Santa Clara University
School of Law

Santa Clara High Technology Law Journal

Volume 14 | Issue 2 Article 1

January 1998

Software Patent Infringement on the Internet and on Modern Computer Systems - Who Is Liable for Damages?

Keith E. Witek

ARTICLES

SOFTWARE PATENT INFRINGEMENT ON THE INTERNET AND ON MODERN COMPUTER SYSTEMS — WHO IS LIABLE FOR DAMAGES?*

Keith E. Witek†

TABLE OF CONTENTS

† Practicing Motorola Patent Attorney #37,475, Motorola Inc., Semiconductor Product Sector, Austin, TX. The opinions expressed herein are my views and not necessarily the views of Motorola Inc. Special thanks to Prof. Mark Lemley and Prof. Neil Natenel, both of the University of Texas Law School at Austin, and Robert L. King, patent attorney, Motorola Semiconductor Products Sector, for their many critiques and comments concerning the content of this paper. In addition, special thanks to my wife, Melissa Witek, for putting up with the long hours writing this paper, and to my daughters Jayme Witek and newborn Jodi Witek who keep life in proper perspective.

I. INTRODUCTION

The information superhighway, or Internet, has caused a considerable stir in the law and legal community over the last decade and led to numerous questions. For example, practitioners have asked what type of behavior constitutes criminal activity on the information superhighway, and how should this criminal activity be prosecuted?[1] What about the right to free speech, the right to privacy, and other constitutional rights on the Internet and the effects on anonymous remailers?[2] What should we do about libel and injurious speech claims over the Internet?[3] How should we deal with copyright infringement?[4] Additionally, which court has jurisdiction over parties committing wrongful conduct over the Internet?[5]

This list of legal concerns regarding the Internet is constantly growing in scope and length. As the Internet grows, both in size[6] and speed,[7] it becomes more likely that massive quantities of soft-

[1] Catherine Therese Clarke, *From Criminet to Cyber-Perp: Toward an Inclusive Approach to Policing the Evolving Criminal Mens Rea on the Internet*, 75 OR. L. REV. 191 (1996); Sean Adam Shiff, *The Good, the Bad and the Ugly: Criminal Liability for Obscene and Indecent Speech on the Internet*, 22 WM. MITCHELL L. REV. 731 (1996) (discussing criminal liability on the Internet).

[2] Lee Tien, *Who's Afraid of Anonymous Speech? McIntyre and the Internet*, 75 OR. L. REV. 117 (1996); Byron F. Marchant, *On-Line on the Internet: First Amendment and Intellectual Property Uncertainties in the On-Line World*, 39 HOW. L.J. 477 (1996); Mitchel L. Winick et al., *Attorney Advertising on the Internet: From Arizona to Texas — Regulating Speech on the Cyber-Frontier*, 27 TEX. TECH L. REV. 1487 (1996); Jeffrey E. Faucette, *The Freedom of Speech at Risk in Cyberspace: Obscenity Doctrine And A Frightened University's Censorship of Sex on the Internet*, 44 DUKE L.J. 1155 (1995) (discussing constitutional issues raised by the Internet).

[3] *See* Stratton Oakmont, Inc. v. Prodigy Servs. Co., 23 MEDIA L. REP. (BNA) 1794 (N.Y. Sup. Ct. 1995); Cubby v. Compuserve, 776 F. Supp. 135 (S.D.N.Y. 1991); United States v. Thomas, 74 F.3d 701 (6th Cir. 1996) (analyzing libel law and defamation on the Internet).

[4] *See* Religious Tech. Ctr. v. Netcom On-Line Communication Servs., Inc., 907 F. Supp. 1361 (N.D. Cal 1995); Playboy Enters., Inc. v. Frena, 839 F. Supp. 1552 (M.D. Fla. 1993); Sega Enters. Ltd. v. MAPHIA, 857 F. Supp. 679 (N.D. Cal. 1994) (discussing copyright infringement on the Internet).

[5] For jurisdictional law on the Internet, *see* Compuserve, Inc. v. Patterson, 89 F.3d 1257, (6th Cir. 1996); California Software, Inc. v. Reliability Research, Inc., 631 F. Supp. 1356 (C.D. Cal. 1986); Playboy Enters., Inc. v. Chuckleberry Publ'g, Inc., 687 F.2d 563 (2d Cir. 1982). *See also* Gary L. Gassman, Moot Court Competition Bench Memorandum: *Internet Defamation: Jurisdiction in Cyberspace and the Public Figure Doctrine*, 14 JOHN MARSHALL J. COMPUTER & INFO. L. 563 (1996); Richard S. Zembek, Comment, *Jurisdiction and the Internet: Fundamental Fairness in the Networked World of Cyberspace*, 6 ALB. L.J. SCI. & TECH. 339 (1996).

[6] By late 1997, over 100 million computers were projected to be coupled together, worldwide, across the Internet. *See* DOUGLAS E. COMER, INTERNETWORKING WITH TCP/IP 8 (2d ed. 1991) ("Within seven years of its inception, the Internet had grown to span hundreds of individual networks located throughout the United States and Europe.... Both the size and use of the Internet continued to grow much faster than anticipated.").

[7] New telecommunication technologies, such as Integrated Services Digital Network (ISDN), Asynchronous Transfer Mode (ATM), and Asymmetric Digital Subscriber Line (ADSL), are capable of enabling megabit transmission rates, the likes of which will enable online real-time video as well as speedy software communication never before seen in electronic

ware and data structures[8] will be purchased, transmitted, and hacked[9] over the Internet.[10] Further, many users consider software to have fluid qualities, moving across state lines and national boundaries with ease on the Internet.[11] This fluid nature of software will only increase with new innovations, such as the network computer (NC).[12] The Internet, unlike other manufacturing environments, can enable users to create millions of copies of software or data structures and distribute these copies to millions of computer users within very short periods of time. In essence, modern computers and the Internet may be viewed as high speed micro-factories that can create and copy software at an astounding rate. Even unsophisticated computer users on the Internet can inflict extensive financial damage on soft-

long-distance communication. *See* BUD GATES & DONALD GREGORY, VOICE AND DATA COMMUNICATIONS HANDBOOK 477-554 (1996).

[8] A data structure is defined as "a physical or logical relationship among data elements, designed to support specified data manipulation functions." IEEE STANDARD COMPUTER DICTIONARY (1991).

[9] "Hacking" a computer program is an electronic form of stealing or manner of illegally copying the software or data structures from a rightful cyberspace possessor.

[10] For example, technology referred to as "applets" is currently causing a technical stir on the Web. An applet is a stripped-down, or locked, version of a software application that can be downloaded from the Internet or an intranet. Applets enable the user to demo the full-featured application. Certain software applications may be required to view or run the user's data file. For example, if the data is created using Program A, the applet is code compatible with Program A and allows the receiver to view the data even if the receiver does not have the entire Program A at the receiving end. Any interception or theft of this data, however, will also steal the applet which may result in patent infringement. *See* Jason Snell, *Web Publishing: Version Two*, MACUSER, Jan. 1, 1997 for an example discussion of applets.

In addition, many computer games are now played over the Internet. One entity that enables Internet game play is known as Total Entertainment Network (TEN). *See PGL: PGL Finals to be Webcast Live*, M2 PRESSWIRE, Feb. 2, 1998. Moreover, it is always possible for a user to send a friend a game or program over the Internet whereby the software owner suffers economic harm. Even worse, hackers can post these games at websites for thousands of people to illegally download within a few hours.

[11] Some envision that software will soon be commonly purchased over the Internet without ever leaving the comforts of home. For some examples, *see Internex Readies EC-Based Software Purchase Service*, ELECTRONIC COMMERCE NEWS, Nov. 18, 1996, *available in* 1996 WL 13830257; S.L. Millin, *With Help from Release, Egghead Plans to Test Try-Before-You-Buy Concept*, SOFTWARE INDUSTRY REPORT, Nov. 18, 1996, *available in* 1996 WL 8349536. However, increased availability of software on the Internet as a result of on-line purchasing capabilities may lead to increased software hacking on the Internet.

[12] A network computer (NC) is an inexpensive computer that executes software from a remote location and does not store or execute software locally. Such configuration saves the user money since the NC need not be equipped with the high-priced circuitry required to store and handle software, locally (e.g., DRAM memory). However, an NC must access its software along telecommunication interfaces, e.g., the Internet, more than other systems.

ware manufacturers or owners in a very short period of time and without any possibility of a preemptive warning.

In this technological environment, the United States Patent and Trademark Office (USPTO) and other countries' intellectual property organizations[13] are issuing software patents which contain newly-emerged software claim styles, such as software article of manufacture patent claims,[14] *Alappat* means-plus-function software patent claims,[15] and software data structure patent claims.[16] These software claims emerged in U.S. patent law during the 1994-1995 time period, and such claims are just now being granted in many U.S. patent applications.[17] However, what ultimately will happen when these claims are infringed is still an unknown. Due to the recent acceptance and use of these software patent claims, the author did not locate case law or articles which specifically discuss the application of these new software patent claims to either software or data transfers on the Internet. However, similar lawsuits involving the Internet, software, and computer technology have already begun to appear in the copyright context.[18]

[13] *See* Naomi AZA ssia, *Patent Protection for Software in Israel*, 29 COMPUTER 90, 90-91 (1996); Gert D. Kolle, *Patentability of Software-Related Inventions in Europe: Law and Practice Under the European Patent Convention*, 27 IIC 660 (1995) (published by VCH Verlagsgesellschaft mbH, P.O. Box 10 11 61, D- 69451 Weinheim, Germany); Yoshikazu Tani, *Protection of Computer Software in Japan*, PATENTS & LICENSING, Feb. 1996, at 7.

[14] *See infra* Part I.C.1. *See also In re* Beauregard, 53 F.3d 1583, 1584 (Fed. Cir. 1995) (stating computer programs embodied in a tangible medium, such as floppy diskettes, CDs, and magnetic tape, are patentable subject matter under 35 U.S.C. § 101 as articles of manufacture and must be examined under 35 U.S.C. § § 102 and 103).

[15] *See infra* Part I.C.2. *See also In re* Alappat, 33 F.3d 1526, 1545 (Fed. Cir. 1994) (en banc) (holding that novel software present on a known computer architecture "creates a new machine").

[16] *See infra* Part I.C.3. *See also In re* Lowry, 32 F.3d 1579, 1583-84 (Fed. Cir. 1994) (finding that data structure is patentable since the data structure was viewed as being specific electrical or magnetic structural elements in a computer memory).

[17] With the new GATT rules, patents are issuing in roughly 2-3 years and are enforceable for twenty years from the filing date of the patent application. *See* Mark A. Lemley, *An Empirical Study of the Twenty Year Patent Term*, 22 AIPLA Q.J. 369 (1994).

[18] *See Netcom*, 907 F. Supp. at 1368 (holding Internet access provider was not directly liable for unauthorized copies of copyrighted work that were made and stored on its computer while transmitting computer bulletin board service's (BBS) Usenet postings to and from the Internet); Playboy Enter. Inc. v. Frena, 839 F. Supp. 1552, 1556 (holding subscription computer bulletin board service directly infringed magazine's copyrights by distributing copyrighted photographs, even if bulletin board operator did not know that the photographs had been uploaded by subscribers onto bulletin board); Sega Enter. Ltd. v. MAPHIA, 857 F. Supp. 679, 686 (holding both the computer bulletin board company and the individual controlling

While an Internet user is capable of inflicting extensive financial damage in a short period of time, the nature of the Internet may prevent a software owner from obtaining an adequate remedy. In the future, software owners, who suffer financial damage due to hacking on the Internet, will likely file software patent infringement lawsuits to obtain a remedy due to any infringement of these new software claims. In many cases, the hacker or thief performing the infringing activity and causing the financial damage may be judgment proof, e.g., if the hacker is a minor, or the hacker's identity is hidden from detection using technology such as anonymous remailers or encryption. In these cases, the software owner cannot be made whole by simply suing the hacker who intentionally performed the infringing act.

The U.S. courts have yet to decide from a policy perspective who should bear the risk and liability for mass patent infringement over the Internet. The current patent law doctrine allows the courts to follow one of two primary theories. The courts can either: (A) hold the Internet-enabling entities[19] liable for all the damage when the software owner cannot be made whole due to judgment-proof defendants; or (B) find that patent law shelters the Internet-enabling entities from massive liability, thereby avoiding any chilling effect on the Internet while placing the risk of financial loss from Internet hacking on the software manufactures and database creators. To determine which of the two choices are likely to be adopted in U.S. patent law, one must examine recently published copyright policy discussions and federal copyright decisions due to the lack of patent infringement case law involving the Internet environment.

that bulletin board were liable for direct and contributory infringement after unknown Internet users uploaded unauthorized copies of video games to the bulletin board. Subsequently, other Internet users downloaded the unauthorized video games, thereby creating additional infringing copies. The defendants allegedly knew of and facilitated the infringing activity.).

[19] "Internet-enabling entities" include Internet hardware manufacturers (such as computer manufacturers and software developers), Internet service providers (companies that provide computer users access the Internet), Internet resource owners, and telecommunications service providers. Copyright case law suggests it is more likely that Internet service providers and similar entities will be subject to software patent infringement claims in the future than Internet software and hardware manufacturers. *See In re* Lowry, 32 F.3d 1579 (Fed. Cir. 1994). However, all Internet-enabling entities which create or provide Internet technologies to the market are considered, herein, to render a complete picture of software patent liability on the Internet.

