does not reach all the claims of the invention. *See* 35 U.S.C. § 271(f) (2000).

59. The categories of subject matter that clearly do not enjoy patent protection are abstract algorithms, laws of nature, physical phenomena, and products of nature, such as living organisms. Diamond v. Chakrabarty, 447 U.S. 303, 309–10 (1980). The living organism limitation does not apply to living organisms that have been fundamentally modified by man and would not be found in nature, such as gene-spliced bacteria. *Id.* Therefore, under this proposed subject matter limitation, no protection would be provided for components such as chemicals merely found in nature, living organisms, or their byproducts that were discovered in the wild and not modified in any way, even if they were a substantial part of the invention.

60. As noted previously, this is not currently considered a limitation. However, neither the Supreme Court nor Congress has made a final definitive statement regarding the scope of software patentability. *See infra* notes 61–83 and accompanying text.

61. The general model for a computer includes hardware comprised of electronic circuitry implementing data storage and digital logic functions. Software provides the instructions that control how the digital logic functions operate. One scholar compares a computer program to a machine:

> The program text represents the machine before it has been turned on. The executing program represents the powered-up machine in active operation. There is no fundamental distinction between the passive program text and the active executing program, just as there is none between a machine before and after it is turned on.

DAVID GELERNTER & SURESH JAGANNATHAN, PROGRAMMING LINGUISTICS 2 (1990).

62. The first electronic digital computer, the ENIAC (Electronic Numerical Integrator and Computer), was completed in 1946 for the military. ENIAC's inventors subsequently developed the first commercial electronic computer, the UNIVAC I (Universal Automatic Computer), in the early 1950s. ROBERT P. MERGES ET AL., INTELLECTUAL PROPERTY LAW IN THE NEW TECHNOLOGICAL AGE 519–20 (Supp. 2002).

1990s.[63] For this reason, it is questionable whether Congress, in adopting the extraterritorial infringement provisions included in the Patent Law Amendments of 1984,[64] considered software as a possible component for infringement purposes under § 271(f).[65] Nevertheless, a brief review of the evolution of acceptance of software as patentable subject matter provides some insight into similar concerns about software's intangible nature and suggests that software should not be distinguished from mechanical components under § 271(f) merely because of its nonphysical nature.

Through the 1960s, there was a general view that software was not patentable subject matter. A 1966 Presidential Commission recommended that software not be patentable.[66] The issue of software patentability first received Supreme Court attention in 1972 with a landmark case, *Gottschalk v. Benson*.[67] *Benson* rejected patentability of mathematical algorithms, which are the heart of most software, but did not explicitly reject software patentability.[68] A 1978 Supreme Court case, *Parker v.*

---

63. In 1996, the United States Patent and Trademark Office finally provided some guidance for its examiners in reviewing computer inventions by issuing Examination Guidelines for Computer-Related Inventions, reprinted in 17 J. MARSHALL J. COMPUTER & INFO. L. 311 (1998). The Federal Circuit removed perhaps the last basic limitation on software patentability in 1998 by holding that a program that merely manipulates numbers, as long as the process produces a "'useful, concrete, and tangible result' . . . renders it statutory subject matter, even if the useful result is expressed [only] in numbers . . . ." State St. Bank & Trust Co. v. Signature Fin. Group, 149 F.3d 1368, 1375 (Fed. Cir. 1998) (quoting *In re* Alappat, 33 F.3d 1526, 1544 (Fed. Cir. 1994)).

64. Patent Law Amendments Act of 1984, Pub. L. No. 98-622, 98 Stat. 3383. Portions of these amendments were subsequently codified as 35 U.S.C. § 271(f)(1)–(2) (2000).

65. There are at least two reasons for questioning whether Congress considered software in this context. First, the status of software as patentable subject matter was not clearly established at this time. *See infra* notes 66–71. In addition, the state of the art in computer hardware and the associated software was much more limited than it is today. Current embedded applications where computer hardware and software replaces older analog electronics or physical components, such as the machine parts in *Deepsouth*, were much less pervasive in the early 1980s. *See infra* notes 163–165.

66. REPORT OF THE PRESIDENT'S COMMISSION ON THE PATENT SYSTEM 12–13 (1966). This rejection was based in part on rejection of direct attempts to patent programs and subsequent indirect attempts to work around these rejections. The Commission found that the patent office rejected direct patenting because of an inability to determine a patent's novelty: "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 mere registration and the presumption of validity [associated with issued patents] would be all but nonexistent." *Id.* at 13. In response, attorneys tried to work around this direct ban by clever claiming efforts, such as drafting the claims as a "process, . . . rather than as a program itself." *Id.* The result of these efforts was to further confuse the issue. The report therefore recommended that any form of patenting "should not be permitted." *Id.*

67. 409 U.S. 63 (1972).

68. Although *Benson* did not expressly reject software patentability, it did reject mathematical algorithms as being analogous to laws of nature that are not patentable subject matter. The algorithm at issue in *Benson* implemented a BCD to binary

Flook,[69] further muddied the waters by holding that "a claim for an improved method of calculation [in the form of software in a digital computer], even when tied to a specific end use, is unpatentable subject matter under § 101."[70] As a consequence, software, which could be viewed as merely a set of mathematical algorithms, was basically disregarded as patentable subject matter until the 1980s.[71]

A 1981 Supreme Court case, *Diamond v. Diehr*, reopened the door for software patentability.[72] *Diehr* involved a process for curing rubber that included software for calculating cure time. The patent was rejected as nonstatutory by the patent office.[73] In reversing this rejection, the Supreme Court distinguished the practical use of computation in this context from proscribed uses, such as abstract ideas. By allowing protection, the Court recognized the use of software as a component of a machine, such as the rubber making device at issue, that produces some tangible, physical output—in this case, cured rubber.[74] This holding suggested that the Court finally recognized that software can play a role equivalent to hardware in machines and devices and might, in effect, be interchangeable with hardware in some applications.[75]

---

conversion that required a series of bit manipulations and produced only numbers as the output. *Id.* at 65–68. While *Benson* was read by many as rejecting software as patentable subject matter, the Court did not explicity reject software as patentable subject matter, but rather implied that it might be protectable if tied to a specific practical application rather than just producing numbers as an output. As Justice Douglas wrote, "The mathematical formula involved here has no substantial practical application . . . and in practical effect would be a patent on the algorithm itself." *Id.* at 71–72. Nevertheless, one commentator noted that "despite the express statement that this was not a prohibition on patents for computer programs, some concluded that computer programs were not patentable." STOBBS, *supra* note 29, at 42.

69. 437 U.S. 584 (1978).

70. *Id.* at 595 n.18. This dismissal of algorithms tied to a specific end use appeared to close the "practical application" exception implied by *Benson* and therefore to software patentability in general.

71. STOBBS, *supra* note 29, at 41.

72. 450 U.S. 175 (1981).

73. In the patent application process, the process claims relating to calculating curing time were rejected solely because they were drawn to nonstatutory subject matter in light of *Benson*. *Id.* at 179–81.

74. The Court held that "[t]ransformation and reduction of an article 'to a different state or thing' is the clue to the patentability of a process claim that does not include particular machines." *Id.* at 184 (quoting *Benson*, 409 U.S. at 70).

75. The patent at issue in Diehr was claimed as a process. The Court noted that processes enjoy statutory protection under 35 U.S.C. § 101 and explicitly recognized the interchangeability of hardware and software in specific process steps by noting that "[o]ur conclusion regarding respondents' claims is not altered by the fact that in several steps of the process a . . . digital computer [rather than an electronic or mechanical

Diehr was the last Supreme Court case addressing software patentability; however, the Federal Circuit continued to expand software protection. In *In re Alappat*,[76] the Federal Circuit extended protection to software that enhances video displays on an oscilloscope. In *In re Lowry*,[77] the court extended protection to data structures stored in memory, and in *In re Beauregard*[78] the court authorized protection of software as articles of manufacture. In the most significant recent software case, *State Street Bank and Trust Co. v. Signature Financial Group*,[79] the Federal Circuit rejected earlier limitations that software provide tangible results, allowing claims for an invention that merely processes and outputs numbers. *State Street* involved a data processing system, claimed in apparatus form, that processed daily stock prices and output share value, both in the form of numbers only. Judge Rich characterized the claims at issue as a machine composed of software elements.[80] He then erased the earlier conceptual distinction between software and hardware, writing that the relevant test was not whether the device is composed of hardware or software elements but that it produces "a useful, concrete . . . result."[81]

These Supreme Court and Federal Circuit cases demonstrate an increasingly expansive view of software patent protection as well as recognition of software's importance within the patent system. This indicates that the law will recognize software as patentable subject matter—irrespective of whether tangible, physical results are produced—as long as it conforms with traditional limitations on protection for mere abstract concepts.[82] As a consequence, software should clearly be given legal recognition as a potential component part of a machine or apparatus, or even as a standalone machine itself.[83]

---

device is] used." *Id.* at 185. Moreover, the Court held that "a claim drawn to subject matter otherwise statutory does not become nonstatutory because [it uses] a computer [program]." *Id.* at 181.

76. 33 F.3d 1526 (Fed. Cir. 1994). Alappat's invention related to a method for creating a smooth oscilloscope waveform on a cathode ray tube display, similar to typical television tubes. *Id*. at 1538–39. The Federal Circuit reversed a Patent Board of Appeals decision rejecting patentability of a oscilloscope display raterizer function that could be programmed on a general purpose computer. *Id*. at 1545. The rejected claims at issue in *Alappat* were made in apparatus, rather than process, form. *Id.* at 1538–39.

77. 32 F.3d 1579, 1580–85 (Fed. Cir. 1994).

78. 53 F.3d 1583, 1584 (Fed. Cir. 1994).

79. 149 F.3d 1368 (Fed. Cir. 1998).

80. Judge Rich, writing for a three judge panel, held that "[w]hen independent claim 1 is properly construed . . . , it is directed to a machine . . . ." *Id.* at 1371.

81. *Id.* at 1373.

82. The Court has frequently emphasized that mere abstract ideas, mathematical formulas, or laws of nature are not patentable. *See, e.g.*, Gottschalk v. Benson, 409 U.S. 63, 67 (1972); Diamond v. Diehr, 450 U.S. 175, 185 (1981).

83. In *Alappat*, claim 15 consists of a rasterizer system (machine) that includes analog and digital hardware as well as distinct, protectable software elements. *In re* Allappat, 33 F.3d 1526, 1538–39 (Fed. Cir. 1994). In *State Street*, computer hardware

III. PATENT INFRINGEMENT AND EXTRATERRITORIAL ACTIVITIES

Once a patent is issued, the owner has the legal right to prevent others from infringing his invention. Infringement involves any "act that interferes with one of the exclusive rights of a patent, copyright, or trademark owner."[84] The various intellectual property regimes each include unique statutory requirements for infringement.[85] As a general rule, however, infringement involves misuse of a protected right within defined subject matter,[86] temporal limits,[87] and geographical limits.[88] Infringement liability under § 271(f) is an exception to this rule and imposes liability in a way that is directly counter to these traditional subject matter and geographic limitations.[89]

A. Direct Infringement

The most common and straightforward form of patent infringement is direct infringement.[90] Direct infringement liability is imposed on anyone who "without authority makes, uses, . . . or sells any patented invention, within the United States . . . ."[91] There are several important limitations to direct infringement that narrow potential liability. First, the scope of what is covered by the patent must be assessed and then compared to the potentially infringing device.[92] A patent is directly infringed only if all the claimed elements are present in the allegedly

---

and software comprise the entire claimed invention under claim 1. As Judge Rich held, this claim "is directed to a machine." 149 F.3d at 1375. Therefore, the entire software system is the machine and the individual software elements are its component parts.

84. BLACK'S LAW DICTIONARY 785 (7th ed. 1999).

85. Patent infringement is statutorily detailed in Title 35 of the United States Code. *See* 35 U.S.C. § 271 (2000). Analagous copyright infringement provisions are included in title 17. *See* 17 U.S.C. § 501 (2000 & Supp. 2004).

86. Patent infringement applies only to a "patented invention," as set out explicitly by the patent's claims. *See, e.g.*, 35 U.S.C. § 271(a), (c).

87. Patent duration is currently codified as "ending 20 years from the date on which the application for the patent was filed in the United States . . . ." 35 U.S.C. § 154(a)(2) (2000).

88. Direct patent infringement applies only to those who make, use, offer to sell, or sell any patented invention "within the United States." 35 U.S.C. § 271(a). The term "United States" is defined as "the United States of America, its territories and possessions." 35 U.S.C. § 100(c) (2000).

89. This distinction is discussed later in this Comment. *See infra* notes 119–30 and accompanying text.

90. 35 U.S.C. § 271(a).