Concerning policy choice (A), a court may conclude that patent infringement is a strict liability tort[20] and hold the Internet-enabling entities liable under theories similar to those applied for copyright infringement in *Sega Enterprises Ltd. v. MAPHIA*[21] and *Playboy Enterprises, Inc. v. Frena*.[22] Alternatively, the courts may decide to hold the Internet-enabling entities liable under a theory of inducement or vicarious liability in a manner similar to that found in copyright law.[23] Under policy choice (B), a court may choose to provide Internet-enabling entities with the protection of contributory infringement and require volition or a volitional act[24] on the part of the

[20] *See* Kewanee Oil Co. v. Bicron Corp., 416 U.S. 470, 478 (1974) (noting that patent law forbids independent creation of the patented matter and needs no intent or knowledge of infringement); Thurber Corp. v. Fairchild Motor Corp., 269 F.2d 841, 845 (5th Cir. 1959) (holding that infringement does not depend on the good faith or innocent mind-set of the infringer); Metal Film Co. v. Melton Corp., 316 F. Supp. 96, 111 n.15 (noting that neither lack of knowledge of the patent nor lack of intent to infringe is a defense for the issue of patent infringement).

[21] 857 F. Supp. at 686 (holding that electronic bulletin board operator was infringing under both direct and contributory infringement due to a bulletin board user's uploading of illegal copies of software to the bulletin board).

[22] 839 F. Supp. at 1556 (finding that electronic bulletin board operator was liable for the transfer of copied, digitized photographs through his bulletin board despite the operator's lack of knowledge of the transfer).

[23] *See* Columbia Pictures Indus., Inc. v. Aveco, Inc., 800 F.2d 59 (3d Cir. 1986) (providing the necessary equipment to infringe may constitute contributory infringement or inducement); Demetriades v. Kaufman, 690 F. Supp. 289 (S.D.N.Y. 1988) (noting that if one encourages, requests, or promotes the infringement, one can be held liable under an inducement theory of copyright law); Fonovisa, Inc. v. Cherry Auction, Inc., 76 F.3d 259 (9th Cir. 1996) (holding that vicarious liability can be found where: (1) the defendant has a right and ability to control the acts of the infringer, and (2) the defendant receives a direct financial benefit from the infringement; here, the defendant who coordinated and held a flea market and was held vicariously liable where a flea market is similar to some Internet service providers and on-line purchasing sites)).

[24] For tort liability, one must have had either the prerequisite intent and/or committed the wrongful act. In certain circumstances, one can justifiably be found liable for committing a harmful act although the actor had no intent to do wrong (i.e., strict liability). In extreme circumstances, it may also be reasonable to hold a defendant liable for damages for simply having had the requisite intent *absent* an accompanying bad act. However, it is intuitively improper to hold someone liable when that person did not intend to committed a directly harmful act but merely an act remotely related to the infringing activity. If it were possible for one to be held liable for massive damages without directly committing an act and without intent or knowledge, it may be more efficient to randomly, yet equally, assign liability to citizens and corporations and do away with lengthy trials. Even strict liability, which has no element of intent, requires that the defendant performed some act or had a direct causation or link to the infringing activity.

Patent infringement, a strict liability tort, does not require that the defendant have any intent; however, the patent law does require that the defendant commit some "act" to contribute, Internet-enabling entities before finding patent infringement liability, as was done via *Religious Technology Center v. Netcom On-Line Communication Services* and *Sony Corp. v. Universal City Studios, Inc.* in the copyright context.[25]

Both policy perspectives (A) and (B), above, have advantages and disadvantages. It is therefore unclear which theory, or some combination thereof, the Court of Appeals for the Federal Circuit (CAFC) will adopt for patent infringement on the Internet. The author believes the volitional requirement and contributory infringement approach of choice (B) is more reasonable for U.S. patent law' on the Internet. This approach spreads the risk of loss among the many software vendors and avoids chilling effects on the manufacture, operation, and use of Internet technology. Additional benefits will be discussed herein.

Part I of this paper will introduce a "typical" hacker scenario to illustrate the problem of software and database hacking on the Internet, as well as provide a background for copyright infringement theories as they relate to patent infringement on the Internet. It also discusses the newly-utilized software patent claims, which could cause software patent liability issues resulting from use of the Internet. Part II examines the types of software and database copies typically encountered on the Internet and how these copies may result in

aid, induce, or further the infringement before being held liable. *See* cases cited *supra* note 20. If the defendant committed no act that he could control or prevent, what infringement avoidance incentive does assessing massive damages and injunctions serve? *See* Eastman Oil Well Survey Co. v. Sperry-Sun Well Surveying Co., 131 F.2d 884, 887 (5th Cir. 1944) (basing liability on "what the defendant is doing" infringes or not); Orthokinetics, Inc. v. Safty Travel Chairs, Inc., 806 F.2d 1565, 1578 (Fed. Cir. 1986) (finding defendants personally liable for *acts of direct infringement*) (emphasis added). "Act" may also include omission or failure to act after one is given knowledge of infringement. Suppose one told an Internet entity that infringing software was on his service and he chose to do nothing for two years? *See also* Black & Decker (U.S.), Inc. v. Home Prod. Mktg., Inc., 929 F. Supp. 1114, 1121 (N.D. Ill. 1996) (noting that either acts or omissions accompanied by knowledge can result in direct infringement), *reconsideration denied,* 935 F. Supp. 1010 (N.D.Ill. 1996).

[25] *See Netcom*, 907 F. Supp. at 1368 (holding that "incidental copies" are not direct or contributory infringement); *see also,* Marobie-FL, Inc. v. the National Association of Fire Equipment Distributors, *available in* 1997 U.S. Dist. LEXIS 18764 (N.D. Ill. Nov. 13, 1997) (generally supports the rationale of *Netcom*); Sony Corp. v. Universal City Studios, Inc., 464 U.S. 417, 435 (1984) (applying fair use and contributory infringement theories so that the act of supplying the technology to enable illegal copying would not be hampered by massive liability or injunction, especially where the technology provided has many other non-infringing useful public purposes).

patent infringement liability. Part III identifies technology, which could enable users to infringe the software patent claims, discussed in Part I.C., by creating electronic and magnetic software copies, as discussed in Part II. Parts IV and V consider the U.S. patent law, which could enable a court to adopt either policy perspective (A) or (B); strict liability tort or liability only when the entity commits a volitional act, respectively. Lastly, Part VI offers reasons why the second theory, requiring a volitional act of infringement, is more reasonable for U.S. patent law concerning the Internet.

A. A Common Hacker Scenario

A computer hacker illegally gains access to a computer program available on an Internet page or network. The Internet page is physically resident within the memory of a computer[26] located somewhere in New York City, New York. The computer hacker downloads the computer program to his home computer in San Francisco, California by clicking on an HTML menu item.[27] The computer in New York uses a specific Internet program[28] and a specific operating system[29] to begin the electronic transfer of the hacked software. The New York site stores the requested hacked copy from the computer's hard disk to dynamic random access memory (DRAM)[30] within the computer using the computer's microprocessor. The microprocessor within the computer in New York copies the program, bit-for-bit, from internal computer storage areas to an external modem across conductive cabling[31] or by using network ac-

[26] Computers are publicly sold or manufactured by the following entities or under the following brand names: Apple Computer ("Apple"), Compaq, Dell, Gateway 2000, Hewlett Packard, IBM, Packard Bell, Sun Microsystems, Toshiba, etc.

[27] IAN S. GRAHAM, HTML SOURCEBOOK (2d ed. 1996) (noting HTML code can be created and maintained by using software provided by many different sources).

[28] Servers are provided by the following manufacturers or brand names: Apache, Apple, CERN, Microsoft, NCSA, Netscape, Novell, etc.

[29] Examples of operating systems include: Apple's System 7.0, IBM OS/2, Microsoft DOS, Pink, UNIX, VMS, Windows NT, Windows95, etc.

[30] DRAM and like memory are manufactured and/or assembled into printed circuit boards or SIMMs by such entities as Advanced Micro Devices ("AMD"), Cypress Semiconductor, Fujitsu, Goldstar, Hitachi, Hyundai, IBM, Intel, Matsushita, Micron, Mitsubishi, Motorola, NEC, Samsung, Seimens, Texas Instruments, Toshiba, etc.

[31] Providers of computer and/or telecommunication cabling include Asante, Cabletron, Cisco, and Farallon.

cess hardware.[32] Before reaching the modem, intermediate copies may be made using various glue hardware/logic[33] within the computer located in New York. The modem creates copies using one or more microcontrollers or digital signal processors (DSPs)[34] and sends the hacked computer copy along the telephone lines where hundreds or thousands of computers and routers will produce even more copies.[35] Before the modem even connects to the telephone lines, potentially several Fortune 500 companies have contributed to the creation of one or more infringing copies of the hacked software.[36]

The hacker will receive a copy of the hacked program at his California computer, using his modem or communications card, via the communications network. The hacked software will have traveled a path of interconnecting computer networks on the Internet, and many illicit copies will have been created along the way.[37] At the hacker's computer in California, the software program is downloaded by more microcontrollers or DSPs to a hard disk where another copy of the program is created. The hacker takes a floppy disk[38] and copies the hacked program onto the floppy disk. The floppy disk is used to transfer the hacked software to yet another computer. The floppy drive on this other computer copies the hacked program to more random access memory (RAM). The hacker executes the hacked software to see the hacked graphical user

[32] Network access hardware is available from the following manufacturers or under the following brand names: 3com, AMD, Cisco, Motorola, Megahertz, Xircom, etc.

[33] Integrated circuit ("IC") components are made by many different manufactures and will not necessarily interface directly with each other due to different timing constraints, protocols, output and input terminal configurations, etc. When compatibility is an issue, "glue logic" is placed between the ICs to render one compatible with the other. General computer integrated circuit components are made by manufacturers such as AMD, Hewlett-Packard, Hitachi, IBM, Intel, Motorola, NEC, Seimens, Texas Instruments, and Toshiba.

[34] Microcontroller and/or DSP manufacturers include Hitachi, Lucent, Motorola, NEC, and Texas Instruments.

[35] The telecommunications infrastructure is massive and involves thousands of corporations, governmental entities, etc. Such involved corporate entities include MCI, Sprint, AT&T, local service providers (e.g., Southwestern Bell), America On-Line, Prodigy, Northern Telecom, IBM, Iridium, Motorola, PrimeCo, British Telecom, and Ericsson.

[36] *See* discussion *infra* Part III.B-D and accompanying Figures.

[37] *See* discussion *infra* Part III.A, III.E and accompanying Figures.

[38] Floppy disks and like disks are made by 3M, iomega, Opus, Sony, TDK, BASF, Scotch, Verbatim, etc.

interface (GUI)[39] on a computer monitor, resulting in the code being loaded into the video RAM (VRAM) of the computer. The hacker may even convert electronic copies in the RAM to a printer to obtain a tangible copy of the hacked code.

The hacker then copies the hacked program onto an Internet service site[40] where thousands of users locate and download the hacked program, each user then creates additional hacked copies. The users download the hacked software at no cost, causing substantial economic loss to the original program owner. In parallel with this hacker, and in the same week, thousands of other computer hackers perform similar computer hacking operations on other computer programs or data structures, using similar hardware and services. By the end of the week, millions of hacked copies of software have been temporarily or permanently created around the globe so that hundreds or thousands of corporations and individuals are potentially exposed to some degree of intellectual property liability.[41] If the original owner holds software patents, which exclude others from making, using, or selling the hacked software, a question arises as to who should be held liable for such infringing activities over the Internet.

B. *Use of Copyright to Remedy the Scenario's Hacking*

The original program owner has registered copyrights covering the computer instructions and computer code structure contained within the hacked program.[42] Relying on copyright protection, the original owner files a claim against the hacker only to discover that the hacker is sixteen years old and judgment-proof, or alternatively, unidentifiable due to anonymous remailing[43] or extensive encryp-

[39] Graphical user interfaces (GUI) use a mouse, Windows, and a point-and-click environment to improve the efficiency of human interface to the computer and software.

[40] A few examples are America On-Line, Compuserve, Flash Net, and Prodigy.

[41] *See* discussion *infra* Parts V, VI.

[42] *See* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983); Whelan Assocs. v. Jaslow Dental Lab., Inc., 797 F.2d 1222 (3d Cir. 1986); Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992); Lotus Dev. Corp. v. Borland Int'l, Inc., 64 USLW 4059, 96 Cal. Daily Op. Serv. 315, 64 USLW 3465, 116 S. Ct. 561, 116 S. Ct. 39 (1996 and 1995); Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435 (9th Cir. 1994); Sega Enters. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992), for the general law concerning copyright protection of software.

[43] *See* ANDRE BACARD, THE COMPUTER PRIVACY HANDBOOK 65-68 (1995).

tion.[44] Attempting to make himself whole, the original software owner files a claim against the several corporations involved in the computer program's electronic trek from New York to California.[45] The original owner argues that the unauthorized copies made by one or more of these corporations and/or individuals constitute copyright infringements. Direct infringement occurred since the unauthorized copies were "perceived, reproduced, or otherwise communicated"[46] directly with the aid of a device for more than a transitory duration.

Many of the copies made on the electronic trip from New York to California existed for only a few microseconds while other copies may be resident in some form of computer storage for years, raising a wide range of copyright fixation issues. The original software owner argues that many of these copies are "fixed"[47] under the Copyright Act. To make sense of the fixation requirement in computer contexts, a distinction is typically made between "memory"[48] and "storage"[49] within computer systems. For a decade or so, courts

[44] *See generally* WILLIAM STALLINGS, PRACTICAL CRYPTOGRAPHY FOR INTERNETWORKS (1996).

[45] It will be shown in later sections that literally thousands of infringing copies can be made on the trek from New York to California.

[46] 17 U.S.C. § 101 (1994) (defining copies as "material objects, other than phonorecords, in which a work is fixed by any method now known or later developed, and from which the work can be perceived, reproduced, or otherwise communicated, either directly or with the aid or a machine or device").

[47] *Id.* (Defining fixed: "[a] work is fixed in a tangible medium of expression when its embodiment in a copy . . . is sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration. A work consisting of sounds, images, or both, that are being transmitted, is 'fixed' . . . if a fixation of the work is being made simultaneously with its transmission.").

[48] "Memory" has one or more of the following characteristics: (1) a high speed of operation; (2) entirely electronic in operation; (3) coupled directly or in close proximity to the main microprocessor of the computer (also referred to as the central processing unit (CPU)); (4) loses all data stored once power is removed; and (5) maintains data for only seconds at a time even when continually powered due to functional state changes. Examples of "memory" include: random access memory (RAM), static random access memory (SRAM), fast static random access memory (FSRAM), dynamic random access memory (DRAM), buffers, caches, queues, register files, video RAM (VRAM), computer screen displays, wire transmissions, wireless transmissions, and optical fiber transmissions.

[49] "Storage" has one or more of the following characteristics: (1) relatively slow in operation when compared to other computer components; (2) at least partly mechanical and not entirely electronic; (3) not coupled directly for access by the main microprocessor or central processing unit (CPU) of the computer; (4) retains binary computer data long after power is removed from the device; and (5) intended to maintain data for months or years at a time. Examples of "storage" devices include: floppy disks, hard disks, tape storage, magnetic drums, read only memory (ROM) integrated circuits, CDs, read/write CDs, ferroelectric memory (such

have held that in particular situations the use of "storage" in an improper manner will lead to copyright infringement.[50] Therefore, the original software owner will claim that the many "storage" copies created on the way from New York to California constitute copyright infringements and one or more of the equipment manufacturers, owners, or operators of the "storage" devices should be held liable for the damage as direct infringers[51] or as contributory infringers.[52]

In addition to finding liability for improper computer "storage" copies, courts have recently held that copies in short-duration "memory" or random access memory (RAM) are of sufficient fixation for purposes of copyright infringement.[53] Memory copies on

as PZT devices), and printer paper. Storage may also include electrically erasable programmable read only memory (EEPROM), EPROM, and flash memory, which border between "storage" and "memory."

[50] *See* Williams Elecs., Inc. v. Artic Int'l, Inc., 685 F.2d 870, 874 (3d. Cir. 1982) (stating that although ROM is utilitarian, the data it contains is fixed and subject to copyright protection; copying of another's data to ROM constitutes copyright infringement); Stern Elecs., Inc. v. Kaufman, 669 F.2d 852, 855-56 n.4 (2d Cir. 1982) (explaining that all portions of the program, once stored in memory devices anywhere in the game are fixed in a tangible medium within the meaning of the Copyright Act); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1249 (3d. Cir. 1983) (explaining that fixation is satisfied through the embodiment of the expression in the ROM devices); Apple Computer, Inc. v. Formula Int'l, Inc., 725 F.2d 521, 525 (9th Cir. 1984) (extending copyright protection to computer programs); Tandy Corp. v. Personal Micro Computers, Inc., 524 F. Supp. 171, 173 (N.D. Cal. 1981) (imprinting a computer program on a silicon chip, which then allows the computer to read the program and act upon its instructions, falls easily within copyright protection); West Publ'g Co. v. On Point Solutions, Inc., *available in* 1994 WESTLAW 778426, at 2 (N.D.Ga. Sept. 1, 1994) (enjoining the selling, distributing, and copying of unauthorized RAM, floppy disk, and CD-ROM copies for purpose of copyright infringement); Computer Assocs. Int'l, Inc. v. Altai, Inc., 1992 WL 139364, at *8 (2d Cir. 1992) (asserting that non-literal structures of computer tape and computer disks were intended to be considered "fixed" for the purpose of copyright); Works Fixed in a CD-ROM Format, 37 C.F.R. § 202.20(c)(xix) (1998).

[51] Direct infringement may be found when a plaintiff can read each element of his claim onto the defendant's invention. *See generally* 5 CHISUM ON PATENTS: A TREATISE ON THE LAW OF PATENTABILITY, VALIDITY, AND INFRINGEMENT, § 16 (1997) (discussing nature of direct infringement of patent claims).

[52] Contributory infringement may be found if a defendant knowingly and materially contributed to a direct infringer's wrongful act. Frequently, several companies will jointly develop and produce a complex device or process, and different parties will contribute pieces to the infringing invention. In this situation, even the noninfringing parties may be held liable as a contributory to the direct infringement. *See generally* 5 CHISUM ON PATENTS: A TREATISE ON THE LAW OF PATENTABILITY, VALIDITY, AND INFRINGEMENT, § 17 (1997) (discussing nature of contributory infringement of patent claims).

[53] *See* MAI Sys. Corp. v. Peak Computer, Inc., 991 F.2d 511, 518 (9th Cir. 1994) (finding that copyright infringement occurs when a computer program is transferred from a permanent storage device to a computer's RAM without the owner's permission); Triad Sys. Corp. v. Southeastern Express Co., 64 F.3d 1330, 1335 (9th Cir. 1995) (explaining that copy-

computers servicing the Internet are more abundant than storage copies, as will be discussed *infra* in Parts III and IV. Due to this recent trend in copyright case law, more computer and communication manufacturers, owners, and service providers may be subject to liability for a single infringing transmission from New York to California.

In some circumstances, federal courts indicated, for public policy reasons, that it is not feasible or rational to expose computer hardware manufacturers, computer owners, software providers, and telecommunications service providers to massive copyright liability for the automatic and unknowing copies created by a culpable third party who inappropriately uses the Internet.[54] While direct infringement has occurred due to the hackers' activity on the Internet, the manufacturers, owners, service providers, etc., will likely be deemed contributory infringers, and contributory infringement requires an element of "knowledge" or "intent."[55] It seems very unlikely that

ing a program in RAM constitutes infringement); Advanced Computer Servs. of Mich., Inc. v. MAI Sys. Corp., 845 F. Supp. 356, 363 (E.D. Va. 1994) (concluding where a copyrighted program is loaded into RAM and maintained there for minutes or longer, the RAM representation of the program is sufficiently "fixed" to constitute a "copy" under the Copyright Act); MicroSparc, Inc. v. Amtype Corp., 592 F. Supp. 33, 35 (D. Mass. 1984) (using a program inputted into a computer constitutes a potential copyright violation); NLFC, Inc. v. Devcom Mid-Am., Inc., 45 F.3d 231, 235 (7th Cir. 1995) (loading software into a computer constitutes the creation of a copy under the Copyright Act); *In re* Indep. Serv. Orgs. Antitrust Litig., 910 F. Supp. 1537, 1541 (D. Kansas 1995) (transferring a computer program from storage device to a computer's RAM constitutes copy for purposes of copyright law); ISC-Bunker Ramo Corp. v. Altech, Inc., 765 F. Supp. 1310, 1332 (N.D. Ill. 1990) (copying a program from disk into the computer's memory directly infringes the copyright).

[54] *See Netcom*, 907 F. Supp. at 1368 (finding neither access provider nor BBS operator liable for direct infringement since access provider did not take any affirmative action that directly resulted in copying, other than installing and maintaining an electronic system whereby software automatically forwarded and stored copies of messages). *But see* Playboy Enter., Inc. v. Frena, 839 F. Supp. 1552, 1554 (M.D. Fla. 1993) (holding the defendant liable for direct infringement although the defendant lacked intent and knowledge); Sega Enters. Ltd. v. MAPHIA, 857 F. Supp. 679 (N.D. Cal. 1994) (finding direct copyright infringement by computer bulletin board company and individual in control of bulletin board whereby video games were uploaded to bulletin board by unknown users and subsequently downloaded by users to make unauthorized copies, which copying was known and facilitated by defendants).

[55] RCA Records v. All-Fast Sys., Inc., 594 F. Supp. 335 (S.D.N.Y. 1984) (analyzing the machine owner's liability and machine manufacturer's liability under the rubric of contributory infringement, not direct infringement). *See also Sega*, 857 F. Supp. at 683 (N.D.Cal. 1994) (finding contributory infringement since defendant solicited the copying of infringing programs and therefore had knowledge); Sony Corp. v. Universal City Studios, Inc., 464 U.S. 417, 487 (1984) (attaching liability for contributory infringement where defendant "with knowledge of the infringing activity, induces, causes or materially contributes to the infringing conduct of

the Internet-enabling parties possessed sufficient knowledge or intent in the above hacker hypothetical to warrant liability of any kind under any contributory infringement or inducement theory. Moreover, it seems equally likely that the lack of volition or act which has any reasonable nexus to the infringing activity would result in many Internet-enabling entities being absolved of any direct liability in a manner similar to the *Netcom* case.[56]

Having registered copyrights for the hacked program, the original software owner may pursue a theory of vicarious liability against infringers under copyright law. Vicarious liability will hold the manufacturers, owners, and service providers on the Internet liable when they: (1) had the right and ability to control the infringer's acts; and (2) received a direct financial benefit from the infringement.[57] Once again, the companies in the hacker hypothetical[58] exert little, if any, control over the hacker's conduct. There has been no direct financial benefit gained by most, if any, of the Internet-enabling entities through contract or through any other means by the infringing activity.[59] The only financial benefit these corporations accrued was the initial sale of the device, sale of Internet software, sale of a computer component, or a service arrangement provided without any infringing software being present. This type of first sale benefit has little nexus to the act of direct infringement. Thus, it will be difficult for the original software owner to prove direct infringement, vicarious liability, or contributory infringement on the part of

another" (quoting Gershwin Publ'g Corp. v. Columbia Artists Management, Inc., 443 F.2d 1159, 1162 (2d Cir. 1971)); Apple Computer, Inc. v. Microsoft Corp., 821 F. Supp. 616, 625 (N.D.Cal. 1993) (noting that "participation of the alleged contributory infringer must be 'substantial'").

[56] *Netcom*, 907 F. Supp. 1361. *See* Orthokinetics, Inc. v. Safety Travel Chairs, Inc., 806 F.2d 1565, 1578 (Fed. Cir. 1986) (finding defendants personally liable for *acts of direct infringement*) (emphasis added).

[57] See *Netcom*, 907 F. Supp. at 1375; Shapiro, Bernstein & Co. v. H.L. Green Co., 316 F.2d 304, 306 (2d Cir. 1963) for discussions of vicarious liability.

[58] See *supra* Part I.A.

[59] Artists Music, Inc. v. Reed Publ'g, Inc., 31 U.S.P.Q.2d (BNA) 1623 (S.D.N.Y. 1994) (finding no vicarious liability for trade show organizers who used fixed fees and thereby did not derive any financial benefit from exhibitor's allegedly infringing musical performances); *Netcom*, 907 F. Supp. at 1376 (holding that defendants must show some control over the infringing party's conduct and direct financial benefit from the infringing activities of its users in order to be held liable under a vicarious liability theory).

any of the above listed Internet-enabling corporations when applying copyright theories of *Netcom.*

In general, the potentially thousands of infringing copies made on the information superhighway path from New York to California constitute direct infringements of the copyrighted software only by the hacker, so a remedy can only come from the hacker, absent contributory infringement. This is because copyright law, under a *Netcom* rationale, has economically-justified liability barriers to protect Internet hardware, software, and service providers in the modern communication environment. These barriers are rooted in such copyright doctrines as contributory infringement, fixation, fair use, and vicarious liabilit as discussed in the case law previously cited. These barriers to copyright infringement and general copyright law, as it relates to the Internet and associated technology, have been the primary focus of intellectual property attention in the legal community.[60] This focus may need to be shifted to another threat: software