91. *Id.*

92. *See* Markman v. Westview Instruments, Inc., 517 U.S. 370, 384 (1996).

infringing device; a device that does not include all the claimed elements does not infringe.[93] The second limitation is territorial—the infringing activity must occur within the United States.[94] Courts have long held patent protection to this basic geographic limitation.[95] Therefore, anyone making, using, or selling a device that includes all the claimed elements outside the territorial limits of the United States does not infringe under traditional infringement analysis.[96]

B. The Extraterritorial Manufacturing Problem

As a consequence of these limitations, a manufacturer would not traditionally be liable for making some or all of the components of an infringing product if they completed assembly outside the country.[97] This led some companies to procure most or all of the components of an infringing device domestically. To avoid infringement liability, they merely shipped the unassembled parts overseas for final assembly and sale.[98]

i. Deepsouth v. Laitram

The Supreme Court confronted the extraterritorial manufacturing problem in an important 1972 case, *Deepsouth Packing Co. v. Laitram Corp.*[99] *Deepsouth* strictly limited direct infringement liability under § 271(a)[100] to making a patented invention when the assembly occurs overseas. The Laitram Corporation was actively involved in design and manufacturing of mechanical devices for shrimp processing and held two valid U.S. patents for machinery used in the process of deveining

93. Larami Corp. v. Amron, 27 U.S.P.Q. 2d 1280, 1283 (E.D. Pa. 1993) (citing London v. Carson Pirie Scott & Co., 946 F.2d 1534, 1538–39 (Fed. Cir. 1991)), *aff'd*, 1996 U.S. App. LEXIS 8953 (Fed. Cir. Apr. 19, 1996).

94. 35 U.S.C. § 271(a). The United States includes "the United States of America, its territories and possessions." 35 U.S.C. § 100(c).

95. *See, e.g.*, Dowagiac Mfg. Co. v. Minn. Moline Plow Co., 235 U.S. 641, 650 (1915); Brown v. Duchesne, 60 U.S. (19 How.) 183, 195 (1857).

96. *See infra* notes 116–17 and accompanying text.

97. Note also the use limitation. Section 271(a) proscribes use of a patented device but not mere possession. Therefore, if someone merely owns, but does not use, a patented product, he is not liable for patent infringement under § 271(f). He may, however, be liable if he made the device, even if he did not use it. *See* 35 U.S.C. § 271(a).

98. This practice would understandably raise the ire of competitors, particularly if the final assembly was merely a trivial effort. Nevertheless, as noted previously, *supra* note 93, direct infringement cannot occur unless all the elements of the claimed invention are met, which requires complete assembly into an operable unit.

99. 406 U.S. 518, 519 (1972).

100. 35 U.S.C. § 271(a). Among other actions, § 271(a) imposes liability for "making" an invention that is covered by the claims of a patent. *Id.*

shrimp.[101] Both of these patents were combination patents, a common form of patented invention made from a series of unpatented parts where:

> "[n]one of the parts referred to are new, and none are claimed as new . . . or stated to produce any given result. The end . . . [is] accomplished by the union of all, arranged and combined together in the manner described. And this combination, composed of all the parts mentioned . . . is stated to be the improvement, and is the thing patented."[102]

The value of this type of invention lies in the combination rather than the individual parts.[103] Therefore, while individual parts of these inventions would not enjoy patent protection, their assembly into the complete invention constitutes infringement. This distinction is important in understanding the scope of the term "components," as used in the extraterritorial infringement provisions of § 271(f).[104]

---

101. *Deepsouth*, 406 U.S at 519–20. Deveining is the process of opening and removing the shrimp's intestines, also known as veins (or sand bags), to make them more palatable. *Id.* at 519 (quoting Laitram Corp. v. Deepsouth Packing Co., 301 F. Supp. 1037, 1040 (E.D. La. 1969)). The process also removes the shell, making the shrimp edible. *Deepsouth*, 406 U.S at 519 (quoting *Laitram*, 301 F. Supp. at 1040). One patent, issued in 1954, covered a "slitter" which exposed the shrimp's veins by using water pressure to force the shrimp through a series of razor blades, which slit the shrimp open. *Deepsouth*, 406 U.S at 520. The second patent, issued in 1958, disclosed a "tumbler," a device to mechanically remove the veins from shrimp whose backs were opened by the slitter. *Id.*

102. *Id*. at 521 (quoting Prouty v. Ruggles, 41 U.S. (16 Pet.) 336, 341 (1842)).

103. While none of the components of these machines enjoyed patent protection, both of Laitram's inventions were nevertheless protected for "in some way exceed[ing] the sum of [their] parts." *Id.* at 521–22 (quoting Great Atl. & Pac. Tea Co. v. Supermarket Equip. Corp., 340 U.S. 147, 152 (1950)). As the court notes:

> The slitter's elements . . . were: an inclined trough, a "knife" (actually, knives) positioned in the trough, and a means (water sprayed from jets) to move the shrimp down the trough. The tumbler's elements include a "lip," a "support member," and a "means" (water thrust from jets). As is usual in combination patents, none of the elements in either of these patents were themselves patentable at the time of the patent, nor are they now. . . . Invention was recognized because Laitram's assignors combined ordinary elements in an extraordinary way—a novel union of old means was designed to achieve new ends.

Id. at 520 (citations omitted). This synergy of components was also described by the District Court as being protectable because "when combined [the parts] co-act in such a manner to perform a new function and produce new results." *Laitram*, 301 F. Supp. at 1063.

104. Although the term "component" is not explicitly defined in the patent act, it was likely intended to refer to an article such as the parts described here, which do not enjoy patent protection by themselves, but contribute to a combination invention such as the shrimp deveining machine at issue in *Deepsouth*. Similar language is used in the

Deepsouth made similar machines and held related patents.[105] The District Court established, however, that Laitram had the "superior claim"[106] to essentially identical machinery and was, therefore, entitled to exclude Deepsouth from the domestic market[107] and exercise standard remedies.[108] Deepsouth sought to modify a complete injunction imposed by the Fifth Circuit[109] allowing it to continue its practice of shipping its deveining machinery to foreign customers for use abroad. This practice consisted of shipping the domestically infringing machine to foreign customers in three separate boxes, each containing only parts of the nearly two-ton machine. Assembly abroad could be completed in less than one hour.[110]

In a close decision,[111] the Court held that Deepsouth's practice of shipping the unassembled machine abroad did not constitute infringement. The Court first dismissed liability for contributory infringement, noting that "'if there is no [direct] infringement of a patent there can be no contributory infringer.'"[112] It then dismissed Laitram's suggestion that

---

statutory provisions addressing contributory liability for domestic activities. *See, e.g.*, 35 U.S.C. § 271(c) (2000).

105. *Deepsouth*, 406 U.S at 519.

106. *Id.* (citing Laitram Corp. v. Deepsouth Packing Co., 443 F.2d 928, 934 (5th Cir. 1971)).

107. A key concept of patent law is the exclusion right of patent protection. A patent does not provide any rights of use, it only allows a patent holder to exclude others from the market. This exclusion principle includes the right to prevent others from "making, using, . . . or selling the invention throughout the United States . . . ." 35 U.S.C. § 154(a)(1) (2000). The Court describes this as "the keystone provision of the patent code." *Deepsouth*, 406 U.S. at 522. Once the Court established priority of invention, Laitram was entitled to enjoin Deepsouth from the domestic market for making, using, or selling any invention that fell within Laitram's patent claims. *Id.* at 519–22.

108. Typical patent remedies include injunctions as well as damages. Injunctions are the standard remedy to preclude present and future infringement. 35 U.S.C. § 283 (2000). In addition, liability can be imposed for past damages including actual damages or a reasonable royalty. 35 U.S.C. § 284 (2000). Neither the injunction nor the award of damages for domestic infringement was disputed in this case. *Deepsouth*, 406 U.S. at 523.

109. *Deepsouth*, 406 U.S. at 524. The injunction against this type of component export imposed by the Fifth Circuit was in conflict with previous decisions finding no liability for component export by the Second, Third, and Seventh Circuits. *Id*. at 525. The Fifth Circuit's panel decision held that allowing component export was "an artificial, technical construction . . . [that] [subverted] the Constitutional scheme of promoting 'the Progress of Science and useful Arts,'" as provided in the Constitution. *Id.* (quoting *Laitram*, 443 F.2d at 939 (citing U.S. CONST. art. I, § 8, cl. 8)).

110. *Id.* at 523–24.

111. The Court's decision was five to four. *See id.* at 532 (listing four dissenting Justices).

112. *Id.* at 526 (quoting Mercoid Corp. v. Mid-Continent Inv. Co., 320 U.S. 661, 677 (1944) (Frankfurter, J., dissenting on other grounds)). There is no direct infringement here because the purchaser of the machine is abroad and presumably any making or use of the machine will occur outside the United States. *See* 35 U.S.C. § 271(a),(c).

Deepsouth makes and sells the invention. In order to sell, Deepsouth must first make the patented invention, and not something less that "fell short of infringement."[113] In rejecting export of components as making the patented product, the Court addressed a conflict over the definition of what constitutes making[114] and cited "a line of decisions so firmly embedded in our patent law as to be unassailable absent a congressional recasting of the statute."[115] This precedent limits liability for a combination patent to only an "operable assembly of the whole and not the manufacture of its parts."[116] Therefore, if components are merely exported, there is no liability for making the patented invention when assembly occurs abroad, and, consequently, there will be no liability for selling the invention because the invention was not made in the United States. The Court concluded by noting that inventors seeking to gain extraterritorial protection have a satisfactory means of securing it by seeking foreign patent protection.[117] The Court also noted that Congress is free at any time to redefine the outer boundary of patent protection if it so desires.[118]

---

113. *Id.* at 527.

114. The Fifth Circuit adopted a definition that "'makes' means what it ordinarily connotes—the *substantial* manufacture of the constituent parts of the machine." *Laitram*, 443 F.2d at 939 (emphasis added). This conflicts with earlier precedent from the Second Circuit that held that something is not "made" until it is completely assembled. Radio Corp. of Am. v. Andrea, 79 F.2d 626, 628 (2d Cir. 1935) ("No wrong is done the patentee until the combination is formed.").

115. *Deepsouth*, 406 U.S. at 528.

116. *Id.* The Court cited an earlier decision that held that "a patent on a combination is a patent on the assembled or functioning whole, not on the separate parts." Mercoid Corp. v. Minneapolis-Honeywell Regulator Co., 320 U.S. 680, 684 (1944); *see also* Aro Mfg. Co. v. Convertible Top Replacement Co., 365 U.S. 336, 344 (1961) ("[I]f anything is settled in the patent law, it is that the combination patent covers only the totality of the elements in the claim and that no element, separately viewed, is within the grant.").

117. *Deepsouth*, 406 U.S. at 531. The Court also noted that Laitram did not adequately explain why it had not secured foreign patent protection despite the fact that the wording of 35 U.S.C. § 154 reveals a Congressional intent to have inventors seek foreign protection for overseas activities. *Id.*

118. *Id.* at 530. When Congress drafted the infringement statute it gave no indication that it intended to expand existing case law regarding patent scope. The Court also noted the "[n]ation's historical antipathy to monopoly and . . . repeated congressional efforts to preserve and foster competition," suggesting a strong preference against expanded patent protection. *Id.* (citation omitted).

ii. Congressional Response to Deepsouth, 35 U.S.C. § 271(f)

At several points in *Deepsouth*, the Supreme Court pointed out the Congressional prerogative of changing patent law by statute.[119] It took twelve years, but Congress responded by enacting The Patent Law Amendments of 1984,[120] the first extraterritorial infringement provision since the 1952 Patent Act.[121] The legislative history of § 271(f) is fairly sparse, with no House or Senate Reports, and only a brief analysis provided in the Congressional Record.[122] The Congressional Record states that the Patent Law Amendments were intended to:

> prevent copiers from avoiding U.S. patents by supplying components of a patented product in this country so that the assembly of the components may be completed abroad. This proposal responds to the United States Supreme Court decision in *Deepsouth Packing Co. v. Laitram Corp.*, 406 U.S. 518 (1972), concerning the need for a legislative solution to close a loophole in patent law.[123]

This brief section represents the extent of legislative history regarding § 271(f). The remainder of the Congressional report merely summarizes the statutory language of § 271(f), element by element.[124]

iii. 35 U.S.C. § 271(f) Statutory Provisions

Section 271(f) is divided into two subsections, § 271(f)(1) and § 271(f)(2). Both subsections require three basic elements to impose liability: knowledge, component status, and supply. The language of §§ 271(f)(1) and (2) is similar to existing statutory infringement provisions for induced and contributory infringement, but, unlike these provisions, neither requires a direct infringement to impose liability.[125]

---

119. In addressing case law defining "making," the Court noted that a contrary interpretation would be improper "absent a congressional recasting of the statute." *Id.* at 528. "When, as here, the Constitution is permissive, the sign of how far Congress has chosen to go can come only from Congress." *Id.* at 530. "We would require a clear and certain signal from Congress before approving [the position advocating extraterritorial liability] . . . ." *Id.* at 531.