[60] Barry D. Weiss, *Barbed Wires and Branding in Cyberspace: The Future of Copyright Protection, in* UNDERSTANDING BASIC COPYRIGHT LAW 1996, at 397 (PLI Pats., Copyrights, Trademarks, and Literary Prop. Course Handbook Series No. 450, 1996); Henry H. Perritt, Jr., *Cyberliability, in* LITIGATING LIBEL AND PRIVACY SUITS 1996, at 173 (PLI Pats., Copyrights, Trademarks, and Literary Prop. Course Handbook Series No. 446, 1996); Rex S. Heinke & Lincoln D. Bandlow, *Roadblocks and Exit Ramps on the Information Superhighway, in* LITIGATING LIBEL AND PRIVACY SUITS 1996, at 203 (PLI Pats., Copyrights, Trademarks, and Literary Prop. Course Handbook Series No. 446, 1996); Sylvia Khatcherian, *Liability on the Internet,* N.Y. ST. B.J., May-June 1996, at 34; Janice R. Walker, *Protecting Cyberspace: Copyright and the World Wide Web,* FED. LAW., May 1996, at 42; Mark C. Morril & Sarah E. Eaton, *Protecting Copyrights On-Line: Copyright Liability for On-Line Service Providers,* 8 J. PROPRIETARY RTS. 2 (1996); Jessica Litman, *Revising Copyright Law for the Information Age,* 75 OR. L. REV. 19 (1996); Peter Jaszi, *Caught in the Net of Copyright,* 75 OR. L. REV. 299 (1996); Karen S. Frank, *Potential Liability on the Internet, in* CABLE TELEVISION LAW 1996, at 417 (PLI Pats., Copyrights, Trademarks, and Literary Prop. Course Handbook Series No. 437, 1996); Joseph V. Myers III, Note, *Speaking Frankly About Copyright Infringement on Computer Bulletin Boards: Lessons to be Learned From Frank Music, Netcom, and the White Paper,* 49 VAND. L. REV. 439 (1996); Ian C. Ballon & Heather D. Rafter, *Computer Software Protection, in* TECHNOLOGY LICENSING AND LITIGATION, at 81 (PLI Pats., Copyrights, Trademarks, and Literary Prop. Course Handbook Series No. 431, 1996); Byron F. Marchant, *On-Line on the Internet: First Amendment and Intellectual Property Uncertainties in the On-Line World,* 39 HOW. L.J. 477 (1996); Giorgio Bovenzi, *Liabilities of System Operators on the Internet,* 11 BERKELEY TECH. L.J. 93 (1996); John Carmichael, *Comment, In Support of the White Paper: Why Online Service Providers Should Not Receive Immunity From Traditional Notions of Vicarious and Contributory Liability for Copyright Infringement,* 16 LOY. L.A. ENT. L.J. 759 (1996); Andrea Sloan Pink, Comment, *Copyright Infringement Post Isoquantic Shift: Should Bulletin Board Services Be Liable?,* 43 UCLA L. REV. 587 (1995); Edward A. Cavazos & G. Chin Chao, *System Operator Liability For A User's Copyright Infringement,* 4 TEX. INTELL. PROP. L.J. 13 (1995); Kelly Tickle, Comment, *The Vicarious Liability of Elec-*

and database patent infringement. Currently, this threat has not been reduced by a *Netcom*-type, liability-limiting decision and could result in significant liability for some Internet-enabling entities under the additional patent theories of importation, exportation, vicarious liability, direct infringement, and contributory infringement and/or inducement.

An important issue is: should software patent law provide a remedy for the original software owner against Internet-enabling entities where public policy, visited under copyright law in *Netcom* and *Sony,*[61] has refused to do so? Currently, no case law suggests that the courts will not follow a very broad reading of "strict liability" and "infringing acts" to impose liability, even direct liability, on some Internet-enabling entities under theories similar to *Sega* and *Frena.*[62] Well-settled patent doctrines, such as direct infringement, contributory infringement, inducement, vicarious liability, infringement due to exportation, infringement due to importation, and the like, in the context of massive patent infringement on the information superhighway, could go either way.

As discussed in Parts II and III, *infra*, of this article, a broad patent law interpretation of direct infringement on the Internet by virtue of making, importing, exporting, or using infringing copies on the Internet should greatly concern Internet-enabling entities. However, if the courts exclude the possibility of direct infringement by these entities, the more defendant-friendly doctrines of inducement, vicarious liability, and contributory infringement should protect most

tronic Bulletin Board Operators for the Copyright Infringement Occurring on Their Bulletin Boards, 80 IOWA L. REV. 391 (1995); Niva Elkin-Koren, *Copyright Law and Social Dialogue on the Information Superhighway: The Case Against Copyright Liability of Bulletin Board Operators*, 13 CARDOZO ARTS & ENT. L.J. 345 (1995).

[61] *See Netcom*, 907 F. Supp. at 1368-69 (holding that when defendant's system made temporary copies of plaintiffs' works, not mean defendant caused the copying); Sony Corp. v. Universal City Studios, Inc., 464 U.S. 417, 446-50 (1984) (finding that entities who merely provide hardware, which is subsequently used by infringers to perform infringing acts, were neither contributory infringers nor liable under the fair use doctrine).

[62] *See Frena*, 839 F. Supp. at 1555 (finding that electronic bulletin board service operator was liable for the transfer of copied, digitized photographs through his bulletin board despite the operator's lack of knowledge of the transfer); *Sega*, 857 F. Supp. at 686 (direct and contributory copyright infringement by computer bulletin board company and individual in control of bulletin board was found when unauthorized copies of video games were made when games were uploaded to bulletin board by unknown users and subsequently downloaded by users to make additional copies, which copying was known and facilitated by defendants).

of the Internet-enabling entities that have no knowledge, volition, or nexus to the infringing act while simultaneously ensuring that the Internet-enabling entities that are acting with knowledge and/or volition will pay the consequences for their culpable acts.

C. Enter the Software Patent

Recent CAFC case law and USPTO policy have validated new styles of software patent claims that may have repercussions on the Internet. The following sections carefully examine new software article of manufacture claims,[63] software means-plus-function claims,[64] and data structure claims[65] recently allowed by the USPTO. These new software claims styles, in combination with the unsettled policy of who should bear the risk of intellectual property damage on the Internet, may cause some concern among Internet enabling entities.

1. The Software Article of Manufacture Claim

If copyright law does not provide an adequate remedy for the hacking activity illustrated in Part I.A., *supra*, an owner of a valid U.S. software patent may seek a remedy under a theory of patent infringement. The software patent issued to the patent holder protects new and novel algorithms or data structures developed for and used within the original software, which have now been illegally copied by the hacker.[66]

[63] Software "article of manufacture claims" are claims covering a computer program or data structure embodied in a computer-readable medium, such as a memory device. For example, an article of manufacture claim could read on a Windows® 95 embodiment in a disk, RAM, or CD.

[64] Software "means-plus-function claims" are structural claims directed towards the physical structure of a computer that contains the novel software for which a patent is sought.

[65] "Data structure claims" are claims, independent of any physical element, that read on the physical structure of data stored in memory. Data is arranged in a computer in some functional manner so that the data is useful.

[66] *See In re* Beauregard, 53 F.3d 1583, 1584 (Fed. Cir. 1995) (finding that computer programs embodied in a tangible medium, such as floppy diskettes, are patentable subject matter under 35 U.S.C. § 101 as articles of manufacture and must be examined under 35 U.S.C. §§ 102-103). *Beauregard* bears no judicial precedence, but the case clearly indicates the USPTO's willingness to accept article of manufacture claims for software inventions. *See also* Guidelines for Examination of Computer-Related Inventions, 60 Fed. Reg. 28778 (1995) (proposed June 2, 1995) (proposing PTO Examiner guidelines for evaluating the patentability of computer-related inventions); Manual of Patent Examining Procedures § 2106 (rev. July, 1996); *In Re* Lowry, 32 F.3d 1579, 1583 (Fed. Cir. 1994) (holding that data structures designed

For the purposes of this subsection, assume that the software patent contains software article of manufacture claims. To infringe these claims, one need only have used, sold, copied, imported, offered for sale, or made the storage structure containing the software. No computer is needed for infringement, and the software on the storage structure arguably need not be executed for infringement. Software article of manufacture claims are a relatively new addition to software patent law.[67] Thus, many practitioners may not be familiar with the form and purpose of this type of claim.

Initially, software patents contained: (1) method claims reciting the process that occurred when the software was executed; and/or (2) apparatus or structure claims which recited a computer structure containing the novel software. Companies which exclusively produced and distributed software on floppy disks or CDs did not execute the software, and therefore, arguably, did not directly infringe the method claims. Additionally, many software manufacturers do not manufacture or distribute computers or computer hardware components which infringe apparatus or structure claims. The software article of manufacture claim developed over the years in an attempt to render software manufacturers as direct infringers, rather than mere contributory infringers.

Software article of manufacture claims first received wide notoriety and scrutiny in *In re Beauregard.*[68] In *Beauregard*, the USPTO Board of Appeals rejected a claim directed to a computer program embodied on a computer readable medium, such as a floppy disk, under the "printed matter doctrine." The "printed matter doctrine" stands for the proposition that an invention primarily embodied on printed matter (e.g., newspapers, paintings, books, advertisements, photographs, etc.) should be protected under copyright law and not patent law.[69] The USPTO and the Federal Circuit held conflicting

to permit computer to run more efficiently "impart a physical organization on the information stored in memory").

[67] *In re* Beauregard, 53 F.3d 1583 (Fed. Cir. 1995) (extending credibility to software patent claims).

[68] *Id.* at 1584.

[69] *See In re* Lowry, 32 F.3d 1579, 1583 (Fed. Cir. 1994) (questioning the applicability of the "printed matter rejection" to computer memory); *In re* Gulack, 703 F.2d 1381, 1384 (Fed. Cir. 1983) (finding no functional relationship between the printed material to the substrate, the USPTO did not extend "patentable weight to the content of the printed matter"). *See*

views concerning the printed matter doctrine for many years; the USPTO refused to allow patent claims to issue if they read on matter found on a printed page or in printed form. The USPTO clung to the doctrine since the early 1970s to curtail an expected flood of software patents, whereas the Federal Circuit generally disliked the doctrine when applied against software.

Beauregard appealed to the CAFC, citing adverse case law associated with the printed matter doctrine.[70] While awaiting a decision, the Commissioner of Patents and Trademarks Office conceded that software claimed as being embodied on a computer readable medium will be patentable subject matter under 35 U.S.C. § 101.[71] Further, the Comissioner stated that these claims must be reviewed under 35 U.S.C. §§ 102 and 103. Finding no case or controversy, since the claims now passed muster under 35 U.S.C. § 101, the CAFC reversed the USPTO Board of Appeals; and the patent was again examined in the USPTO under 35 U.S.C. §§ 102 and 103.

Subsequent to *Beauregard*, the USPTO issued new patent examining guidelines for software-related inventions, which briefly discuss the software article of manufacture claim.[72] In addition to allowing some software article of manufacture claims, these newly proposed guidelines for software inventions weaken or fully discard

also In re Royka, 490 F.2d 981, 985 (C.C.P.A. 1974) (reemphasizing to USPTO that printed matter may well constitute structural limitations upon which patentability can be predicated).

[70] *See Lowry*, 32 F.3d at 1583 (noting that printed matter may well constitute structural limitations upon which patentability can be predicated); *In re* Miller, 418 F.2d 1392, 1396, (C.C.P.A. 1969) (printed matter in an article of manufacture claim can be given patentable weight); *In re* Sterling, 70 F.2d 910, 912 (C.C.P.A. 1934) (patentable novelty cannot be predicated upon printing alone, but must reside in physical structure; the mere arrangement of printed matter on a sheet or sheets of paper does not constitute patentable subject-matter).

[71] *Beauregard*, 53 F.3d at 1584.

[72] *See* MANUAL OF PATENT EXAMINING PROCEDURES § 2106 (rev. July, 1996); *Examination Guidelines for Computer-Related Inventions (Discussion Draft)*, 51 PAT., TRADEMARK & COPYRIGHT J. (BNA) 422, 428-429 (Jan. 25, 1996) (indicating that article of manufacture claims will be allowed over 35 U.S.C. § 101 if they are: (1) limited to a specific computer readable medium; or (2) related to a generic computer readable medium and recite specific software, wherein the software or method performed by the software is novel and produces a useful practical result). "When functional descriptive material [such as data structures or computer programs] is recorded on some computer readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases." *Id.* at 427.

While the *Examination Guidelines Discussion Draft* is not an absolute acceptance of software article of manufacture claims as *Beauregard* may have suggested, it is a step in the right direction.

reliance on the printed matter exception to software patentability.[73] Therefore, these new USPTO software examination guidelines have opened the door for continued software article of manufacture claim allowances in the near future.

A consequence of article of manufacture claims being drafted to include the manufacture of the software onto a floppy disk or CD is that these claims are also arguably infringed by Internet software creation or software transmission on any computer readable medium.[74] The Internet consists of billions of computer-readable elements, all of which provide a basis from which liability for infringement of software article of manufacture claims is possible.

Further, these software article of manufacture claims may be infringed by the mere existence or embodying of the software code on some computer or telecommunications medium.[75] Therefore, any entity creating a copy of a program in RAM or any similar computer readable medium has the potential of infringing the software article of manufacture claims possessed by the original software owner.

Furthermore, a patent claim does not suffer from a fixation limitation as does copyright protection. Even if an entity copies the hacked software for one femto-second,[76] infringement is possible under patent law given a properly drafted software article of manufacture claim. As will be shown herein, computers, modems, memory devices, buffers, state machines, telecommunications nodes,

[73] *See Examination Guidelines for Computer-Related Inventions (Discussion Draft)*, 51 Pat., Trademark & Copyright J. (BNA) 422 (Jan. 25, 1996). The *Examination Guidelines Discussion Draft* expressly states that abstract ideas, natural phenomenon, and laws of nature are unpatentable subject matter, but neglects to address the printed matter doctrine, thereby seeming to abandon this doctrine as it was applied to software patents.

[74] Other readable media include: magnetic tape, optical disc, compact disc (CD), hard disk, floppy disk, ferroelectric memory, electrically erasable programmable read only memory (EEPROM), flash memory, EPROM, read only memory (ROM), static random access memory (SRAM), dynamic random access memory (DRAM), ferromagnetic memory, optical storage, charge coupled devices, smart cards, and similar type storage media.