120. Patent Law Amendments Act of 1984, Pub. L. No. 98-622, 98 Stat. 3383 (codified in part as 35 U.S.C. § 271(f)).

121. The Act adds 35 U.S.C. § 271(f)(1) and 35 U.S.C. § 271(f)(2), and adds a number of additional patent provisions addressing statutory registration, foreign filing licenses, prior art, joint inventors, and arbitration of patent interferences. Patent Law Amendments Act of 1984, Pub. L. No. 98-622, 1984 U.S.C.C.A.N. (89 Stat. 3383) 5827–36.

122. "No House or Senate Report was submitted with this legislation [enacting 35 U.S.C. § 271(f)]." *Id.* at 5827.

123. *Id.* at 5828.

124. *Id.* at 5828–40.

125. Both induced and contributory infringement fall into the general category of indirect infringement. Induced infringement was a codification of existing case law that

Section 271(f)(1) imposes liability for anyone who knowingly supplies components in a manner that actively induces combination outside the United States where the combination would infringe if done within the United States.[126] This subsection closely mimics the facts of *Deepsouth* where an entire unassembled device was shipped in component form and is analogous to induced infringement. Liability under § 271(f)(1) does not require that the particular components be specially made for the patented invention, but does require that either the entire infringing invention or a substantial portion be shipped.

Section 271(f)(2) addresses a different concern and is similar to induced infringement under 35 U.S.C. § 271(b). Liability under § 271(f)(2) is imposed when one supplies a component that is especially made or adapted for use in the invention with knowledge that the combination would result in infringement if done within the United States.[127] Under § 271(f)(2), liability can be imposed even for export of a single component, provided it was made with knowledge of the patented invention and has no other substantial noninfringing use.[128] Section

---

"extended infringement liability to persons who encouraged and assisted the infringement of others." SCHECHTER & THOMAS, *supra* note 24, at 472; *see also* 35 U.S.C. § 271(b) (2000). Contributory infringement concerns the "situation where one party sells a specially manufactured component that the customer intends to employ in the practice of a patented invention." SCHECHTER & THOMAS, *supra* note 24, at 472; *see also* 35 U.S.C. § 271(c) (2000).

126. The complete provision reads:

> Whoever without authority supplies or causes to be supplied in or from the United States all or a substantial portion of the components of a patented invention, where such components are uncombined in whole or in part, in such manner as to actively induce the combination of such components outside the United States in a manner that would infringe the patent if such combination occurred within the United States, shall be liable as an infringer.

35 U.S.C. § 271(f)(1) (2000).

127. The language of § 271(f)(2) states:

> Whoever without authority supplies or causes to be supplied in or from the United States any component of a patented invention that is especially made or especially adapted for use in the invention and not a staple article or commodity of commerce suitable for substantial noninfringing use, where such component is uncombined in whole or in part, knowing that such component is so made or adapted and intending that such component will be combined outside of the United States in a manner that would infringe that patent if such combination occurred within the United States, shall be liable as an infringer.

35 U.S.C. § 271(f)(2) (2000).

128. This provision is similar to contributory infringement liability under 35 U.S.C. § 271(c), which imposes liability for selling or offering to sell a "component of a patented machine, . . . constituting a material part of the invention, knowing the same to be especially made or especially adapted for use in an infringement of such patent, and

271(f)(1) could implicate liability for software export if general purpose software was exported as part of a protection combination invention, as long as a substantial portion of the components were exported. Conversely, § 271(f)(2) liability could arise in the software context if a specific code were developed to implement functionality included in a patented invention, even if the patent enjoyed no international protection and the competing device were manufactured outside the United States. Because liability under this section is different from traditional infringement provisions, a proposal for bounding it in the context of software inventions under § 271(f) is presented in Part V of this Comment.

iv. Appellate Interpretation of § 271(f) Infringement

Cases involving patent law are under exclusive federal jurisdiction,[129] and appeals of patent cases are heard exclusively by the Court of Appeals for the Federal Circuit or the U.S. Supreme Court.[130] As a consequence, circuit splits, as frequently occur among the regional federal circuits, are nonexistent in patent law cases. This leads to more uniformity and consistency in patent law interpretation.

In the twenty years since § 271(f) was enacted, the Supreme Court has not heard any cases concerning this statute. Moreover, the Federal Circuit has only heard two significant § 271(f) cases, both within the past four years.[131] As a consequence, precedential interpretation of § 271(f) is very limited.[132] The two cases addressing § 271(f) provide limited guidance on the component question, suggesting only that § 271(f) will be narrowly construed, but may still capture a large potential component export market.

The first case, *Rotec Industries, Inc. v. Mitsubishi Corp.*,[133] was decided in 2000 and addressed interpretation of § 271(f)(2)'s component supply language.[134] Rotec argued that § 271(f)(2)'s language should be

---

not a staple article or commodity of commerce suitable for substantial noninfringing use." 35 U.S.C. § 271(c).

129. 28 U.S.C. § 1338(a) (2000).

130. 28 U.S.C. § 1295 (2000).

131. One earlier case in 1991 addressed § 271(f) minimally, summarily dismissing its applicability. *See* Standard Havens Prods., Inc. v. Gencor Indus., Inc., 953 F.2d 1360, 1374 (Fed. Cir. 1991). The relevance of this case is discussed in Section 4 of this Comment.

132. While binding precedential decisions from the Federal Circuit are limited, several district courts have heard § 271(f) cases. Relevant reported cases are addressed in Part IV of this Comment.

133. 215 F.3d 1246 (Fed. Cir. 2000).

134. *Id.* at 1257. Section 271(f)(2) imposes liability to those who supply or cause to supply any component of a patented invention. 35 U.S.C. § 271(f)(2) (2000). *Rotec* fell within the scope of § 271(f)(2) rather than § 271(f)(1) because it involved overseas use of a crane system, where several components were specifically designed for the device,

broadly interpreted to include "offer[s] to supply" based on an argument that Congress "surely intended to strengthen the patent laws" through its amendment of § 271(a).[135] The court roundly rejected this argument noting that "if Congress wanted to amend 271(f)(2) . . . it could have easily done so."[136] This suggests that the Federal Circuit is unwilling to broadly interpret § 271(f) without clear Congressional intent. As a consequence, absent express language, it appears that the Federal Circuit will not add provisions to § 271(f) that are not textually present.[137]

The second significant § 271(f) case, *Waymark Corp. v. Porta Systems Corp*.,[138] was decided in 2001 and addressed supply of components that are exported with the intention of combination, but never actually assembled abroad.[139] As noted previously, § 271(f)(2) imposes liability for "supply" of components of a protected invention.[140] Porta Systems shipped components of an infringing battery monitoring system to Mexico; however, the components were never assembled into the final product.[141] The court noted, however, that the language of § 271(f)(2) does not specify assembly abroad as an element. Therefore, adding an extraterritorial assembly requirement would improperly modify the statutory language.[142] As a result, merely supplying components, even without any subsequent assembly, is sufficient to impose liability.[143] This case again suggests that the Federal Circuit will construe the

---

rather than being staple articles of commerce. *Rotec*, 215 F.3d at 1248–49.

135. *Rotec*, 215 F.3d at 1257. Rotec sought to reinterpret § 271(f) in light of Congressional amendment of § 271(a) to include both actual sales as well as offers to sell, where an offer to sell is not explicitly codified in § 271(f)(2). *Id.* at 1258. Under this rejected broad view, liability would be imposed based on mere negotiation or agreement to supply without actual transfer from the United States. *Id.* In this case, the components never had a United States origin or transfer, being made in France, Japan, and China. *Id.* at 1259.

136. *Id.* at 1258.

137. The court noted that the supply language from § 271(f) is "clear on its face." *Id.* Textual provisions limiting supply of components are distinguishable from an "offer to supply" when no language addressing offers to supply are included in the text of § 271(f). *Id.*

138. 245 F.3d 1364 (Fed. Cir. 2001).

139. *Id.* at 1365.

140. *See infra* note 142.

141. *Waymark*, 245 F.3d at 1365.

142. *Id.* at 1368. The court is stating that liability is imposed when someone supplies the component with the intention that it be assembled abroad, not that it ultimately is assembled. This result might, however, create an anomaly related to supply of software components if the breadth of what constitutes supply is interpreted as in *Eolas*. *See infra* notes 245–47 and accompanying text.

143. *Waymark*, 245 F.3d at 1368.

language of § 271(f) narrowly and will not imply any conditions or extensions that are not textually present.

In sum, the Federal Circuit has taken a narrow textual approach towards interpreting § 271(f). As a consequence of *Waymark*, however, the door may be opened to expansion of § 271(f) liability. This may occur because the textual language of § 271(f) imposes liability in a broad, unprecedented way—by merely exporting unprotected components to an extraterritorial location where the components do not need to be assembled into an infringing device.[144] As a result, *Waymark* will undoubtedly create more scrutiny of potentially infringing exports. Moreover, with the rapid growth of software development and integration of software into almost every type of product, software components will be a natural area on which to focus this attention.

IV. A PROPOSED ROLE FOR SOFTWARE IN § 271(F) INFRINGEMENT

Against this background, is there a place for software components under § 271(f), and, if so, how should the scope of protectability be bounded? As noted previously, there is virtually no precedential case law on § 271(f) and only one significant case besides *Eolas* and *Imagexpo* that involves software.[145] Therefore, the scope of software integration into § 271(f) is open for interpretation. Among the three primary elements of § 271(f)—knowledge, component status, and supply—the knowledge requirement is not unique to software; however, component status and supply present unique issues. Therefore, incorporation of software into § 271(f) liability should focus on these two elements.

This Comment proposes that software enjoy protection as a component and should not be entirely excluded under § 271(f), as proposed by Microsoft in *Imagexpo* and *Eolas*.[146] The scope of what is protectable should, however, be narrowly construed and limited to software in an executable form that is ready to be incorporated into a product as part of a manufacturing process or its equivalent. Source code or intermediate

144. One commentator recently predicted that "[t]he door to infringement actions under the underused § 271(f) provision has been cracked open; . . . the *Waymark* decision will operate to open the door wide." Joan E. Beckner, Note, *Patent Infringement by Component Export:* Waymark Corp. v. Porta Systems Corp. *and the Extraterritorial Effect of U.S. Patent Law*, 39 HOUS. L. REV. 803, 806 (2002).

145. *See* Enpat, Inc. v. Microsoft Corp., 6 F. Supp. 2d 537 (E.D. Va. 1998). This case is discussed later in the Comment. *See infra* notes 220–22 and accompanying text.

146. Microsoft claims that a component can only be a "tangible, physical element of a patented device," a categorization it believes would exclude software. Imagexpo, L.L.C. v. Microsoft Corp., 299 F. Supp. 2d 550, 552 (E.D. Va. 2003). Moreover, rather than a "component," Microsoft claims that software "is analogous to [a chemical] formula," merely describing a repeated series of process steps. Eolas Techs., Inc. v. Microsoft Corp., 274 F. Supp. 2d 972, 973 (N.D. Ill. 2003).

software forms, such as object code unlinked with particular memory spaces, should not impose liability because they are interim results of a design, as opposed to a manufacturing process. In addition, only patents claiming a system or apparatus, as opposed to a process or method, should enjoy protection under § 271(f). Because software is frequently claimed in the method or process form, this will further limit the scope of protectability.

In addition to concerns about the scope of protection, software raises novel concerns regarding the supply element of § 271(f) liability.[147] To address these concerns, this Comment proposes that liability for supplying a software component should be imposed irrespective of the means of export, as long as the channel of supply does not irreversibly transform the software in the supply process. This allows for imposition of liability for modern electronic means of export, such as over the Internet or via wireless channels, in addition to actual physical transport and accommodates the realities of modern software distribution. As a limitation to this expansive provision, however, infringement liability should be imposed on a one-to-one basis with the software component units exported. Export of a single master disk, as was done by Microsoft in *Eolas* and *Imagexpo*, should not impose general liability for all overseas manufactured units.[148] Conversely, software components distributed via email or downloaded from a web page should impose liability for each download, but only on a one-to-one basis.

A. Why Software Should Be Considered a Component Under § 271(f)

A threshold concern is whether software can constitute a component at all for liability purposes. If software does not fit within some acceptable definition of a component, discussion of infringement liability under § 271(f) is irrelevant.

---

147. Section 271(f)(1) imposes liability on anyone who "without authority supplies . . . components of a patented invention . . . ." 35 U.S.C. § 271(f)(1) (2000). Section 271(f)(2) contains essentially the same language, imposing liability for anyone who "supplies . . . any component of a patented invention . . . ." 35 U.S.C. § 271(f)(2) (2000). Software raises issues because it can be readily distributed over the Internet, or via wireless channels, in addition to physical channels and, as a consequence, can be transformed in transmission and copied indefinitely upon receipt. Tangible parts have neither of these properties.

148. This element of the proposal is counter to the District Court's holding in *Eolas* that imposed liability for all units created from the single exported master disk. *Eolas*, 274 F. Supp. 2d at 974.

What constitutes a component under § 271(f) is not explicitly defined.[149] Moreover, few cases have dealt with defining components or bounding their scope, although one district court case, *Bristol-Myers Squibb Co. v. Rhone-Poulenc Rorer, Inc.*, proposed a context-based test.[150] Nevertheless, a general definition can be inferred from general patent law limitations and § 271(f)'s legislative history.[151] While not enjoying standalone patent

149. The term "component" is not defined in the patent statutes and is not used outside of § 271. While no explicit definition of component is given, patent law allows for common usage, as in dictionary definitions, to construe the scope of inventions. *See, e.g.*, Markman v. Westview Instruments, Inc., 52 F.3d 967, 980 (Fed. Cir. 1995), *aff'd*, 517 U.S. 370 (1996). A typical dictionary definition describes a component as "a constituent part" or an "ingredient." WEBSTER'S NEW COLLEGIATE DICTIONARY 270 (9th ed. 1989). Case law addressing related infringement provisions, such as the doctrine of equivalents, refers to components as composites of elements related to claim limitations. *See* Corning Glass Works v. Sumitomo Elec. U.S.A., Inc., 868 F.2d 1251, 1259 (Fed. Cir. 1989).

150. In typical cases, components are constituent parts that come together in the manufacturing process to make the final product. For example, in a recent case involving an artificial turf product, the court considered pile fabric that was combined with a subsurface, sand, and resilient particles to make the final product, to be a component. Fieldturf, Inc. v. Southwest Recreational Indus., Inc., 235 F. Supp. 2d 708, 733–34 (E.D. Ky. 2002) (rejecting § 271(f)(1) liability because only one of four constituent components was exported), *vacated in part on other grounds*, 357 F.3d 1266 (Fed. Cir. 2004). *But see* Moore U.S.A. Inc. v. Standard Register Co., 144 F. Supp. 2d 188, 195 (W.D.N.Y. 2001) (holding that paper and glue are components of a protected document assembled abroad). Most § 271(f) cases merely assume that parts constituting less than the whole are components. Perhaps the closest attempt at defining the extent of components was made in a recent pharmaceutical case where the issue of chemical compound divisibility was raised. In this case, a claimed element included a chemical ester, of which no subdivisions were identified in the patent itself or its prosecution history, known in patent parlance as the file wrapper. To determine whether precursor elements of this compound were components, the court referenced a hypothetical person of ordinary skill in the art (POSA). If this person would recognize that the compound had constituent parts, the court would conclude that these were components for § 271(f) liability purposes. Bristol-Myers Squibb Co. v. Rhone-Poulenc Rorer, Inc., No. 95 Civ. 8833, 2001 U.S. Dist. LEXIS 16895, at *9–10 (S.D.N.Y. Oct. 19, 2001), *aff'd*, 326 F.3d 1226 (Fed. Cir. 2003). This suggests a context-based, intuitive interpretation. If a person having ordinary skill in a particular art would recognize that an invention was composed of constituent parts, these constituent parts would be its components. One recent case attempted to greatly broaden the reach of liability by arguing that merely having a U.S. facility, where design and development of integrated circuit chips was presumably done, implicated liability even when the chips were unquestionably manufactured outside the United States. The plaintiff argued that "the chips, in some metaphysical sense, [came] from the United States [because instructions for their disposition also came from the United States]." This attempt failed at the summary judgment stage. Pellegrini v. Analog Devices, Inc., No. 02-11562-RWZ, 2003 U.S. Dist. LEXIS 7598, at *1–2 (D. Mass. May 7, 2003), *aff'd*, 375 F.3d 1113 (Fed. Cir. 2004).

151. Reference to legislative history is an accepted means of statutory interpretation. "When the language of a statute fails to provide clear and unambiguous direction, we may turn to the statute's legislative history." Rotec Indus., Inc., v. Mitsubishi Corp., 215 F.3d 1246, 1252 (Fed. Cir. 2000) (citing Toibb v. Radloff, 501 U.S. 157, 162 (1991) ("[W]e look first to the statutory language and then to the legislative history if the statutory language is unclear.")).

protection, a component should be something that could conceivably enjoy statutory patent protection.[152] Implicitly, a component must be an element of something less than a fully protectable invention.[153] In the patent realm, a component should relate in some way to the claims of the infringed invention, rather than the invention's actual embodiment.[154] Moreover, because § 271(f) liability arose in a specialized context, product manufacturing, a component should be in the form in which it will actually be used in an assembly process rather than in the form of a predecessor design element, such as source code.[155]

---

152. While this requirement is not explicit, it can be reasonably implied by making an analogy to statutory subject matter limitations. These limitations preclude protection for certain subject matter and it would seem irrational to bar patentability for a particular subject matter, but still impose liability when that subject matter is merely part of a larger apparatus invention. *See supra* notes 59–60 and accompanying text. In any event, at this point in time, this limitation does not impose a barrier to liability for software components under § 271(f) because software itself has achieved a high degree of acceptance as patentable subject matter. *See supra* notes 67–83 and accompanying text.

153. The language of § 271(f) refers to a component or components as a subset of some patentable invention and also references extraterritorial combination so as to produce the protectable whole, thereby implying that a component must be something less than a protectable invention. *See* 35 U.S.C. § 271(f) (2000). Moreover, the legislative history notes that § 271(f) was intended to overrule the *Deepsouth* practice of combining an unassembled, and consequently unprotected, combination extraterritorially. *See supra* note 123 and accompanying text; *see also infra* note 160.

154. Patent protection is limited to the claims defining the scope of the invention, not a particular embodiment of parts of an invention that does not include all of the claims. Section 271(f) specifically addresses this concern by requiring that a component combined outside the United States "would infringe the patent [only] if such combination occurred within the United States [and was a direct infringement]." 35 U.S.C. § 271(f)(1). This is related to the general rule limiting a patent's scope to its claims which must "particularly point[] out and distinctly claim[] the subject matter . . . [of the] . . . invention." 35 U.S.C. § 112 (2000). This tie in between components and claim elements was noted by the Federal Circuit in a related case which stated that an "'[e]lement' may be used to mean a single [claim] limitation, but it has also been used to mean a series of [claim] limitations which, taken together, make up a component of the claimed invention." *Corning*, 868 F.2d at 1259.

155. This requirement can be inferred by considering the distinction between manufacturing and product design and development. Design and development is the process of developing and transforming an idea into a product. This process deals with the creative aspect of a product's birth, whereas manufacturing deals with production based on "an organized plan and . . . division of labor." WEBSTER'S NEW COLLEGIATE DICTIONARY 725 (9th ed. 1989). The design and development process, which was never included in § 271(f)'s case law or legislative history, should create a logical boundary on what constitutes a manufacturing component because nothing in the case law or legislative history suggests that § 271(f) should extend beyond manufacturing or assembly of patented inventions. *See, e.g.*, Patent Law Amendments Act of 1984, Pub. L. No. 98-622, 1984 U.S.C.C.A.N. (89 Stat. 3383) 5827; Deepsouth Packing Co. v. Laitram Corp., 406 U.S. 518, 532 (1972). Moreover, any contrary interpretation of

Software, at least in some forms, clearly meets these requirements, strongly suggesting it should enjoy component status under § 271(f).[156] First, while initially being questioned as patentable subject matter, software has become widely accepted in both apparatus and method form.[157] Because software itself is protectable in apparatus form, it follows that, applying the Person of Skill in the Art (POSA) test from *Bristol Myers*,[158] any distinct subprograms or software elements would then be components. This POSA test can also be extended to inventions composed of both hardware and software to reach the same result: that distinct software elements of composite inventions are also components.[159] In either of these situations, an individual program, even if it does not explicitly enjoy protection because it lacks statutory requirements such as novelty or nonobviousness, would still be a constituent part of the protected invention and, therefore, be a component.[160]

Software can also be tied specifically to the patent's claims by implementing claimed features. Moreover, software in many contemporary inventions can be directly interchangeable with mechanical or electronic

---

§ 271(f) would be fundamentally at odds with general patent law limitations on scope of protection and territory.

156. Prior to *Eolas*, only one significant reported case, *Enpat v. Microsoft Corp.*, addressed software as a component and in that case it was rejected, although for other reasons. *See* Enpat, Inc., v. Microsoft Corp., 6 F. Supp. 2d 537, 539 (E.D. Va. 1998) (rejecting application of § 271(f) to method patents, which have no "component"). Despite the lack of software component cases, at least a few commentators have suggested, without significant elaboration, that software might impose component liability under § 271(f). *See* STOBBS, *supra* note 29, at 660–62 (suggesting but rejecting software component liability under § 271(f) in an Internet example); Keith E. Witek, *Software Patent Infringement on the Internet and on Modern Computer Systems—Who is Liable for Damages?*, 14 SANTA CLARA COMPUTER & HIGH TECH L.J. 303, 335 (1998) (suggesting that software elements of a protected program might enjoy component status under § 271(f) when transmitted abroad time serially).

157. *See supra* notes 67–84 and accompanying text; *see also* STOBBS, *supra* note 29, at 153–87.

158. This test looks at the invention or its subparts in light of a person skilled in the art. If he would view the invention as having separate parts, these constitute a component. Bristol-Myers Squibb Co., v. Rhone-Poulenc Rorer, Inc., No. 95 Civ. 8833, 2001 U.S. Dist. LEXIS 16895, at *9–10 (S.D.N.Y. Oct. 19, 2001), *aff'd*, 326 F.3d 1226 (Fed. Cir. 2003); *see also supra* note 150.

159. This situation is common in embedded systems which are typically composed of analog and digital hardware, interface elements such as sensors, and associated software. All of these elements can be part of an apparatus patent's claims.

160. To further elaborate, now that software has been accepted as patentable subject matter, *see supra* notes 67–84, it follows that a software invention could itself be a composite invention, composed of multiple independent software parts. If the aggregate may enjoy protection, then the separate elements, even if not independently protectable, should be considered components because when added together they comprise the whole, just as the machine parts in *Deepsouth*, when put together, constituted the complete invention.

functions that would clearly enjoy protection under § 271(f).[161] This is perhaps best seen in the context of embedded systems. Embedded systems are devices or products that use processors and software internally without the user's knowledge.[162] Unlike desktop computers, the processing function is hidden from view and is effectively transparent to the user. Examples abound, with use of embedded processors expanding rapidly.[163] For example, some newer automobiles include over one hundred processors performing a variety of functions, many of which were traditionally done with hardware.[164] The consequence of this

161. Hardware and software, particularly in embedded application, are frequently interchangeable. The identical function provided by software can be built directly into hardware in the form of custom ICs or programmable hardware devices like FPGAs, further blurring the boundaries. The choice of whether to use hardware or software is often driven more by cost or functionality than any inherent distinction between hardware and software. *See generally* Lara Simsic, *Accelerating Algorithms in Hardware*, EMBEDDED SYSTEMS PROGRAMMING, Feb. 2004, at 24.

162. While definitions may vary in technical details, the term "embedded systems" is widely used in the hardware and software development communities to refer to special purpose embedded processing systems that are typically comprised of a microcontroller or microprocessor, associated software (typically in the form of firmware), and analog and digital I/O. *See* Definitions of Embedded System on the Web, Google, *at* http://www.google.com/search?hl=en&lr=&oi=defmore&q=define:EMBEDDED+SYSTEM (last visited Feb. 9, 2005) (linking to a variety of definitions, all including the basic elements of special purpose hardware, software, and I/O). A number of software development publications focus on embedded software development. One of the most popular is *Embedded Systems Programming* magazine and its associated website, located at http://www.embedded.com (last visited Feb. 9, 2005). Other popular publications covering embedded systems design and development include *EDN* magazine, located at http://www.edn.com (last visited Feb. 9, 2005), and *Embedded Computing Design*, located at http://www.embedded-computing.com (last visited Feb. 9, 2005).

163. Processors are being incorporated in a hidden fashion into a huge array of devices. Examples range from consumer products like toys, watches, calculators, televisions, and audio equipment to less conventional applications like toasters, refrigerators, and even clothing. Most products that have some type of built-in intelligence implement it with embedded processors and their associated software. *See infra* notes 164–65 for some examples. In the interest of full disclosure it should be noted that, prior to attending law school, the author, an electrical engineer, worked extensively with embedded processor technology, designing and developing a number of embedded systems products, several of which are covered by U.S. patents. *See, e.g.*, U.S. Patent No. 6,609,431 (issued Aug. 26, 2003); U.S. Patent No. 5,999,889 (issued Dec. 7, 1999); U.S. Patent No. 5,233,537 (issued Aug. 3, 1993).