[75] The *Beauregard*-type claim is an article of manufacture claim or apparatus claim. Merely having software that performs the same indicated steps will directly infringe this apparatus claim; no processing steps or code execution is required for infringement. For example, regarding hacked copies on the Internet, these copies are "made" and "used" for Internet electronic transmission and utilize hardware, software, and equipment owned by the service providers. The software patent holders will argue the hacker's actions constitute software patent infringement under 35 U.S.C. § 271(a)-(d), whereby illicit "making" or "using" of the software comprises the infringing act. 35 U.S.C. § 271(a)-(d) (1988).

[76] One quadrillionth of a second, or $1x10^{-15}$ second.

mainframes, and all other electronic devices function as high-speed micro-factories.[77] It would not be difficult for these high-speed micro-factories to produce new and multiple copies of the hacked code every microsecond, resulting in many thousands of infringing copies every second. These electronic copies may exist for only a fraction of a second and yet, with no fixation requirement in patent law, such transitory copies may subject a legal entity to substantial liability.[78] The above characteristics of the article of manufacture software claim renders this claim style potentially problematic for Internet-enabling entities.

In order to better understand the threat to the Internet enabling entities, consider the following two examples of software article of manufacture claims for a hypothetical piece of computer software that, when executed, does the following: (1) sorts a list of items; (2) searches the list of items; and (3) transmits a selected item from the list.

The first style of manufacture claim follows:

1. A data provider stored on computer readable medium, the data provider comprising:

   a first plurality of computer instructions, which when provided to a CPU, sorts a list of items;

   a second plurality of computer instructions, which when provided to a CPU, searches the list of items to identify a selected item; and

   a third plurality of computer instructions, which when provided a CPU, transmits the selected item.

[77] Currently electrical devices used for software execution and electronic routing operate at frequencies of 200MHz to 800MHz. This theoretically means that approximately 200-800 million operations can occur in a single second when using just one of these devices.

[78] Copyright law may provide a means to avoid the limitation of fixation as well. When a program is transmitted over the Internet, the program is divided into segments. These segments are packaged into "datagrams," and the datagrams are packed into a "frame" which contains additional material to enable Internet transmissions. Thus, one may argue that an Internet transmission creates a derivative work which requires no fixation. *Cf.* Mirage Editions, Inc. v. Albuquerque A.R.T. Co., 856 F.2d 1341 (9th Cir. 1988). *But see* Lewis Galoob Toys, Inc. v. Nintendo of Amer., Inc., 964 F.2d 965 (9th Cir. 1992) ("quasi fixation" discussed for derivative works).

The second style of article of manufacture claim is illustrated below:

> 2. A data provider stored on computer readable medium wherein the data provider is executable by a central processing unit (CPU), the data provider, when executed by the CPU, causes the CPU to comprise:
>
> sort circuitry which provides a list of sorted items;
> identification circuitry which identifies a selected item in the list of sorted items; and
> transmission circuitry which transmits the selected item.[79]

These types of claims were designed to catch in a "direct infringement net" all of the software manufacturers that produced either disks or CDs containing infringing software for mass markets. However, these claims go further than just catching the software manufacturer; these claims are arguably infringed each time infringing code is copied into RAM, CD, tape, disk, ROM,[80] or any

[79] The author has seen many *Beauregard*-type claims using the words "when executed by a central processing unit (CPU)," "computer readable medium," or similar language. Such language should be used carefully for at least two reasons. First, the software may be communicated or purchased from the Internet in a compressed or encrypted form. This raises the issue of whether encrypted or compressed files be presented to a computer for execution. One may argue that a CPU cannot directly process compressed or encrypted files since they do not contain directly usable computer instructions for the CPU. Second, patents issued from applications filed after June 8, 1995 will expire twenty years from the earliest effective filing date. By 2010, it would not be surprising to see "sneaker net" purchasing (i.e., walking to the mall to buy software) replaced by wireless, optical, or satellite purchasing and downloading of software. Is optical light a "computer readable medium?" By avoiding these "when executed by a CPU" and "computer readable medium" limitations, the patent may have a longer, useful life and avoid hyper-technical infringement defenses.

[80] One may ask why manufacturers worry about software infringement on ROM since it cannot be written to with new copies. Article of manufacture claims cause problems for IC manufacturers when with ROM since when an entity orders a product containing ROM, the required software in the ROM is provided to the IC manufacturer. The IC manufacturer then makes masks, or templates, which are used for laying out the pattern of the IC. The masks contain the structures needed to "hardwire" the software in a fixed, physical manner onto the substrate of the IC. In another form, generic ICs are first manufactured in bulk, and subsequent fuse blowing steps or one time programmable (OTP) methods are used to permanently transfer the required code to the IC. In either form, the IC manufacturer is arguably a direct infringer of the software article of manufacture claims even if that company does not own a computer to execute the software. Furthermore, the mask maker may be a contributory infringer. Companies that manufacture microcontrollers containing ROM or OTP memory must

like computer readable medium for even nanosecond durations. As one should see, this has a significant impact on Internet communications since the electronic transmission of software may be infringement of a software article of manufacture claim, in addition to any asserted copyright infringement theories.

The USPTO admission in *Beauregard*, the weakening of the printed matter exception, and the advent of USPTO acceptance of software article of manufacture claims in their examination guidelines are significant. Without article of manufacture claims, software inventions were claimed in structure/apparatus claims and process claims.[81]

Typically, "structure claims" include elements like CPUs, printers, memory chips, display screens, counters, registers, addresses, and other electronic devices.' None of these electronic devices exist within software or on common computer storage devices, such as floppy disks and CDs. The "process claims" contain actions which must be performed, such as printing, transmitting, sorting, encoding, communicating, or receiving. Under the hacker scenario, these actions infringe if the software is executed by someone using a CPU. Arguably, no infringement occurs when the actions are merely routing, shipping, or manufacturing the code onto a computer readable medium. Therefore, the structure or apparatus claims and the process claims do little to discourage a software manufacturer from copying code onto a disk and shipping the disk to customers without the CPU, printer, register, and without actually executing the code.

Despite the inherent advantage of these new software claims, which allow U.S. patent protection for software executed on a personal computer, mainframe, or supercomputer, the following question arose: to what extent can these software article of manufacture claims create liability resulting from passive transmission on the Internet via ownership, operation, control, censoring, and manufactur-

be cautious of software article of manufacture infringement. EPROM and EEPROM can be written to more than once and therefore differ from ROM or OTP components.

[81] "Structure/apparatus claims" read on a physical object, whereas "process claims" read on a method for performing a process.

ing of Internet resources which use, make, export, import, or otherwise effects transmission of TCP/IP datagrams[82] over the Internet?

2. The Software Means-Plus-Function Claim

In addition to the article of manufacture claim, the *Alappat*-style means-plus-function claim[83] is likely to be found in many newly issued software patents and in patents dating back into the mid-1980s. The "means" form provided under 35 U.S.C. § 112, para. 6, is a way of reciting structure, functionally. Therefore, a computer that contains novel and patentable software can be claimed as a means/structure for implementing the novel software algorithm.[84] The software means claims usually read on a computer that contains the software, whereby some portion of the computer (not just the disk or CD, as in *Beauregard*) is needed to show infringement.

An example of the *Alappat* means-plus-function claim directed to the same hypothetical example used above for the article of manufacture claim is:

3. A data provider comprising:
   means for sorting a list of elements to form a sorted list;
   means for searching for a selected item in the sorted list; and
   means for transmitting the selected item to another location.

In the case *In re Alappat*, the CAFC held that claims similar to the claim listed above create "a new machine, because a general purpose computer in effect becomes a special purpose computer once it is programmed to perform particular functions pursuant to instructions from program software."[85] Under this theory, if the patents owned by the original software owner contain novel algorithms claimed in a means-plus-function apparatus form, then every modem, memory chip, computer, etc., that copies the software into

[82] When transferring data or software on the Internet over telecommunications facilities, the data and/or software is broken into one or more segments and packaged into TCP/IP datagrams for transmission. The Internet packaging includes the packet of data and a header.

[83] *See supra* note 64.

[84] *In re* Alappat, 33 F.3d 1526, 1545 (CAFC 1994) (en banc).

[85] *Id.* at 1545; *see also In re* Iwahashi, 888 F.2d 1370, 1374 (Fed. Cir. 1989).

RAM or a like computer readable medium arguably infringes the claim by creating a "new machine." However, it can be argued that any claim drafted in a means-plus-function format should be read as a "new machine" or "new computer" under *Alappat* and not simply as a memory device storing novel computer code. *Alappat* mentions article of manufacture claims in passing[86] and seems to disapprove of any interpretation of means-plus-function claims as a software article of manufacture claim.

The Court in *Alappat* stated, "[a] patent can be awarded to one who 'invents or discovers' something within the enumerated classes of subject matter — 'process,' 'machine,' 'manufacture,' 'composition of matter.' These terms may not be read in a strict literal sense entirely divorced from the context of the patent law."[87]

The Court further stated that with respect to article of manufacture claims,

> if a claim to a compact disc or piano roll containing a newly discovered song were regarded as a 'manufacture' and within § 101 simply because of the specific physical structure of the compact disc, the 'practical effect' would be the granting of a patent for a discovery in music.[88]

Therefore, there is room for one to argue that *Alappat* means-plus-function claims should not be read as broadly as an article of manufacture claim, but should be limited to a "new machine" as per the holding of *Alappat.* This interpretation of the claim may itself create some unique problems on the Internet.

If the software means-plus-function claim is not limited to a new machine and can be extended to the "new article of manufacture," then the software means-plus-function claim may be infringed anywhere and in any manner that the *Beauregard* article on manufacture claims are infringed. However, if the software means-plus-function claim is limited to a "new machine," then two possible scenarios exist.

[86] *Alappat,* 33 F.3d at 1541-1543, 1552-1553.

[87] *Id.* at 1553. *Alappat* suggests that simply providing an article of manufacture claim is not enough. One must look to all other judicially created doctrines of patent law and 35 U.S.C. § 101 case law to determine patentability of an article of manufacture in the software technological area.

[88] *Id.* at 1554.

In the first scenario, the simple presence of the software on a "machine" constitutes an infringing act regardless of whether the software is executed or not.[89] The software's presence within a memory qualifies as a "new machine" without needing to be executed. Thus, even code temporarily residing in memory during its travel over the information superhighway, where the memory can be linked to a CPU, infringes a valid software patent.

However, in the second scenario, the software may need to be executed on the machine before infringement of software means-plus-function claims occurs.[90] In this case, passive transmission or routing of a software program as data on the information superhighway may not be enough for infringement of *Alappat* claims.

Therefore, while the *Alappat* claims may be of interest to Internet enabling entities, the *Alappat* means-plus-function software claims may be more narrowly applicable than the *Beauregard* article of manufacture claims for software routed or communicated, via the Internet. This narrower application is due to the fact that *Alappat* means-plus-function claims usually contain "active" functional limitations under the sixth paragraph of 35 U.S.C. § 112.[91]

The hacked software communicated over the Internet is typically not executed to perform the functions recited in the *Alappat* claims when being routed on the Internet. Rather, in many of the routing computers and memory devices on the Internet, the ability to

[89] *In re* Certain Surveying Devices, 214 U.S.P.Q. (BNA) 900 (U.S. Int'l Trade Comm'n 1981) (holding that although mere capability does not constitute infringement, "reasonable capability to infringe" constitutes infringement).

[90] Hap Corp. v. Heyman Mfg. Co., 311 F.2d 839 (1st Cir. 1962) (noting that it is not what the infringing device might have done, but it is what the device was intended to do or actually did); Oak Indus., Inc. v. Zenith Elecs. Corp., 726 F. Supp. 1525 (N.D. Ill. 1989) (awarding damages only for those devices found to actually infringe and not for those devices merely capable of infringement); Berkey Photo, Inc. v. Klimisch-Repro, Inc., 388 F. Supp. 586 (S.D.N.Y. 1975) (holding that infringement cannot be established by what the device might do, but infringement must be established on what the device actually did).

[91] An "active" functional limitation differs from a "passive" functional limitation. An active functional limitation is one that states, for example, "code for performing X." This statement in a claim implies that in order to infringe, the code must be performing X or capable of performing X, or else there is no infringement. A "passive" functional limitation is one that states "code which is capable of doing X," "code which is adapted to do X," or "code which, when provided to a compatible CPU, does X." These passive limitations are arguably infringed if the code is merely capable of eventually performing X, and need not be currently executing X.

execute the software to perform the functions recited for the "means" in the claims may not be possible.[92] Since every element of a claim must be infringed to result in liability, if the device or service through which the code is passed does not execute the code to perform the "function" portion of the means-plus-function claim, there may be no infringement.[93]

Some cases state that it is not what the "machine"[94] actually does, but what the machine is capable of doing that triggers infringement.[95] The other line of cases hold the opposite view

[92] Routing computers and memory devices on the Internet may be unable to execute the software commands since the file may be: (1) encrypted and not directly executable; (2) compressed and not directly executable; or (3) packaged in a protocol-layered manner, packaged into frames, or formed into data packets (e.g., TCP/IP datagrams) for communications over the Internet and are therefore not directly executable by any CPU. In addition, there may be hardware compatibility issues. If the hacked code is written to execute on a Pentium™ processor, then the hacked file which is transferred via a UNIX Sun workstation, which cannot execute Pentium™ code, cannot be executed on the UNIX system. Likewise, the router communicating the hacked software on the Internet may be used solely for routing purposes and not equipped with the physical memory-to-CPU connections to execute the code in any manner.