164. A Princeton University electrical engineering professor references these prolific uses in an online presentation. *See* Wayne Wolf, *Computers as Components*, *at* http://www.ee.Princeton.edu/~wolf/EE464/Overheads/ch1-1.ppt (2000). In 1970, a typical car had about $75 in electronics, mostly in the form of the radio. Today, the average car has $2000 worth of microprocessors and related hardware, controlling everything from the brakes to the power windows. On some luxury cars the total processor cost can approach $10,000. These systems improve almost every aspect of a

growth of embedded systems is that software is increasingly used to replace mechanical and electronic functions previously implemented in hardware as well as to implement new, previously impractical functions. It would be unrealistic to claim that a particular system would contain protectable components if they were in a mechanical or electronic form but not if they were software-based.[165]

Software can also be readily incorporated into a product as part of a standard manufacturing process. In this context, software programs are treated as merely another part of the assembly process. As an example, software for an embedded processor installed on a product's circuit board might be transmitted by a programmer in disk form, CD form, or via the Internet to a manufacturing facility. The production line would then assemble the electronics and mechanical housing. The software would then be loaded, via a production line device programmer, into the microcontroller. This would normally be followed by a test stage and packaging, which completes the manufacturing process. In this typical example, software is integrated—as a component of the completed product—in a way that is essentially indistinguishable from other elements of a device; this suggests that software should be treated the same as any other tangible, physical component.

Despite the fact that software has generally become acceptable as patentable subject matter and software components can be closely analogized to similar mechanical or electronic ones, several arguments suggest that software should not be treated as a component under § 271(f). First, § 271(f) was enacted specifically in response to *Deepsouth*, which involved only manufacture, sale, and assembly of mechanical machine parts potentially infringing on an apparatus patent.[166] This limitation was noted twelve years later, when § 271(f) was enacted, and the limited legislative history clearly states it was made in direct

---

vehicle's operations and have been particularly important in improving safety. Paul A. Eisenstein, *Silicon Chip Takes Over the Auto*, DETROIT NEWS, *at* http://www.detnews.com/specialreports/2000/goldenage/silicon/silicon.htm (Jan. 10, 2000).

165. A good example of this is fly–by-wire systems in modern aircraft. Flight control was traditionally maintained through control surfaces, such as rudders or ailerons, that were mechanically linked to the pilot's control yolk. Contemporary aircraft, including both passenger and high performance military jets, replace this mechanical and hydraulic linkage with fly-by-wire systems comprised of processing software at the flight yolk and control surface that performs the same functions provided by the mechanical linkage. *See, e.g.*, Airbus Media Centre, Our Advantages: Fly-by-Wire, *at* http://www.airbus.com/media/fly_by.asp (2005) (explaining Airbus's fly by wire system used on commercial jets). Numerous other examples of this functional replacement role of software abound. Moreover, it would not be surprising today to see some of the functions implemented by the shrimp deveining machinery in *Deepsouth* replaced or enhanced by processors and software.

166. Deepsouth Packing Co. v Laitram Corp., 406 U.S. 518, 519–23 (1972).

response to *Deepsouth* to deter manufacturers acting as copyists.[167] A narrow reading of the statutory provision based on this legislative history might imply that liability should be limited to tangible, mechanical parts of machines such as the boxed parts of the shrimp processing machinery at issue in *Deepsouth*. Several reported district court § 271(f) cases have raised similar arguments in the context of chemical exports.[168] For example, in *W.R. Grace & Co. v. Intercat, Inc.*, Intercat claimed that because the allegedly infringed patent and component was for a chemical composition, rather than a mechanical or structural apparatus, § 271(f) did not apply.[169] This argument failed. The court roundly rejected a mechanical-only interpretation, stating that "[n]owhere in the statute or its legislative history is there a limitation to components of machines and other structural combinations."[170] Therefore, refusal to apply the statute to a chemical composition patent, merely because the underlying chemical components are not mechanical parts, "would be tantamount to legislating additional language to a statute" that arbitrarily precludes some categories but not others.[171] Extending protection beyond mechanical to chemical components, but not to software, would create an arbitrary, unsupported, and artificial limit.

Microsoft has raised similar arguments in *Eolas* and *Imagexpo* that have, to date, proven similarly unsuccessful. Microsoft argued that its exported software is "a template, similar to a design, instruction, or recipe."[172] Alternately, Microsoft claimed that the software was distinguishable from the chemicals in *W.R. Grace* because it is analogous to

167. *See supra* notes 119–24 and accompanying text.

168. *See, e.g.*, W.R. Grace & Co. v. Intercat, Inc., 60 F. Supp. 2d 316, 320 (D. Del. 1999); Lubrizol Corp. v. Exxon Corp., 696 F. Supp. 302, 312 (N.D. Ohio 1988). These two cases were cited by the District Court's memorandum opinion in *Eolas* as "[t]he closest cases" to the one at issue. Eolas Techs., Inc. v. Microsoft Corp., 274 F. Supp. 2d 972, 973 (N.D. Ill. 2003).

169. *W.R. Grace*, 60 F. Supp. 2d at 320. This argument rested on a narrow reading of § 271(f)'s legislative history. Intercat argued that § 271(f)(2) did not apply because the legislative history "'states that the statute only covers components of machines and other structural combinations, since the section was enacted specifically to overrule' *Deepsouth*." *Id.* (quoting defendant Intercat's pleadings).

170. *Id.* at 321. The court in *Lubrizol* rejected a similar argument; however, it did not provide any analysis for its decision. *Lubrizol*, 696 F. Supp. at 325 (enjoining supply of chemical components of a lubricating compound outside the United States under § 271(f)).

171. *W.R. Grace*, 60 F. Supp. 2d at 321.

172. Imagexpo, L.L.C. v. Microsoft Corp., 299 F. Supp. 2d 550, 552 (E.D. Va. 2003).

the chemical formula rather than the chemical itself.[173] Under this view, software is not a tangible, physical object and, therefore, cannot be included as a component in an apparatus or device.[174] However, this argument again fails to recognize that software can be both conceptually and functionally more than merely a process or formula. A particular instance of executable code, as stored in a product's memory, can be considered a standalone entity,[175] and its final result can be indistinguishable from a comparable mechanical or electronic component that would clearly enjoy protection. A formula, on the other hand, merely provides instructions for creating the component, rather than implementing the functionality itself.

Another potential argument against protection of software rests on the recent Federal Circuit decision in *Bayer AG v. Housey Pharmaceuticals, Inc.*,[176] decided under 35 U.S.C. § 271(g), a statutory provision related to § 271(f). *Housey* involved a drug discovery research tool that was covered by a U.S. process patent.[177] Under § 271(g), infringement occurs if the resulting product of the patented process is imported into the United States.[178] The process at issue in *Housey* produced data assessing the efficacy of a chemical compound for pharmaceutical purposes, rather than a tangible item such as a chemical compound or drug.[179] The Federal Circuit rejected protection for this data alone based primarily on its interpretation of what constitutes manufacturing for purposes of statutory interpretation.[180]

---

173. *Eolas*, 274 F. Supp. 2d at 973–74.

174. Microsoft relies primarily on *Enpat* for this argument. *Imagexpo*, 299 F. Supp. 2d at 552. However, *Enpat* specifically dealt with only a process or method patent with no component or apparatus claims, whereas the potential infringement here rests on construing the software as a part of a patented component invention. Enpat, Inc. v. Microsoft Corp., 6 F. Supp. 2d 537, 539 (E.D. Va. 1998).

175. To paraphrase, when code gets loaded into a device, it exists uniquely in every instance of that device that is assembled or manufactured. This distinguishes it from a formula or template which is typically limited in quantity to one or a few instances and exists outside the resultant product of the template or formula. As the court in *Imagexpo* noted, in the manufacturing process, the software, in the form of executable code, "becomes an integral ingredient in the finished computer product." *Imagexpo*, 299 F. Supp. 2d at 553.

176. 340 F.3d 1367 (Fed Cir. 2003).

177. *Id.* at 1369.

178. The language of § 271(g) states that "[w]hoever without authority imports into the United States . . . a product which is made by a process patented in the United States shall be liable as an infringer, if the importation . . . occurs during the term of such process patent." 35 U.S.C. § 271(g) (2000).

179. *Housey*, 340 F.3d at 1369, 1377. The process at issue is a "method of determining whether a substance is an inhibitor or activator of a protein whose production by a cell evokes a responsive change in a phenotypic characteristic other than the level of said protein in said cell per se . . . ." *See* claim 1 of U.S. Patent No. 4,980,281 (issued Dec. 25, 1990).

180. *Housey*, 340 F.3d at 1377.

The legislative histories of § 271(f) and § 271(g) are somewhat related.[181] The Federal Circuit in *Housey* noted that even if the legislative history does not demonstrate a specific intent to limit the scope of coverage of § 271(g) (and possibly by analogy § 271(f)), nothing suggests "that Congress was concerned that the preexisting statutory scheme failed to reach intangible information, or that the substantive coverage of the Act, as opposed to the available remedies, was to be expanded."[182] This distinction rests on the court's emphasis on extraterritorial manufacturing as a basis for enactment of § 271(g),[183] as well as its interpretation of what constituted manufacturing. The *Housey* court construed manufacturing narrowly, limiting its results to "physical goods that were manufactured and does not include [intangible information, such as data,] generated by a patented process . . . ."[184] This suggests that the Federal Circuit might view components under § 271(f) as limited to tangible items and, therefore, preclude protection for intangible elements of a combination invention, such as data or software components. While this case is probably the closest the Federal Circuit has come to interpreting § 271(f) in the software context, it is still distinguishable on several points. First, *Housey* deals with a separate statutory provision related to process, rather than apparatus, patents. As at least one district court case has noted, § 271(f) does not apply to process patent claims.[185] In addition, while the legislative histories of § 271(f) and § 271(g) are somewhat related, they deal with distinct concerns and were ultimately enacted during different legislative sessions, with § 271(g) receiving much more elaborate legislative consideration based on a variety of policy concerns not implicated by § 271(f). Section 271(f)'s legislative history, on the other hand, suggests a

181. What eventually became § 271(g) was initially proposed in 1983 in both the House and Senate. A subsequent bill in the following year again proposed a precursor of § 271(g): "[i]f the patent invention is a process, whoever without authority uses or sells within, or imports into, the United States during the term of the patent therefor a product produced by such process infringes the patent." *Id.* at 1374 (quoting S. REP. NO. 98-663, at 30 (1984)).

182. *Housey*, 340 F.3d at 1374.

183. The court noted that "[e]ach and every reference to the provision that became section 271(g) describes it as directed to manufacturing." *Id.* Moreover, the introductory language to the 1984 amendments specifically states that it is intended to "avoid encouraging manufacturing outside the United States." Patent Law Amendments Act of 1984, Pub. L. No. 98-622, 1984 U.S.C.C.A.N. (89 Stat. 3383) 5827.

184. *Housey*, 340 F.3d at 1368.

185. Enpat, Inc. v. Microsoft Corp., 6 F. Supp. 2d 537, 539 (E.D. Va. 1998).

very limited response to a single case, *Deepsouth*. Finally, § 271(g) deals with the output product of a patent protected process, whereas § 271(f) deals with the component inputs of a manufacturing process.[186] Consequently, the *Housey* Court's narrow reading of the scope of protection under § 271(g) should not provide a strong inference that § 271(f) should be similarly construed, or that it should preclude components such as software that can clearly be an input component to a manufacturing process.

B. The Scope of Protectable Components

Assuming, as proposed, that software is considered a component under § 271(f), the next question is what constitutes protectable software for liability purposes under § 271(f). Dictionary definitions of software are very broad, typically including "the entire set of programs, procedures, and related documentation associated with a . . . computer system."[187] Software has also been described as "what empowers a computer to handle information and to control information flow."[188] These definitions provide a general, amorphous scope. As a consequence, export of any of these forms of software could potentially impose liability.

The key to assessing the scope of protectability under § 271(f) lies in examining its legislative purpose in light of the general scope of protection provided by patent law. Protection should be narrowly construed under this analysis, in light of the limited purpose behind § 271(f), to reach only components used as part of a manufacturing process or its equivalent. This limitation can be justified for at least two reasons. First, as discussed previously, § 271(f) was enacted in direct response to a perceived patent protection problem associated with the manufacturing process.[189] The legislative history clearly indicates that

---

186. 35 U.S.C. § 271(f)–(g) (2000).

187. WEBSTER'S NEW COLLEGIATE DICTIONARY 1120 (9th ed. 1989). One commentator recently summarized various reference definitions including:

> computer programs and applications . . . that can be run on a particular computer system . . .[,] something used or associated with, and usually contrasted with hardware . . .[, t]he programs, routines, and symbolic languages that control the functioning of the hardware and direct its operation . . .[,] the instructions which control what a computer does . . .[, and] the instructions that control what a computer can do.

GREGORY A. STOBBS, SOFTWARE PATENTS: 2004 CUMULATIVE SUPPLEMENT 9–10 (2004). From these definitions, "it is clear that the term 'software' means many things to many people." *Id.* at 10.

188. STOBBS, *supra* note 29, at 49.

189. *See supra* notes 118–23 and accompanying text.

the scope should be limited to this specific situation.[190] A second justification for a narrow interpretation lies in the general purposes and limitation of patent protection. As a rule, patent protection is limited in claim scope as well as territorially.[191] Section 271(f) liability is counter to these fundamental patent limitations because it extends liability to extraterritorial acts and unassembled invention components that do not infringe a patent's claims. Therefore, § 271(f) constitutes a narrowly limited exception to these general rules and should be construed accordingly.[192]

i. Source versus Executable Code Forms

Construing software narrowly under this approach, the question of scope of protection then becomes one of determining what forms of software are usable as components in a manufacturing process. While software definitions are broad, they can be divided into two distinct categories: source code and executable code.[193] While liability could conceivably be imposed for export of component forms falling within either of these categories, protection under § 271(f) should be limited to software in the executable form. This limitation is appropriate because

190. While limited, the legislative history states that "[t]his proposal responds to the United States Supreme Court decision in *Deepsouth Packing Co. v. Laitram Corp.*," which dealt with the narrow issue of protectability of unassembled components in a manufacturing situation. Patent Law Amendments Act of 1984, Pub. L. No. 98-622, 1984 U.S.C.C.A.N. (89 Stat. 3383) 5827, 5828. The legislative history also states that § 271(f) was enacted to "prevent copiers from avoiding U.S. patents by supplying components . . . so that the assembly of the components may be completed abroad." *Id.* This strongly implies a manufacturing only context.

191. *See supra* notes 32–36 and accompanying text.

192. A general rule of statutory interpretation is that a narrow, specific statutory enactment prevails over a more general one when they are in conflict. *See, e.g.*, Missouri v. Ross, 299 U.S. 72, 76 (1936) (specific statute prevails over general, citing additional authority).

193. One noted computer science scholar discussed software in these two forms:

> [P]rograms have two different sorts of manifestations. On the one hand, they are documents of some kind that give a series of instructions [—such as source code and associated instructions, etc.—] to be executed by a computer. But these passive documents can be turned into active physical processes: when a program is executed, the instructions in the document are carried out. . . . Does "program" [or software] refer to the passive text, or the active event?

GELERNTER & JAGANNATHAN, *supra* note 61, at 1–2. The documents are representative of what is known as source code, the textual description of what the computer hardware is to perform. The actual program execution is carried out by executable code, which can be represented in a textual form but directly represents what operations are being executed on the hardware itself in the exact form used by a processor.

only this form can be directly incorporated into a product as part of a manufacturing, rather than design or development, process.[194]

Source code is the textual description of a computer program's operation, typically written in a standardized computer language such as C or C++.[195] A programmer creates software by converting general functional requirements—as embodied in software specifications, design documents, flowcharts, and other high level planning tools—into one or more source code text files or modules.[196] Software development is typically an interactive process where source code is converted to object code[197] via a compiler, and then linked with other object code modules into executable code.

Some arguments suggest source code might be considered a component for § 271(f) liability purposes. The source code represents the underlying ideas implemented by the resulting executable code. As a consequence, it includes all the algorithms, procedures, functions, and features in the resulting object code and executable program.[198] Therefore, it represents the final desired result and is used to directly create the final executable code. Specific source code is frequently incorporated into patent applications as a represented best mode embodiment of the invention.[199] In addition, there is a very close relationship between source code and

194. The critical distinction here is between design and manufacturing. Design is defined as "to conceive and plan out in the mind," or "to devise for a specific function or end." WEBSTER'S NEW COLLEGIATE DICTIONARY 343 (9th ed. 1989). Nothing in § 271(f)'s legislative history or language suggests that the scope should reach beyond the assembly stage to the mental process of design or its byproducts. Attempts to extend § 271(f) protection beyond the manufacturing context are beyond the case law and legislative history motivating § 271(f).

195. Source code can include multiple text files, including program functions, macros, and compiler instructions that direct how the separate files are combined to create binary of executable code. Source code is human readable following a precisely defined language syntax that is "readily understood by skilled programmers." Conversely, computers cannot understand source code, which is "incomprehensible to a computer." ROBERT P. MERGES ET AL., INTELLECTUAL PROPERTY IN THE NEW TECHNOLOGICAL AGE 539, 541 (2002).

196. For a concise summary of common software development processes, see *id.* at 531–39.

197. Object code is an intermediate software form, closer in nature to executable code than source code. An object code module contains the translated source code, but is in a form that may not be directly executable because it is typically missing address information that is added during a later process known as linking. *See* Byte Craft Limited, Glossary of Embedded Systems Terminology, *at* http://www.bytecraft.com/embeddedglossarysection.html (2005) (defining object code).

198. This degree of specificity would normally be sufficient to grant patent rights over any implementation of the described functionality. *See* STOBBS, *supra* note 29, at 277–89.

199. An additional requirement for patentability is disclosure of a best mode, or the best way known to the inventor for making the invention at the time of application. 35 U.S.C. § 112 (2000); *see also* STOBBS, *supra* note 29, at 265–66, 277–80.

the final executable code.[200] As part of the code compilation process, a complete source code package can be converted by anyone cognizant in using programming tools into the associated object code with minimal effort.[201]

As an example, suppose a programmer creates software to implement one claimed function of a patented device for the international market knowing that it would constitute direct infringement under 35 U.S.C. § 271(a) if made, used, or sold in the United States.[202] This scenario could implicate § 271(f) liability; however, the programmer could test the software in a way that would avoid direct infringement,[203] verify that it operated adequately, then email the source code files to another programmer outside the United States—where it could be readily recompiled into the downloadable executable code. But should source code be considered a component in this scenario and impose liability?[204]

Consider, instead, a variation on this hypothetical. Suppose, in the example above, that instead of sending the source code abroad, she

200. The source code files are translated into the processor specific instructions via the linker and compiler. Once configured, this translation process can appear to be entirely automated. It does, nevertheless, require detailed configuration of the compiler and linker to create the final result. As a consequence, this distinction should place source code on the design side of the design or manufacture boundary.

201. The extent of effort required depends on how many files are included in the source code project and what specific compiler and linker directives need to be set to complete the build of the final executable code. In some cases, a programmer would need to set many output specific parameters; in other cases, he might only need to configure the compiler with the appropriate file names and directories. All of these steps, however, involve some degree of design in configuration of the compiler and linker. Therefore, they should be considered a design, rather than manufacturing, effort and, consequently, any input to the compiler should not be considered a component because it cannot be used directly in a manufacturing process.

202. Domestic liability could be imposed for any of these activities. *See* 35 U.S.C. § 271(a) (2000).

203. Direct infringement could be avoided as long as she does not assemble all the claimed elements of the protected invention. *See supra* notes 90–96 and accompanying text. Testing of individual components or subassemblies could avoid infringement in this way. Moreover, the prototype assembly could be designed around a valid patent by using common patent prosecution techniques. Courses teaching these techniques are offered by several patent practitioners. *See, e.g.*, Patent Resources Group, Inc., "Designing Around" Valid U.S. Patents, *at* http://www.patentresources.com/advancedl04/adv_design.html (2004).

204. Under the proposed limitation excluding source code from protection, no liability would be incurred in this case under § 271(f) because the recipient would still need to complete the design process by invoking the compiler to generate executable code. What is exported is not a protectable manufacturing component, but rather precursor elements of the design process.

merely compiles it into its final executable form. This action itself, if done carefully, might not impose direct or indirect infringement liability.[205] But suppose that the programmer now emails the executable file to an overseas manufacturer for integration into an otherwise legal product. Is this example distinct from the previous one, and should it result in liability under § 271(f)?[206]

Several distinctions between these two scenarios suggest that the source code should not enjoy protection whereas the executable code should. First, the source code can be more closely analogized to a formula for the executable component rather than the component itself. As a formula, the source code would not enjoy protection under § 271(f).[207] Following this analogy, one commentator describes source code merely as a "detailed blueprint" for assembling the program's building blocks, the machine instructions, into the final executable code.[208] Second, the source code is unable to be directly integrated into a product without external transformation through human interaction with software development tools, including a compiler and linker.[209] This process makes a fundamental transformation of the source code, converting it from a form merely representative of the desired functionality to a completely different form that is incorporated directly into the invention to implement that functionality. These processes should therefore be viewed as a hard boundary between the design and development process and the manufacturing process. The programmer's work up to generation of the final output, in the form of executable code, is a design, rather than manufacturing effort. Because § 271(f) was enacted to counter a manufacturing, rather than design problem, it should be limited accordingly.

Executable code, on the other hand, represents the final embodiment

---

205. As the Court in *Deepsouth* noted, "[n]o wrong is done the patentee until the combination is formed. . . . [The] monopoly does not cover the manufacture or sale of separate elements capable of being, but never actually, associated to form the invention." Deepsouth Packing Co. v. Laitram Corp., 406 U.S. 518, 529 (1972). Therefore, testing alone would not impose liability unless all the components were combined in a way to include all the claimed elements. If the patented invention includes mechanical components, merely simulating them in software might not constitute assembly. Moreover, there would be no potential indirect liability absent a direct infringement.

206. In this case, the executable code should enjoy protection because it is in a final form and can be readily integrated into a manufactured product without any further changes.

207. Of the relatively few cases addressing § 271(f), they do seem to share the view that formulas and their equivalent do not enjoy protection. For example, in the chemical context, see *W.R. Grace & Co. v. Intercat, Inc.*, 60 F. Supp. 2d 316, 319–21 (D. Del. 1999).

208. STOBBS, *supra* note 29, at 278–81.

209. While the effort required to configure a compiler and linker may be minimal, it still involves some degree of creative input to generate the final executable code.

of the software design and development process. It can be directly loaded into a device's memory, whether it be flash memory on a processor, a separate memory chip, or a hard drive.[210] Software in executable form is processor specific and constitutes the instructions exactly as used during program execution.[211] Manufacture of embedded products typically includes a step where the executable code is loaded, on a device-by-device basis, into the appropriate memory. This process is conceptually very similar to putting mechanical components together. The mere fact that the assembled element is in the form of software, as opposed to hardware, should make no difference in a liability assessment.[212]

ii. Process versus Apparatus Claim Distinctions

In addition to limitations based on the form of software alleged to be infringing, § 271(f) implies that liability be conditioned on the form of the underlying patent as defined by the claims. Under this analysis, patents protected by apparatus claims should enjoy § 271(f) protection whereas process-only patents should not.

As noted previously, § 271(f)'s legislative history suggests it was a

210. These examples represent typical means of storing executable code into a device. Most embedded devices with processors use microcontrollers which include onboard memory in the form of flash or EPROM. *See generally* MYKE PREDKO, PROGRAMMING AND CUSTOMIZING THE 8051 MICROCONTROLLER 2–34 (1999).

211. *See* Wikipedia, Executable, *at* http://en.wikipedia.org/wiki/Executable (last modified Dec. 26, 2004) (defining executable code forms).

212. As a practical matter, exclusion of software as a protectable component could, in effect, serve to remove all protection from devices that use software, at least under § 271(f)(1). Under § 271(f)(1), liability is precluded unless all or substantially all of the components are exported. If software provides any significant functionality, liability would not lie under § 271(f) for export of hardware only because it would probably not be considered substantially all of the components. One recent case addressed this concern in finding that omission of any significant hardware component would preclude liability. *See* Fieldturf, Inc., v. Southwest Recreational Indus., Inc., 235 F. Supp. 2d 708, 733 (E.D. Ky. 2002), *vacated in part on other grounds*, 357 F.3d 1266 (Fed. Cir. 2004). *Fieldturf* involved export of one out of four components of an artificial turf product. The court referenced the plural language of § 271(f)(1) noting that "[t]he language of § 271(f)(1) speaks in terms of 'components' (plural) that are combined or are to be combined as opposed to . . . § 271(f)(2) which imposes liability in connection with . . . [supply] of a singular component." *Id.* Conversely, "exporting a single component . . . of a patented invention abroad will not rise to the level of providing 'a substantial portion of the patented invention.'" *Id.* (citing 35 U.S.C. § 271(f)(1) (2000)). Hence, under § 271(f)(1), export of just the software element, even if it provides key functionality to the device, would not impose liability.

narrowly tailored response to *Deepsouth*, which involved a combination or apparatus patent as distinguished from a process or method patent.[213] This distinction is critical in assessing the scope of protection. Product claims are specifically defined in terms of their elements or components,[214] whereas process claims merely involve a "series of acts performed in order to produce a given result."[215] As a consequence, process claims do not have components in a § 271(f) sense.[216] This distinction was first suggested by the Federal Circuit in a 1991 case that involved export sales of a machine for making asphalt, *Standard Havens Products, Inc. v. Gencor Industries, Inc*.[217] In *Standard Havens*, the process for making asphalt was claimed, but the patent did not include apparatus claims.[218] Without any significant analysis, the Federal Circuit dismissed potential § 271(f) liability.[219]

The same issue was raised in the only significant previously reported § 271(f) case involving software, *Enpat, Inc., v. Microsoft Corp*.[220] In

213. The Supreme Court in *Deepsouth* repeatedly emphasized the composite nature of the shrimp peeling apparatus and exempted export liability because an apparatus invention could not be literally infringed unless it was either made, in complete form, or used within the United States. *See supra* notes 99–118 and accompanying text.