[93] Johnston v. IVAC Corp., 885 F.2d 1574, 1580 (Fed. Cir. 1989) (holding that where a claim does not read an accused device exactly, there can be no literal infringement of the claim); Mannesmann Demag Corp. v. Engineered Metal Prods. Co., 793 F.2d 1279, 1282 (Fed. Cir. 1986) (noting that literal infringement requires every element of the patent claim be met); Hilton Davis Chem. Co. v. Warner-Jenkinson Co., 62 F.3d 1512, 1562 (Fed. Cir. 1995) ((1) the doctrine of equivalents may not be used to enlarge a claim; infringement requires that every limitation of a claim must be met by at least an equivalent substitute in the accused product or process; (2) legal equivalency of a substitute is established by the knowledge in the art at the time the patent is issued; and (3) the substitute must perform essentially the same function as the element which it replaces so that overall the accused product or process and the claimed invention can be said to operate by substantially the same means to achieve the same or substantially the same result), *rev'd sub nom.* Warner-Jenkinson Co. v. Hilton Davis Chem. Co., 117 S. Ct. 1040 (1997) (adhering to doctrine of equivalents, but reversing and remanding on federal court's failure to consider all requirements as related to prosecution history estoppel); Pennwalt Corp. v. Durand-Wayland, Inc., 833 F.2d 931, 935 (Fed. Cir. 1987) (holding that equivalents must by found on an element-by-element basis or a limitation-by-limitations basis within the claims; if limitation is missing, then no equivalents); Markman v. Westview Instruments, Inc., 52 F.3d 967, 1000 (Fed. Cir. 1995) ("a claim is not infringed unless every element thereof is met in the accused device, either literally or by an equivalent"); Texas Instruments, Inc. v. U.S. Int'l Trade Comm'n, 846 F.2d 1369, 1372 (Fed. Cir. 1988) ("It is now settled law that each element of a claim is material and essential and, in order to find infringement, the patent owner must show the presence of every element or its substantial equivalent in the accused device.").

[94] The "means" of the Alappat means-plus-function claim creates a new "machine."

[95] Stearns-Roger Mfg. Co. v. Ruth, 87 F.2d 35, 38 (10th Cir. 1936) (reasoning that if devices were designed so they could operated normally in an infringing way, immaterial that some customers chose not to operate them in that manner); Kearney & Trecker Corp. v. Giddings & Lewis, Inc., 452 F.2d 579 (7th Cir. 1971) (finding infringement even if the device was used in a noninfringing manner since the device could also be used in an infringing manner).

whereby even if a machine can carry out a claimed function but does not actually perform the claimed function, then the machine does not infringe the claim.[96] In any event, regardless of possible problems associated with asserting the *Alappat* claims as related to Internet transmission infringement, currently the *Alappat* means-plus-function claim is more commonly used in U.S. patents than the *Beauregard* article of manufacture claim. Thus, Internet-enabling entities should not ignore *Alappat* claims.

3. The Data Structure Claim

Another software claim style which could be infringed when hacked software routes along the information superhighway is the software data structure claim.[97] While the contents, or the data itself, cannot be patented, the arrangement or physical/organized memory structure used to store the data may be claimed.[98] Furthermore, the USPTO could not conveniently ignore printed matter or data structure limitations in a claim when these types of limitations are present in a claim.[99] The court in *In re Lowry* extended the "new machine" holding from *Alappat* to data structures by holding that a new data structure transferred to a computer creates a "new machine."[100] However, the *Lowry* claim was packed with structural limitations such as "memory" and may not have pushed the data structure claim style to a limit.[101]

[96] *See* cases cite *supra* note 90.

[97] *See supra* note 65.

[98] *In re* Lowry, 32 F.3d 1579, 1583 (Fed. Cir. 1994) (holding data structure as patentable under 35 U.S.C. §§ 102 and 103 since the structure recited was seen as specific electronic structural elements in a computer memory and data structure not analogous to printed matter).

[99] *Id.* at 1582.

[100] *Id.* at 1583.

[101] The *Lowry* claim was recited as follows:

> 1. A memory for storing data for access by an application program being executed on a data processing system, comprising: a data structure stored in said memory, said data structure including information resident in a database used by said application program and including:
>
> a plurality of attribute data objects stored in said memory, each of said attribute data objects containing different information from said database;
>
> a single holder attribute data object for each of said attribute data objects, each of said holder attribute data objects being one of said plurality of attribute data objects, a being-held relationship existing between each attribute data object and its holder attribute data object, and each of said

Suppose one or more of the claims in the software owner's software patents recites a data structure claim. Unlike the *Alappat* means-plus-function claim, the *Lowry*-type data structure claim is infringed without requiring specific use, processing steps, or computer execution involving the data structure.[102] An infringing data structure, which is potentially being transferred from New York to California in a copied database portion of the hacked software, is recreated into electrical or magnetic structures all along the information superhighway route or is put into infringing use by the Internet. In a manner similar to the hacked software covered by the *Beauregard* claims, the original software owner believes that the copies of the hacked database portion exposes many copying parties to liability.

It should be noted that while *Lowry* held that a data structure claim could be found patentable under certain circumstances, abstract recitations to "data structure" as the only structural limitation within the claim may not render the claim patentable under 35

> attribute data objects having a being-held relationship with only a single other attribute data object, thereby establishing a hierarchy of said plurality of attribute data objects;
> a referent attribute data object for at least one of said attribute data objects, said referent attribute data object being nonhierarchically related to a holder attribute data object for the same at least one of said attribute data objects and also being one of said plurality of attribute data objects, attribute data objects for which there exist only holder attribute data objects being called element data objects, and attribute data objects for which there also exist referent attribute data objects being called relation data objects; and
> an apex data object stored in said memory and having no being-held relationship with any of said attribute data objects, however, at least one of said attribute data objects having a being-held relationship with said apex data object.

Id. at 1581. Note that the term "memory" in a claim may help overcome 35 U.S.C. § 101 rejections. The word "memory" may have done so in *Lowry*, and "memory" avoided a 35 U.S.C. § 101 rejection in *In re* Warmerdam. *In re* Warmerdam, 33 F.3d 1354 (Fed. Cir. 1994) (holding that the dependent claim 5 was patentable subject matter while the base claim 1 was unpatentable subject matter, and the only structural addition added to claim 1 by claim 5 was "memory").

[102] In this respect, *Lowry*-type claims are more like *Beauregard*-type claims where mere creation, usage, or existence of the object under a defendant's control may incur liability. However, *Lowry*-type claims recite data structure and not software code embodied on a computer readable medium.

U.S.C. § 101.[103] In *In re Warmerdam*, claim 1[104] was found to contain unpatentable subject matter, under to 35 U.S.C. § 101. The dependent claim 5[105] was found to contain patentable subject matter, pursuant to 35 U.S.C. § 101, since claim 5 identified "memory" as the only relevant structural limitation in addition to referencing claim 1. However, Warmerdam's claim 6,[106] which recited "data structure" as the only relevant limitation in addition to referencing claim 1, was found to be unpatentable subject matter under 35 U.S.C. § 101.

The lesson learned from the combination of *Warmerdam* and *Lowry* is that the CAFC will probably view the term "memory" as a material limitation under 35 U.S.C. § 101 which renders the claims as patentable subject matter, whereas a recitation to simply a "data structure" may be ideologically abstract and interpreted in a nonstructural manner thereby rejectable under 35 U.S.C. § 101. However, the need for a limitation such as "memory" in a data structure claim seems hardly likely to render this claim-style less suited for Internet infringement in the future.

In any event, data structure claims will now be found to be patentable in many circumstances. The communication of hacked data structures through the Internet could pose liability concerns for Internet-enabling entities in addition to the transmission of executable software as discussed above with respect to *Alappat* and *Beauregard.*

[103] *See Warmerdam*, 33 F.3d at 1358 (finding that a claim containing "memory" was patentable over 35 U.S.C. § 101, whereas a claim reciting "data structure" was not patentable over 35 U.S.C. § 101).

[104] Claim 1 of *Warmerdam*:

> 1. A method for generating a data structure which represents the shape of physical object in a position and/or motion control machine as a hierarchy of bubbles, comprising the steps of: first locating the medial axis of the object and then creating a hierarchy of bubbles on the medial axis.

Id. at 1357.

[105] Claim 5 of *Warmerdam*: 5. A machine having a memory which contains data representing a bubble hierarchy generated by the method of any of Claims 1 through 4. *Id.* at 1358.

[106] Claim 6 of *Warmerdam*: 6. A data structure generated by the method of any of Claims 1 through 4. *Id.*

II. The Types of Electronic and Magnetic Copies Routinely Created on the Information Superhighway - The Technical Reason for Concern

Internet services, Internet software, computer hardware, and telecommunication systems are capable of creating many different types of "memory" and "storage" copies.[107] Some of these copies may easily result in either direct or contributory patent infringement liability given the three styles of claims discussed above. Other copies will be limited to contributory infringement theories, while still other copies will be incapable of resulting in any patent liability under any circumstances. In order to make a proper determination of infringement, jurisdiction, who is a proper defendant to sue, etc., the types of copies created on the information superhighway and where they are created must be understood.

A. The Slavish Total Copy

One type of copy which is made over the information superhighway is the "slavish total copy." The author defines a slavish total copy as a 100% total reproduction of the claimed software or claimed data structure at any one point in time within a single structure, either produced by one legal entity or under the control of one legal entity. For example, a posting of an infringing program on America On-Line would result in at least one 100% total copy being made onto the resources provided or controlled by America On-Line. A copy made onto a computer hard disk is usually a 100% total copy of the software which can later be executed. A tape backup contains a 100% total copy of the 100% software copy resident on the hard disk. A floppy disk can contain a 100% copy. A computer may contain enough RAM to store the entire 100% copy from hard disk into RAM. For short patented software code, one TCP/IP datagram on the Internet may be a 100% slavish total copy. Typically, when a hacker copies code off of the Internet, a slavish total 100% copy must be made somewhere in some memory since the receiving computer must have access or control over a 100% copy in order to render the software completely functional.

[107] *See supra* Part I.B. for a discussion of "memory" as compared to "storage."

The slavish total copy has a significant impact in the realm of patent law. First, for direct infringement, all elements of the claimed invention be made, used, sold, offered for sale, or imported.[108] Direct or literal infringement of a U.S. software patent can be easily shown if a slavish total copy is found anywhere on the information superhighway within the United States. The question remains, "Who is responsible for the damages associated with the direct infringement?" Is the hacker, the manufacturer of the hardware, the owner of the hardware, the Internet service provider, and/or the telecommunication company liable for the damage? Second, contributory infringement of a defendant cannot be found absent the finding of direct infringement in some form.[109] Therefore, if there is no direct infringement, all contributory infringement claims will fail. Furthermore, favorable jurisdiction may very well depend upon where the direct infringement occurred.[110]

In summary, parties should always be quick to find slavish total copies on the information superhighway since their presence, or lack thereof, will significantly impact subsequent legal analysis and strategy.

[108] *See* 35 U.S.C. § 271(a) (1994); Aro Mfg. Co. v. Convertible Top Replacement Co., 365 U.S. 336, 344 (1961) (noting that the combination patent covers only the totality of the elements in the claim and not each element, separately viewed, within the patent grant); Interdent Corp. v. United States, 531 F.2d 547, 552 (Ct. Cl. 1976) (explaining that omission of a claimed element from a combination patented avoids infringement); Panduit Corp. v. Stahlin Bros. Fibre Works, 575 F.2d 1152, 1156 (6th Cir. 1978) (indicating that infringement of a claimed combination requires the presence in an accused structure of each claimed element, or its equivalent).

[109] Carborundum Co. v. Molten Metal Equip. Innovations, Inc., 72 F.3d 872, 876 & n.4 (Fed. Cir. 1995) (noting that absent direct infringement of patent claims, there cannot be contributory infringement or inducement of infringement under 35 U.S.C. § 271); Aro Mfg. Co. v. Convertible Top Replacement Co., 377 U.S. 476, 483 (1964) ("if there is no direct infringement of a patent there can be no contributory infringement"); Wells Mfg. Corp. v. Littelfuse, Inc., 547 F.2d 346, 350 n.5 (7th Cir. 1976) ("contributory infringement is defined in terms of direct infringement").

[110] *See* 28 U.S.C. § 1338 (1994); Fourco Glass Co. v. Transmirra Prods. Corp., 353 U.S. 222, 225 (1957) (noting the subject matter jurisdiction of district courts over patent and copyright infringement actions over defendant, whether a person, partnership, or corporation); Art Leather Mfg. Co. v. Albumx Corp., 888 F. Supp. 565, 566 (S.D.N.Y 1995) (finding no jurisdiction where the accused infringer had no business connection with the forum and no infringing activity was performed in the forum); Electro Med. Sys., S.A. v. Cooper Life Sciences, Inc., 34 F.3d 1048, 1057 (Fed. Cir. 1994) (requiring that both sale and infringing activities occur to support subject matter jurisdiction).

B. The Time Serial Copy

A "time serial copy" is a copy that is formed in one device either manufactured or under the control of one legal entity. However, the *entire* claimed invention is not present in the device at any one overlapping point in time.[111] This type of copy is especially important if the serial device is sending data out of the United States in an exportation manner. The time serial copy is best discussed with respect to Fig. 1:

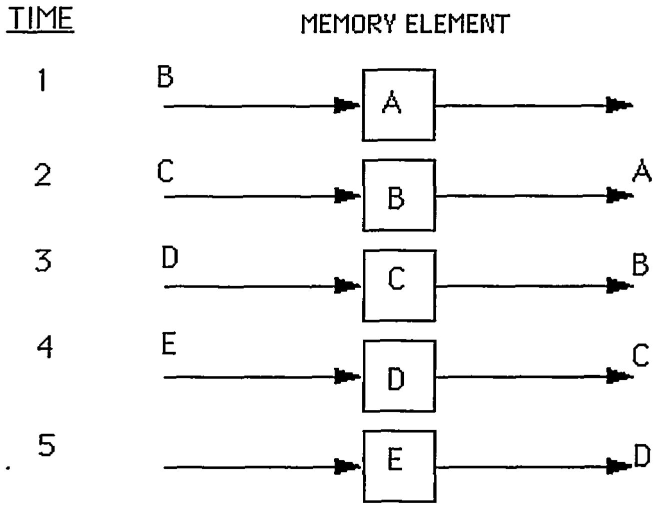

FIG. 1

Many hardware and software operations on the information superhighway and within modern computer systems result in information being stored in a time serial manner as illustrated in Fig. 1. With respect to Fig. 1, suppose a software article of manufacture claim recites a computer program or a data structure having five pieces A, B, C, D, and E, all of which are recited in the software

[111] For example, if time period A ran from 0 to 10 seconds, time period B ran from 3 to 13 seconds, and time period C ran from 6 to 16 seconds; then A, B, and C would be overlapping time periods, between time 6 to 10 seconds, since A, B, and C ran concurrently.

claim.[112] The memory device illustrated in Fig. 1 receives and copies element A into memory in a first time period 1. When storing A, the memory device may either have no room for both B and A or be operating at a speed that is so fast that A is provided as output and erased from the device before B arrives at the input of the memory device. In any event, A is transmitted as output signals and erased from the memory before a time 2, and B enters the memory device and is stored in the memory device in the time 2.[113] In a time period 3, following time period 2, B is erased after proper output transmission and C enters the memory device and is copied. By the end of time period 5, the memory device has copied the entire program into RAM in a time sequence of A, then B, then C, then D, and finally E. However, none of A through E were resident in the memory during an overlapping time period and all of A-E were never resident in the memory at one time.

Time serial copies are very common on the information superhighway and within modern computer systems. Time serial copies are made through telecommunications switching nodes, through modems, on printer spooling systems, through the pipeline(s) of a computer CPU, through a satellite link, in buffers of a CPU, through peripherals that feed a hard disk, over Ethernet lines, in some caches, etc. Many computer and telecommunications memory devices can be viewed as high-speed, time serial manufacturing lines where the entire patented program or data structure is created/copied in sequential incremental pieces, and an entire copy of the hacked software is never stored in the device at any one time. For a time serial copy, a 100% total copy of the program may be serially inputted, copied, outputted, and erased within a mere fraction of a second.

Time serial copies may be direct infringement under 35 U.S.C. § 271(f) and will also expose some entities to contributory infringement claims.[114]

[112] A, B, C, D, and E may each be data bytes, packets of data bytes, individual bits, subroutines of a software program, data elements in a data structure, or similar items.

[113] Depending upon the type of device, Times 1-5 of Fig. 1 may be separated by nanoseconds or several hours.

[114] *See* discussion of this patent law, *infra* Part IV.C.

C. *The Partial Copy*

A partial copy is when the entire software program or data structure is never copied at one point in time like a slavish total copy, nor is a total copy automatically made in a time serial manner over a specified period of time as in the case of a time serial copy. In order to illustrate partial copies, Fig. 2 is provided:

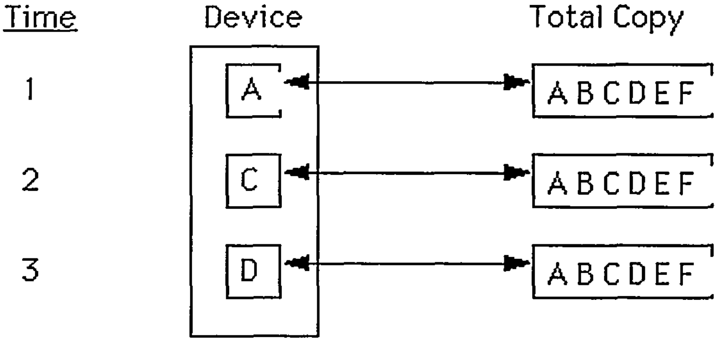

FIG. 2

Assume that a patent holder has a claim on an algorithm having six parts, A through F as illustrated, in main memory in Fig. 2. For the circuitry of Fig. 2, a slavish total copy is provided on a hard disk (not illustrated in Fig. 2). The CPU within the computer, which has access to the hard disk, contains two levels of cache[115]. Fig. 2 illustrates one of these two cache levels labeled as "DEVICE" (hereinafter "device").

The cache only stores parts of the program as needed by the CPU and may never, even if left running for several years, copy all parts A through F of the program - unlike the time serial copy and the slavish total copy which will eventually copy the entire program when given enough time and/or memory space. The partial copy will rarely be a slavish total copy or a time serial copy unless combined with some other device, either manufactured by another entity

[115] Cache memory is a special, fast memory containing active segments of computer programs so that the total execution time of the program is reduced.

or under the control of another entity. Therefore, unlike a time serial copy or a slavish total copy, the likelihood of being a direct infringer of a software claim when responsible for making a partial copy is substantially reduced.

However, if the patent holder has a valid patent claim reciting only the routine D, the use in Fig. 2 is direct infringement since the device makes a slavish total copy of D in Time 3.[116] Furthermore, if the patent holder has a claim that includes only A, C, and D, the functioning of the device in Fig. 2 creates a time serial copy as shown by the dashed line in Fig. 2. Therefore, a partial copy, as illustrated in the device in Fig. 2 can result in later creation of a total copy (unlikely) or a time serial copy (more likely) depending upon the scope of the issued U.S. software claims. The probability that a device, which usually makes partial copies, will eventually make a time serial copy or a slavish total copy increases as the storage capacity of the device increases and as the time of use of the device increases. Some examples of devices which are likely to make partial copies are the cache of a CPU, the CPU pre-fetch buffer, and computer RAM paging from a hard disk. Also, routers and switching nodes on the Internet are designed to transmit from a transmitting point to a receiving point along many different paths whereby each path may never be presented with the entire program or data structure but may only be given pieces of the program or data structure. In these cases, many communication paths on the Internet will create only a partial copy.

D. The Selective Partial Copy

A subset of the partial copy is the "selective partial copy." A selective partial copy, unlike the partial copy, will never be capable of making a total copy regardless of its storage capacity, volume limitation, or its time of access to an infringing 100% total copy. The selective partial copy is best explained referencing Fig. 3:

[116] Software claims may be directed to one million lines of code, a few kilobytes of code, or a single computer instruction or opcode. Some very small memory devices, or even a single TCP/IP datagram, may create either a slavish total copy or result in a time serial copy when the claimed invention is a small piece of code or single instruction.

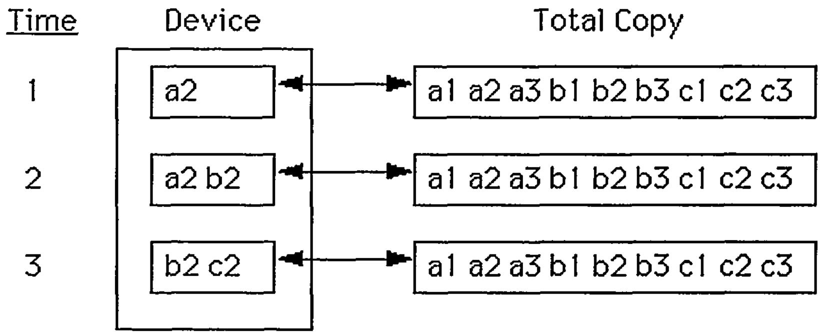
Time
Device
Total Copy
1
a2
a1 a2 a3 b1 b2 b3 c1 c2 c3
2
a2 b2
a1 a2 a3 b1 b2 b3 c1 c2 c3
3
b2 c2
a1 a2 a3 b1 b2 b3 c1 c2 c3

FIGURE 3

In Fig. 3, assume a patent assignee has a patent on a software program which has three subroutines A, B, and C. The patent claims all of A, B, and C in the independent claim of the patent. Each routine A, B, and C contains three types of information. Information with a "1," such as "a1," "b1," and "c1," are branch instructions to be executed by a CPU; information with a "2," such as "a2," "b2," and "c2," are data elements; information with a "3," such as "a3," "b3," and "c3," are all other computer instructions other than branch instructions.

In computer systems, some devices are functionally designed to only copy or store certain types of information from a computer program or a data structure to satisfy a particular purpose. Video RAM (VRAM) copies only graphics information from the executing program; a branch cache stores only the branch instruction elements a1, b1, and c1 in some predetermined form; a data cache (D-cache) will store only the data items a2, b2, and c2 as illustrated in Fig. 3; an instruction cache (I-cache) will store only instructions a3, b3, and c3; and a sound card will process only sound data from the executing program. Even if the entire program is executed or accessed by a CPU or computer, these special purpose devices are capable only of copying certain segments of each element A, B, and C of the claimed device. Therefore, unlike a partial copy which can create an infringing total copy or a time serial copy, given sufficient storage space and time of exposure to the infringing program, the selective copy will never do so. The devices which make selective partial copies will probably never make copies which result in direct infringement since selective partial copies must be coupled with other devices in order to use the entire claimed subroutines A, B, and C. Thus, the creator of a partial copy is most likely only liable under a contributory infringement theory.

A subset of a selective partial copy is the "random partial copy." In many cases, a device makes a copy of an object due to sheer chance. In some cases, there is no apparent algorithm or criterion which determines what device is to copy the infringing code. Resources may be shared on a round-robin basis or randomized schedule; lightning may strike a telephone line, destroying a TCP/IP datagram which must subsequently be retransmitted through an otherwise non-infringing computer. Under such circumstances, the data did not designedly arrive at that particular device. Rather by chance, that particular device created the infringing copy due to an unpredictable event.[117]

E. The Repetitive/Redundant Copy

Once a device makes a copy, that device may erase that copy and make the same copy again at a later time for various functional reasons. For example, when two router nodes of an Internet communication are transmitting a data packet (i.e., datagram) of hacked software using the TCP/IP protocol, an error may occur in transmission. This error may be due to noise in the communications link. This noise may destroy the entire, or a significant portion, of the datagram; and error detection and correction may not be possible. Since the datagram cannot be recovered, the whole packet is re-sent, creating a second infringing copy through the Internet due to the error.

Another example of repetitive copying involves CPU caching. Once a piece of software code or data is cached in a CPU, it may be

[117] Both the Fiber Distribution Data Interface (FDDI) system and TCP/IP can compensate if an unforeseen error or uncontrollable event interrupts the datastream by making copies of the packet to other devices.

erased over time by more current information. After erasure, the same previously-erased data or program portions may be recopied again into the same device at a later time for a new use.

In yet another example, redundant or repetitive copies may be created to ensure the long-term integrity of the data stored. To understand this concept, one need only look to a common, one-transistor DRAM memory cell illustrated in Fig. 4:

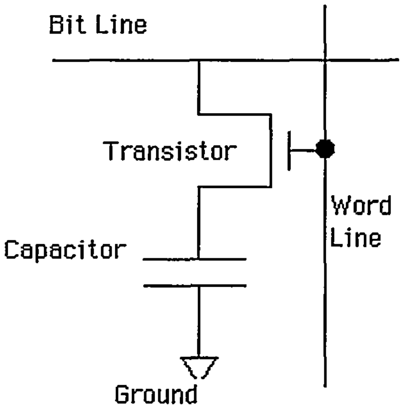

FIGURE 4

A single DRAM cell which stores one bit[118] of information is illustrated in Fig. 4. A modern DRAM integrated circuit (IC) manufactured today will contain either sixteen million of the above cells or sixty-four million of the above cells, plus all of the control and sensing logic to read, write, and maintain data values in these millions of DRAM cells.

The DRAM cell of Fig. 4 is programmed with either a logical one (a voltage above ground, say 2.5 volts by way of example) or a logical zero (a ground voltage, which is 0 volts). Suppose that the DRAM cell of Fig. 4 will contain a logical one value. To program the DRAM cell of Fig. 4 to a logical one state, the bit line of Fig. 4 is set to a logical one, or 2.5 volts; and the word line is also activated so that the transistor is turned on. Since the transistor is turned

[118] The bit is either a logical one (high voltage) or a logical zero (low voltage) state.

on, the 2.5 volts on the bit line can access the capacitor through the transistor and charge the capacitor to a stored voltage of roughly 2.5 volts. After the capacitor is charged, the transistor is turned off (by deactivating the word line). When the word line is deactivated, the capacitor is effectively isolated from the external environment by the disabled transistor and left to retain the charge of a logical one.