214. When a claim is drafted in product or apparatus form "it is defined in terms of its structural elements." SCHECHTER & THOMAS, *supra* note 24, at 293.

215. *Id.*

216. Another form of patents is design patents. These patents are more like copyrights in that they protect "any new, original and ornamental design for an article of manufacture . . . ." 35 U.S.C. § 171 (2000). As such, they are merely "ornamental in character. If the design is dictated by the performance of the article, then it is judged 'primarily functional' and ineligible for design patent protection." SCHECHTER & THOMAS, *supra* note 24, at 310. While design patents do not typically implicate software elements because they exclude functionality, at least one court has rejected any § 271(f) protection for design patents because they have no component parts to assemble. *See* Aerogroup Int'l, Inc. v. Marlboro Footworks, Ltd., 955 F. Supp. 220, 232 (S.D.N.Y. 1997).

217. 953 F.2d 1360 (Fed. Cir. 1991), *rev'd on other grounds*, 1993 U.S. App. LEXIS 11963 (Fed. Cir. May 21, 1993).

218. *Id.* at 1374.

219. The court first analyzed direct and contributory infringement liability. Process claims require actual use of the product to impose liability. Because the machine was not actually used within the United States, there was no direct infringement. Moreover, lack of apparatus claims precluded contributory infringement because contributory infringement is conditioned on a direct infringement. The court then dismissed potential § 271(f) claims by merely stating: "Finally, we do not find the provisions of 35 U.S.C. § 271(f) (1988) implicated." *Id.*

220. 6 F. Supp. 2d 537 (E.D. Va. 1998). One other recent Federal Circuit case involved software in this context but no issues concerning § 271(f) liability were considered because they were not timely addressed at the district court level. *See* Southwest Software, Inc. v. Harlequin Inc., 226 F.3d 1280, 1290 (Fed. Cir. 2000). *Southwest Software* provided the Federal Circuit the opportunity to affirm *Enpat*'s process claim distinction; however, the court declined to comment, noting only that the "argument[s] concerning the application of § 271[(f)] to method claims [were] raised for the first time on appeal; for that reason, we will not consider [them]." *Id.*

Enpat, Microsoft was accused of infringing a process software patent. The court rejected application of § 271(f) on two grounds. First, it relied on *Deepsouth*'s apparatus components as a limit on scope of protectability.[221] It then cited *Standard Havens* as rejection of process protection under § 271(f).[222]

As a consequence, *Enpat* has significant implications for software patent protection. Because software has traditionally been claimed in process form, many of the earlier software patents will not enjoy protection against § 271(f) infringement. Recent trends towards protecting software more broadly with both process and apparatus claims should, however, expand protectability under § 271(f) as apparatus claims for software increase. Therefore, inventors seeking protection against potential § 271(f) infringement should claim any pure software inventions, as well as hybrid hardware and software inventions, in both process and apparatus form. Any analysis of potential § 271(f) liability should carefully consider what form the allegedly infringed invention is claimed. If it is claimed in process form only, § 271(f) should not apply.

C. Software Component Export Problems

In addition to the substantive component requirements, § 271(f) includes an export element requiring that the component or components be supplied or caused to be supplied from the United States.[223] The language used in §§ 271(f)(1) and 271(f)(2) is identical,[224] suggesting that the nature of supplying, unlike what constitutes a component for liability purposes,[225] is the same under both subsections. In many cases, the means of supply will be similar to *Deepsouth*, where a component or series of components are physically transported out of the country.[226]

221. The court emphasized the fact that *Deepsouth* involved machine components protected as part of a product, rather than process, patent. *Enpat*, 6 F. Supp. 2d at 538.

222. *Id.* (citing *Standard Havens*, 953 F.2d at 1374).

223. 35 § U.S.C. § 271(f) (2000).

224. Both subsections state that "[w]hoever without authority supplies or causes to be supplied in or from the United States" is liable for infringement. *Id.*

225. The component distinction rests on the difference between induced and contributory infringement. Section 271(f)(1) imposes liability for components in any form, as long as they make up all or most of the invention. Section 271(f)(2), however, requires the component to be specifically made for the infringing use and have no other substantial commercial purpose. *See* 35 U.S.C. § 271(f).

226. For example, in *Deepsouth* the mechanical parts were presumably merely boxed unassembled in the United States and then shipped to a Brazilian customer for final assembly. Deepsouth Packing Co. v. Laitram Corp., 406 U.S. 518, 523 n.5 (1972).

While it has yet to be addressed in the case law, the intangible nature of software raises conceptual concerns regarding what other activities constitute export for liability purposes. These concerns can be grouped into two general categories: first, what channels of export constitute a "supply" of the component; and second, what quantity of software is actually exported. This Comment proposes that the channel of export should be irrelevant as long as it is nontransformative of the software itself, resulting in a received version identical to the original. The actual quantity exported, however, is critical to the analysis. Unlike *Eolas* and *Imagexpo*, where unlimited liability was imposed for export of a single master unit, liability under § 271(f) should be limited on a one-to-one basis with the quantity exported.[227]

i. Channels of Component Supply

The traditional mode of component export, as exemplified by *Deepsouth*, involved merely shipping a tangible, physical component by a standard means of transport such as by plane, truck, or ship. This basic mode of export can be used equally well in some cases of software export. For example, software in final executable form loaded onto hardware devices, such as code burned onto hardware, such as an EPROM, flash, or a programmable logic device such as an FPGA,[228] could be physically exported in a tangible form by any standard means of transport.[229] Moreover, code stored in an intermediate state that is not directly executable, such as on a hard disk,[230] could also be readily

227. This limitation relates directly to *Eolas*, where liability was imposed for multiple instances of the software even though a single, or limited number of golden masters was shipped abroad. *See* Eolas Techs., Inc. v. Microsoft Corp., 274 F. Supp. 2d 972, 974 (N.D. Ill. 2003).

228. EPROM is an acronym for erasable programmable read only memory, which is memory that can be erased and rewritten. Flash memory is similar to EPROM but allows for rapid reprogramming of specific memory addresses. An FPGA is a field programmable gate array, another type of programmable hardware device that more closely resembles a matrix of digital logic circuits. These devices are just a sample of popular programmable hardware devices that can store code in its final executable form. *See* KEN ARNOLD, EMBEDDED CONTROLLER HARDWARE DESIGN 95–115 (2001) (explaining memory technology including EPROMs); Webopedia Computer Dictionary, FPGA, *at* http://www.webopedia.com/TERM/F/FPGA.html (last modified May 13, 2003) (explaining FPGAs).

229. This mode of export also raises questions regarding whether what is exported constitutes a single component, potentially avoiding liability under § 271(f) as exemplified in *Fieldturf*, or whether the software burned on the device and the device itself constitute separate components, thereby potentially creating liability under both § 271(f)(1) and § 271(f)(2). For purposes of this discussion it is assumed that combinations of hardware and software like this constitute a single component.

230. The hard disk example is conceptually different from programmable hardware because code is not executed directly from a hard disk but rather is transferred to a

exported in this fashion, creating potential § 271(f)(1) or § 271(f)(2) liability.[231]

A more difficult question arises, however, when the nonphysical nature of software, and the associated nontangible means of distribution, are considered. As an example, consider the fact that software in the form of executable code is just a series of numbers, typically represented in binary form or a more compact equivalent.[232] Any executable code can therefore be written as that sequence of numbers. Suppose that a piece of executable code that would constitute a component under § 271(f) was stored on an EPROM device as discussed above. If the EPROM code was then read out, it could be written on a sheet of paper as a series of numbers. If the EPROM was then exported either alone or as part of a kit or combination,[233] it would intuitively constitute supply under § 271(f) as discussed above. Suppose instead that the programmer merely called a colleague outside the United States and read the numbers off the paper over the telephone. Does this constitute supply under § 271(f)? It appears that the same ultimate result would occur, assuming the code could be reloaded into an EPROM or similar device overseas and then assembled into a patented invention.[234] Nevertheless, there are

---

hardware memory device, typically DRAM but also SRAM or even flash and then executed. Nevertheless, assuming software stored on a hard disk is a component for liability purposes, liability should be the same.

231. For example, imagine a kit consisting of a package of parts. The kit, when assembled, is a device consisting of a product housing, electronic and digital hardware, and associated software stored in hardware or on a hard disk. Suppose the kit is boxed and shipped via truck to Mexico for assembly. This would be a straightforward case of infringement under § 271(f) and a close analogy to *Deepsouth*.

232. Binary numbers are merely numbers using a base of two rather than ten as we ordinarily use. Binary numbers can be written as a series of zeros or ones. This becomes cumbersome, however, because the number of digits can quickly become large. Therefore, more compressed number bases, such as hexadecimal, are frequently used to more compactly represent binary numbers. *See* University of Calgary–Computer Science, The Hexadecimal Number Base System, *at* http://pages.cpsc.ucalgary.ca/~becker/231/Notes/Binary/ExtraNotes/hex.html (last visited Feb. 9, 2004) (explaining hexadecimal numbers and conversion from decimal and binary).

233. Export of the EPROM alone should only impose liability under § 271(f)(2) assuming it satisfies the other requirements. Export of a kit or combination including the EPROM could impose liability under either § 271(f)(1) or § 271(f)(2) if there is more than one component, or if the EPROM and associated code were specially made for the infringing product with knowledge that it would infringe.

234. The code in this example could be readily reloaded onto an EPROM or similar device by merely converting it back to an ASCII hex file or similar format and then loading it with a standard device programmer. *See generally* PREDKO, *supra* note 210, at 179–99 (describing the process of programming devices from hex (text-based) files).

several conceptual distinctions in this example. First, the software does not remain in its final form, but rather is transformed repeatedly through the supply process. Second, it lacks any of the physical characteristics of components like chemicals or machine parts as discussed previously. Therefore, the analogy to merely boxing and shipping component parts as in *Deepsouth*, where the components are subsequently assembled unmodified into a final product, fails here.[235]

Although this hypothetical means of distribution by phone is completely impractical for any significant piece of software, it demonstrates the possibilities raised by modern means of software distribution, particularly via the Internet. It is common to transfer software via the Internet either by FTP, email, or web download. A programmer might, for example, develop and test a piece of code for incorporation into a product using embedded hardware and software that is being manufactured in Mexico. Once the code is complete, he emails it to the manufacturer for download into a microcontroller on the product's circuit board during the assembly process. In sending the software, the code is likely transformed repeatedly. First, he may zip or compress the code to minimize transfer and download time.[236] At this point, the code is not only transformed from its executable form, but is also completely unrecognizable and nonfunctional as executable code; it no longer represents an executable program.[237] Next, the file might undergo numerous other transformations, such as ASCII conversion where the binary data is converted into text and packetized[238] for transmission and

235. There is also a conceptual distinction with the chemical cases, where the chemicals were combined or modified as part of the manufacturing process, thereby transforming them. They were not, however, transformed domestically or during shipment, only as part of the manufacturing process. *See* W.R. Grace & Co. v. Intercat, Inc., 60 F. Supp. 2d 316, 319–21 (D. Del. 1999)

236. Compression is commonly used to minimize file size and transmission time as well as decrease the likelihood of file error or retransmission when an in-process transfer is interrupted. This process consists of removing redundancy from the series of numbers comprising the executable code. For more information on zip files and file compression, see Brock Wood, All About .ZIP Files, *at* http://www.eurekais.com/brock/aazip.htm (last updated Nov. 23, 2001).

237. Although it no longer represents the original file or program, it is still related to the original program by a series of mathematical steps constituting the encryption and decryption algorithms. These steps, however, can lead to a relatively high degree of differentiation between the original and resulting compressed file. Using these algorithms, compression on the order of seventy to eighty-five percent is not uncommon. For a summary of maximum loss free compression ratios for common data and executable (program) files, see Maximum Compression, Lossless Data Compression Software Benchmarks/Comparisons, *at* http://www.maximumcompression.com (2005).

238. As an example, binary files are sometimes converted to ASCII text for transmission via email programs. This process is used by some email transfer programs, such as UUENCODE and UUDECODE to transfer binary data. *See* Glossary: UUencode/UUdecode, *at* http://www.learnthenet.com/english/glossary/uuencode.htm

later reassembly.[239] As a consequence, the code may be morphed repeatedly into forms bearing no resemblance to the original executable code.

Despite these distinctions from mere shipment of mechanical parts that remain physically unmodified, the original code will ultimately be reconstructed exactly as it started the process. This will occur in most, but not all cases.[240] Assuming complete reconstruction, the mere fact that the software component is transformed in the process should not be relevant when the resulting code is identical to the transmitted code.[241] Any other interpretation would necessarily disregard the intrinsic transformable properties of digital data. Ultimately, the primary concern is whether what is received is the same as what is transmitted. If the two are identical, then the transport means should not matter, even if the code is radically transformed in the export process.

ii. Quantitative Concerns

A second, and perhaps more important issue raised in the software export context concerns the quantity of software actually exported. Recall the EPROM example from the previous section. Assuming liability could be imposed under § 271(f) for exporting these EPROMs, this suggests that liability should be imposed for each article of software

(last visited Feb. 9, 2005) (explaining UUENCODE).

239. This packetization, using the TCP/IP protocol, is a core element of Internet operation. For a concise summary see G. PETER ALBERT, JR. & LAFF, WHITESEL & SARET, LTD., INTELLECTUAL PROPERTY LAW IN CYBERSPACE 14–18 (1999).

240. Some compression schemes regenerate the original content exactly and are known as lossless. Other schemes, such as JPEG, which is commonly used for image compression, and MP3, for sound compression use, lossy compression which actually transforms the original file to maximize compression. These lossy schemes are not appropriate for software in the form of executable code; however, they might be used in some cases for software components such as images. Executable code must use a lossless scheme and, therefore, the resulting uncompressed file should be an exact duplicate of the original. For an overview of lossy compression as implemented by the JPEG algorithm, see Gordon Richardson, Jpeg Compression Introduction, *at* http://www.photo.net/learn/jpeg (2003).

241. This view replaces the supply or transmission process with a black box. Rather than consider whether the same component is physically moved from one place to another, the black box approach looks at the component going in and coming out. If mechanical parts like cases, nuts, bolts, and other hardware coming out of the boxes at the manufacturer are the same as those going in the boxes at the start of shipment, then what happens en route is irrelevant. Likewise, if the code going into the supply system is identical coming out, then the fact that the process transforms it in some way, even a completely radical way, is irrelevant under this analysis.

exported, with a one-to-one relationship between the export and resulting products. For example, suppose the EPROM is incorporated into a lawn mower or other motorized product (the EPROM might, for example, control engine operation or emissions), that would infringe a U.S. patent if manufactured domestically.[242] Each EPROM device exported should result in liability for one instance of infringement, with the associated liability for lost profits or reasonable royalty as statutorily allowed.[243] If 100,000 EPROMS and the associated code were exported, liability would be imposed for 100,000 infringing articles.[244] There is a problem with this analysis in the case of software, however, that has yet to be addressed in the case law under § 271(f).

While liability for export of software based on the number of units exported seems straightforward, under the recent Microsoft cases, liability for large numbers of infringing devices may be imposed for export of a single copy of the software.[245] For example, suppose Microsoft exports 100,000 copies of Windows that are then subsequently installed on computers in Mexico and sold internationally. As in the EPROM example above, this should result in liability for 100,000 units. If instead Microsoft exports a single "golden master" copy of Windows, as in *Eolas* and *Imagexpo*, and then installs the software repeatedly onto 100,000 units, should this action result in similar liability? Intuition and § 271(f) statutory history suggests it should not. As previously noted, § 271(f) was enacted as a limitation on manufacturing activities and has been narrowly construed within that context. When a single master copy of the software is exported, it represents merely that, a single unit. If that unit is loaded repeatedly into multiple devices abroad, those multiple units are created overseas at the time of manufacture, not in the

242. For this example, assume the product would infringe one or more U.S. patents if made or used domestically. This requires that the product contain all the claimed elements of the patents, as discussed previously, thereby satisfying the requirements of § 271(f)(1) or § 271(f)(2).

243. 35 U.S.C. §§ 283–84 (2000). Microsoft's liability in *Eolas* was assessed in this way. The jury multiplied a value of $1.47 per infringing copy times 354 million copies sold to arrive at total damages of $521 million. Cherry, *supra* note 9, at 47.

244. Under the statutory liability requirements, the patent holder could recover the actual value of lost profits from those 100,000 unit sales depending on certain factors. *See, e.g.*, Panduit Corp. v. Stahlin Bros. Fibre Works, Inc., 575 F.2d 1152, 1154–58 (6th Cir. 1978). Alternately, if there is an insufficient showing of the patent holder's ability to satisfy the market demand or market uncertainty, she might only be entitled to a reasonable royalty, on a per unit basis, based on the actual number of units sold. *Id.* In either case, liability would be assessed based on the quantity of infringing products produced.

245. For example, in *Eolas*, the jury imposed liability on a per unit basis for all sales of Microsoft Windows overseas, despite the fact that multiple instances of Windows were installed from a single "golden master." Eolas Techs., Inc. v. Microsoft Corp., 274 F. Supp. 2d 972, 973–74 (N.D. Ill. 2003).

United States. As a result, there should be no liability because those copies are not actually supplied from the United States, but are created abroad at the time the source code is loaded into the device. Since each instance is actually created outside the United States at the time of manufacture, no liability should be imposed.[246] The alternate view of liability, as embraced by *Eolas*, can create a paradoxical anomaly with *Waymark*, imposing indeterminate liability when the components are not actually assembled abroad.[247] Because of these concerns, export liability under § 271(f) should be limited to the actual quantity of software units exported, not the number of instances created extraterritorially.

D. Policy Considerations

This Comment suggests that § 271(f) be construed narrowly to limit liability for software component export. Under this view, software should enjoy some degree of protection under § 271(f) because it is patentable subject matter and can functionally replicate analogous mechanical component functions. Moreover, software is being increasingly incorporated into all variety of products, not just traditional mainframe or desktop PC applications. As the embedded product market continues to expand, software will become more and more pervasive, and its role in replacing physical parts that clearly receive protection under § 271(f)

246. While components manufactured domestically, then exported, can impose § 271(f) liability, the Federal Circuit has rejected extension to components made outside the United States. *See* Rotec Indus., Inc. v. Mitsubishi Corp., 215 F.3d 1246, 1257–58 (Fed. Cir. 2000) (rejecting liability for components manufactured outside the United States); Pellegrini v. Analog Devices, Inc., No. 02-11562-RWZ, 2003 U.S. Dist. LEXIS 7598, at *1–2 (D. Mass. May 7, 2003) (rejecting liability for integrated circuit chips built entirely outside the United States), *aff'd*, 375 F.3d 1113 (Fed. Cir. 2004).

247. *Waymark* imposes liability for mere shipment of the component, even if no assembly is completed. *See* Waymark Corp. v. Porta Sys. Corp., 245 F.3d 1364, 1368 (Fed. Cir. 2001). If, however, only a single master copy of software is exported, as in *Eolas* and *Imagexpo*, how can liability be assessed if the "golden master" is never actually used to manufacture anything? *Waymark* holds that liability is imposed as long as extraterritorial assembly was intended, but how should it be quantitatively assessed? Should it be based on the quantity the exporter *thought* would be made, or on what the manufacturer actually intended to make, or on some other measure? This argument can apply to other permutations as well, such as when the golden master is used to create more copies than initially authorized. Does this limit liability to the quantity intended to be created by the exporter or is the quantity limited only by how many copies are actually made? What happens if the exporter loses control of the exported software—does liability accrue for any additional copies created as well? Separating the quantity of units exported from extraterritorial product creation may create enormous uncertainty in how this type of liability is assessed.

will only expand. Nevertheless, as previously discussed, protection should be construed narrowly to a manufacturing context, as distinguished from a design context, to avoid overextending patent protection from its traditional limits.

In addition to the substantive reasons previously discussed, several policy concerns support limiting liability under § 271(f). First, there is a general policy concern, reflected throughout the history of patent law, against extending protection beyond the boundaries of the United States.[248] This concern should only increase as international patent systems continue to harmonize through global agreements.[249] Access to patent protection throughout the world is now easier and more readily available to all inventors, suggesting that those who wish to extend their protection overseas file for patent protection internationally if their inventions truly merit global protection.[250]

A second reason relates to the dramatic increase in global manufacturing, particulary in China.[251] Since § 271(f) was enacted in 1984, the nature of manufacturing has changed dramatically, with much of what was once made domestically now being manufactured overseas.[252] Given the cost advantages of manufacturing in countries like China, this trend is likely to continue. Although the stated purpose of § 271(f) was to discourage manufacturing outside the United States, the fact that international manufacturing has expanded dramatically—and the United States has ratified trade agreements such as NAFTA and GATT that further facilitate this expansion—suggests less need for broad protection under § 271(f).[253]

---

248. *See supra* notes 32–57 and accompanying text.

249. These agreements include the Paris Convention, the Patent Cooperation Treaty (PCT), as well as others. *See generally* MARTIN J. ADELMAN ET AL., CASES AND MATERIALS ON PATENT LAW 699–743 (2d ed. 2003) (discussing the various international protection schemes and how to prosecute patents internationally).

250. *Id.*; *see also* STOBBS, *supra* note 29, at 492–519.

251. Globalization has resulted in movement of manufacturing to the lowest cost location. These trends have "resulted in growing international trade . . . and more distributed manufacturing capabilities." SCHECHTER & THOMAS, *supra* note 24, at 517.

252. The United States Bureau of Labor Statistics reports a decline in manufacturing jobs from about 17 million jobs prior to 2001 to around 14.5 million in 2003. *See* U.S. Dep't of Labor: Bureau of Labor Statistics, NAICS 31–33: Manufacturing, *at* http://www.bls.gov/iag/manufacturing.htm (last modified Dec. 29, 2004). At the same time, China's output of manufactured goods, as well as its imports to the United States, have soared. *See* N. Gregory Mankiw, China's Trade and U.S. Manufacturing Jobs, http://www.whitehouse.gov/cea/mankiw_testimony_house_ways_and_means_oct_30.html (Oct. 30, 2003).

253. Recall that the stated purpose of Congress in enacting § 271(f) was to discourage foreign manufacturing. *See supra* notes 118–23 and accompanying text. Because Congress and the President have freely entered into global trade agreements such as NAFTA that encourage overseas manufacturing, the underlying policy basis behind § 271(f), discouraging extraterritorial manufacturing, may be questionable today.

A third reason relates to the trend of outsourcing information technology and software jobs. Over the last several years, companies have increasingly turned to exporting these high skill, and high paying, jobs overseas.[254] Section 271(f)'s role in this trend is indeterminate; however, broad application of § 271(f) to activities like software design and development—that are not distinctly separated from manufacturing, as in other industrial areas—might further encourage employers to export software development jobs. Suppose a company plans to develop a product to target a particular country knowing there is no U.S. patent protection there. Does it make sense to develop the software components of that product in the United States and risk possible infringement merely for emailing the code to the overseas manufacturer when the exact same effort can be done in lower cost countries such as India or China with no similar implications?

For these policy reasons, liability under § 271(f) should be construed narrowly to provide economic incentives for American companies and their domestic employees to compete in an increasingly global marketplace.

V. CONCLUSION

As technical innovations reshape our world, computer hardware and its associated software will undoubtedly play an expanding role. Personal computers and their associated desktop and Internet applications will likely dominate our perceptions of computer and software technology. Behind the scenes, however, embedded software applications will be used to replace traditional mechanical and electronic components in a growing variety of devices and systems. In the future, these embedded processors will be found in almost every type of product. Moreover, use of Internet and distributed applications, in both desktop and portable devices, will expand dramatically as well, blurring the boundaries of where and how software is used. The consequence of these trends is that computer software will be present in more and more products, in many cases providing a critical role, but often hidden from view. Inventors and their employers will undoubtedly try to protect their efforts in developing these innovations through the patent system, particularly

254. *See, e.g.*, Stephen Baker et al., *Software: Programming Jobs are Heading Overseas by the Thousands. Is There a Way for the U.S. to Stay on Top?*, BUS. WK. ONLINE, *at* http://www.businessweek.com/magazine/content/04_09/b3872001_mz001.htm (Mar. 1, 2004).

now that earlier limitations on software patentability have been removed. At the same time, businesses will increasingly turn to offshore product manufacturing to trim costs and capitalize on the increasingly global market for products. This will make it common for software developed in the United States to be exported, in a variety of ways, for integration into products manufactured overseas, opening the door for potential extraterritorial infringement liability under § 271(f).

The law must be flexible in adapting to these changes by allowing protection for nontraditional types of product components such as software. It must, however, do this by recognizing the limited manufacturing-related purposes behind 35 U.S.C. § 271(f) and narrowly tailor potentially infringing components to these situations. Additional policy concerns about the changing role of product design and manufacturing, in light of rapid offshore manufacturing and market globalization, suggest that domestic innovation might be enhanced by limiting the scope of what is a protectable component; perhaps § 271(f) should even be revisited. Taken in concert, these concerns require that software be treated as a component for purposes of § 271(f) liability, but that the scope of protectability be narrowly limited as proposed in this Comment.

As the patent system has demonstrated over the last fifty years, it is capable of recognizing conceptual distinctions associated with new technologies, and ultimately assimilating them without dramatic statutory changes. Software components are no exception. Software can be reconciled with the protections afforded by 35 U.S.C. § 271(f) without creating the excessively broad infringement liability suggested by *Eolas* and *Imagexpo*. This can be done by imposing liability for software component export, but only for those components closely tied to product manufacturing. In addition, liability should only be imposed based on the actual quantity of software units exported, not for mere export of master units that are replicated abroad.

STEVEN C. TIETSWORTH