However, once the capacitor of Fig. 4 is isolated with the logical one charge in place, the capacitor will begin to "leak" the stored charge over time. A voltage on the capacitor that was once programmed to 2.5 volts will dissipate to 2.2 volts, then 1.9 volts, and then 1.5 volts, whereby the voltage will eventually degrade down to zero volts absent some sort of electrical intervention. In order to ensure long term retention of the data in an error-free condition, the control circuitry of the DRAM "refreshes" the DRAM cell many times per second. In order to refresh the DRAM data, the electrical values in the DRAM cells are read from each cell within the DRAM device (creating a time serial copy), using detection circuitry, and then re-written to the DRAM cells (creating another copy, possibly a slavish total copy at every refresh cycle) to the 2.5 volt level. Thus, the capacitor voltage on a DRAM cell over time, when programmed to a logical one state, looks like:

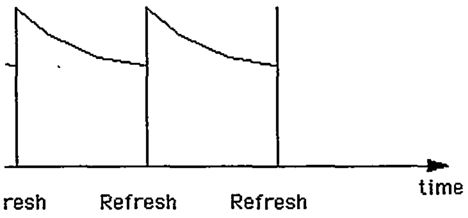

FIGURE 5

each refresh, a new copy of the program
y one of these copies can raise issues,
lity and jurisdiction. However, redun-

dant and repetitive copies are more academic than of practical significance since only one copy is needed to result in liability. Massive amounts of illegal copying by an entity may make that entity appear very culpable to those unfamiliar with the technology. Such illicit copying will affect whom the plaintiff can file a lawsuit against, the appropriate jurisdiction for filing the claim, and the cause of action asserted in the claim — direct or contributory infringement, or inducement of infringement.

However, the number of unauthorized copies created does not affect the economic damages since the number or frequency of copies has no real bearing on patent damages.[119] For example, if a first person sends a copy of Microsoft Word 6.0™ to a second person via the Internet for illegal use by that second person, the Internet transmission may create thousands of unauthorized copies. However, Microsoft, the owner of any patent rights in the software, could only claim damages under theories of reasonable royalty and/or lost prof-

[119] Damages may be based upon "established royalty;" *see* Rude v. Westcott, 130 U.S. 152, 165 (1889); Faulkner v. Gibbs, 199 F.2d 635, 638 (9th Cir. 1952) (accepting a royalty as "established," it must be set prior to the infringement currently complained of, paid by an adequate number of people to imply reasonableness, and uniform at the places where the licenses are issued); Deere & Co. v. International Harvester Co., 710 F.2d 1551, 1557 (1983) (rejecting the cost paid by one licensee as a measure of established royalty is permissible).

Damages may be based upon reasonable royalty; *see* Georgia-Pacific Corp. v. United States Plywood Corp., 318 F. Supp 1116 n.1 (S.D.N.Y. 1970) (citing 35 U.S.C. § 284); Panduit Corp. v. Stahlin Bros. Fibre Works, Inc., 575 F.2d 1152, 1157 (6th Cir. 1978); Del Mar Avionics, Inc. v. Quinton Instrument Co., 836 F.2d 1320, 1326 (Fed. Cir. 1987) (noting that damage award must not be less than a reasonable royalty); Fromson v. Western Litho Plate & Supply Co., 853 F.2d 1568, 1574 (Fed. Cir. 1988) (setting a reasonable royalty as the floor below which a damage award may not fall); Polariod Corp. v. Eastman Kodak Co., 16 U.S.P.Q.2d (BNA) 1481, 1484 (D. Mass. 1990).

Damages may be calculated based upon plaintiff's lost profits or the infringer's financial gain; *see* Story Parchment Co. v. Paterson Parchment Paper Co., 282 U.S. 555 (1931) (holding that jury may award damages based upon just and reasonable inferences when exact and precise calculation not possible due to wrongdoer's actions); TP Orthodontics, Inc. v. Professional Positioners, Inc., 20 U.S.P.Q.2d (BNA) 1017, 1021 (E.D. Wis. 1991); Sun Prods. Group, Inc. v. B & E Sales Co., 700 F. Supp. 366, 383 (E.D. Mich. 1988) (concluding that "lost profits" was the only proper measure of damages); Scripto-Tokai Corp. v. Gillette Co., 788 F. Supp. 439 (C.D. Cal. 1992) (stating that a lost profits award requires both "but for" causation of infringement and a proper, lost profits computation); Datascope Corp. v. SMEC Inc., 879 F.2d 820, 827 (Fed. Cir. 1989); Paper Converting Mach. Co. v. Magna-Graphics Corp., 745 F.2d 11, 21 (Fed. Cir. 1984) (finding that the "but for" causation requirement was satisfied for lost profits when "reasonable probability of sale" exists); Kaufman Co. v. Lantech, Inc., 926 F.2d 1136, 1140-1142 (Fed. Cir. 1991); Lam, Inc. v. Johns-Manville Corp., 718 F.2d 1056, 1064 (Fed. Cir. 1983). Patent infringement damages are not based simply on a blind assessment of the number or frequency of copies made.

its for the single copy of Word 6.0™ and not for the multiple copies created.

F. The Derivative Copy

For patents of mechanical devices, machines and articles of manufacture may be infringing under many theories. One may use literal infringement (is the claim language identical to the accused device?),[120] a function-way-result equivalents test,[121] or an "insubstantial differences" equivalents test.[122] If these theories fail, a plaintiff may attempt to use a theory of contributory infringement[123] or inducement of infringement[124] to support of finding of liability. One may attempt to use agency law or even vicarious liability theories.[125] Each of these theories may fail to produce a remedy for certain types of unauthorized copying on the Internet. This is especially

[120] Markman v. Westview Instruments, Inc., 116 S. Ct. 1384 (1996) (outlining the modern technique for claim interpretation and determination of direct/literal infringement); Engel Indus., Inc. v. Lockformer Co., 96 F.3d 1398, 1405 (Fed. Cir. 1996) ("Literal infringement of a claim exists when every limitation recited in the claim is found in the accused device, i.e., when the properly construed claim reads on the accused device exactly.").

[121] Graver Tank & Mfg. Co v. Linde Air Prods. Co., 339 U.S. 605, 608-609 (1949) (holding that if the allegedly infringing product performs substantially the same function, in substantially the same way to obtain the same result, then it may infringe the patented invention). *See* Pennwalt Corp. v. Durand-Wayland, Inc., 833 F.2d 931, 935 (holding that legal equivalents must by found on an element-by-element basis within the patent claims).

[122] Hilton Davis Chem. Co. v. Warner-Jenkinson Co., 62 F.3d 1512, 1517-19 (Fed. Cir. 1995) (applying the insubstantial differences test for patent infringement under the doctrine of equivalents, which supports the function-way-result methodology but adds additional potential factors for a court to consider), *rev'd sub nom. on other grounds* Warner-Jenkinson Co. v. Hilton Davis Chem. Co., 117 S. Ct. 1040 (1997).

[123] *See supra* note 52.

[124] "Inducement of infringement" is similar to contributory infringement, whereby the inducing party enables a third party to infringe a patent. The inducer actually causes, urges, directly aids in, or promotes the third party's infringing actions. Fromberg, Inc. v. Thornhill, 315 F.2d 407, 411 (5th Cir. 1963).

[125] Von Holdt v. Husky Injection Molding Sys., Ltd., 887 F. Supp. 185, 187 (N.D. Ill. 1995) (plaintiff alleged that under principles of the agency law, defendant has directly infringed plaintiff's patent by supervising, directing and controlling the manufacture of the infringing molds). See also Robert O. Bolan & William C. Rooklidge, *Imputing Knowledge to Determine Willful Patent Infringement*, 24 AIPLA Q.J. 157, 183-90 (1996) (proposing a standard for imputing knowledge for determining willfulness of patent infringement, based on the law of agency, which would clarify who within a company has to have knowledge).

true since patent law does not contain a true analogy to the derivative work doctrine found in copyright law.[126]

For example, copies of a patented program could be reproduced and distributed through the United States, arguably without infringing its software article of manufacture claims. The program copies, processed for electronic routing and Internet transmission, may escape infringement challenges by the patented program's owner by asserting it: (1) does not function in the same way; (2) has substantial differences; and (3) will not have the same result. Moreover, the copies may not even be recognized as identical to the patented code for contributory infringement purposes. Even so, the patented program's owner can assert that somewhere, someone assisted the infringer in making a 100% total copy and this contributory infringer should be held liable for damages.

When a piece of code is transmitted on the Internet today, it is altered from an executable format to a different format. The code is broken into segmented data packets for piecemeal transmission by the Internet packing protocol (IP), and these packets contain framing information, error correction coding, parity information, source addressing, and/or data addressing. This information, or the program itself, may have been compressed using common data compression techniques so that it is no longer directly executable. Or, the information may be encrypted to a binary format which substantially differs from an executable format. Anonymous remailers may afford individuals the opportunity to communicate anonymously.[127]

If the software article of manufacture claim contains elements stating, "a first set of computer instructions which when executed by a CPU performs an operation," then will the packaged, encrypted,

[126] Internet packets or datagrams are units of data transferred from the computer's memory and packed into a "frame" containing additional information to enable Internet communication. In this respect, TCP/IP data packets are similar to derivative works. *See supra* note 78 and accompanying text. While derivative works are within the subject matter of copyright law(which is particularly attractive for plaintiffs alleging infringing activities on the Internet since copyright does not require "fixation" to infringe derivative works), patent law does not provide this avenue of protection.

[127] Anonymous remailers are computers that strip any identifying characteristics of source from TCP/IP datagrams so a recipient computer cannot determine from who or where the data or software was transmitted. Thus, who should the patent holder sue when he cannot identify the source of the infringing transmission?

compressed, non-executable, and anonymous packet traveling along the telephone lines infringe every element or limitation of this claim? If one takes a packet from the TCP/IP protocol and tries to execute it on a computer, the computer will fail to execute the code. Likewise, if one takes compressed code and tries to directly execute it on a computer, the computer execution will fail. Finally, if one takes encrypted code and tries to directly execute it on a computer, the computer execution will, again, fail.

These derivative copies are problematic to patent holders since defendants may try to escape liability for patent infringement by asserting hyper-technical defenses. Patent drafters must be aware of these types of copies and draft claims encompassing these modification or write specifications whereby these changes are found to be within the equivalents analysis.[128] This is especially true as online commerce plays an expanding role in the purchase and transfer of software.

G. *The Transitory Copy or "δ Copy"*

When data is transmitted over conductive copper wire, aluminum integrated circuits ("IC") interconnects, optical fiber, etc., it takes time for the signal to travel from one end of the conductive line to the other. While in transit, the signal is present on the conductive line for a brief, but stable, moment, similar to data stored in computer memory. Since patent law has no fixation requirement, these transitory or delta (δ) copies, existing for a brief moment and then disappearing, may constitute an infringing copy of the patented software invention. For example, a telephone transmission line is typically modeled by a T-model or a Π-model as illustrated in Fig. 6 below:

[128] Hilton Davis Chem. Co. v. Warner-Jenkinson Co., 62 F.3d 1512, 1537 (Fed. Cir. 1995) (holding that the "specification remains important, not as the definition of the invention, but as a description of it and as an aid in interpreting the language of the claims"), *rev'd sub nom. on other grounds* Warner-Jenkinson Co. v. Hilton Davis Chem. Co., 117 S. Ct. 1040 (1997).

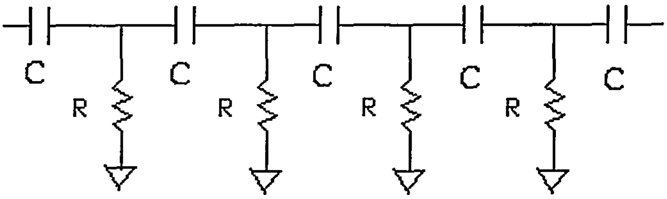

FIGURE 6

A voltage applied across the depicted wire moves a signal from the right end of the line to the left end of the line only after charging the capacitors ("C") along the path. Since electronics devices operate very quickly, the charge need not be present on the line for very long before being recorded in memory at the left end, and the charge on the line is quickly replaced with a charge for a following signal in a very short time serial manner. In a sense, a transmission line, as illustrated in Fig. 6, is similar to the DRAM illustrated in Fig. 4 due to their similar capacitive operation. If DRAM copies infringe patent claims under existing case law, it will be difficult to successfully argue that other capacitive copies do not also constitute patent infringement, regardless of storage duration (i.e., fixation).

III. Understanding Electronic and Magnetic Memory Storage Along the Information Superhighway - A Simplified Walk Through Internet-Enabling Technology

Now that the software patent claim styles are understood and the types of copies created on the Internet have been introduced abstractly, it may be helpful to understand exactly where and why these different types of Internet copies are being made along an Internet communication path and in modern computer systems. Those readers uninterested in a detailed explanation of the technology involved with electronic and magnetic memory storage on the Internet, this section can be skimmed without detracting from the legal content.

A. Memory Copies Made to Enable Long-Distance Data Telecommunications

Fig. 7 illustrates a much-simplified telecommunications system which could electronically interconnect New York to California during the hypothetical hacker discussion introduced in Part I.A., *supra.*

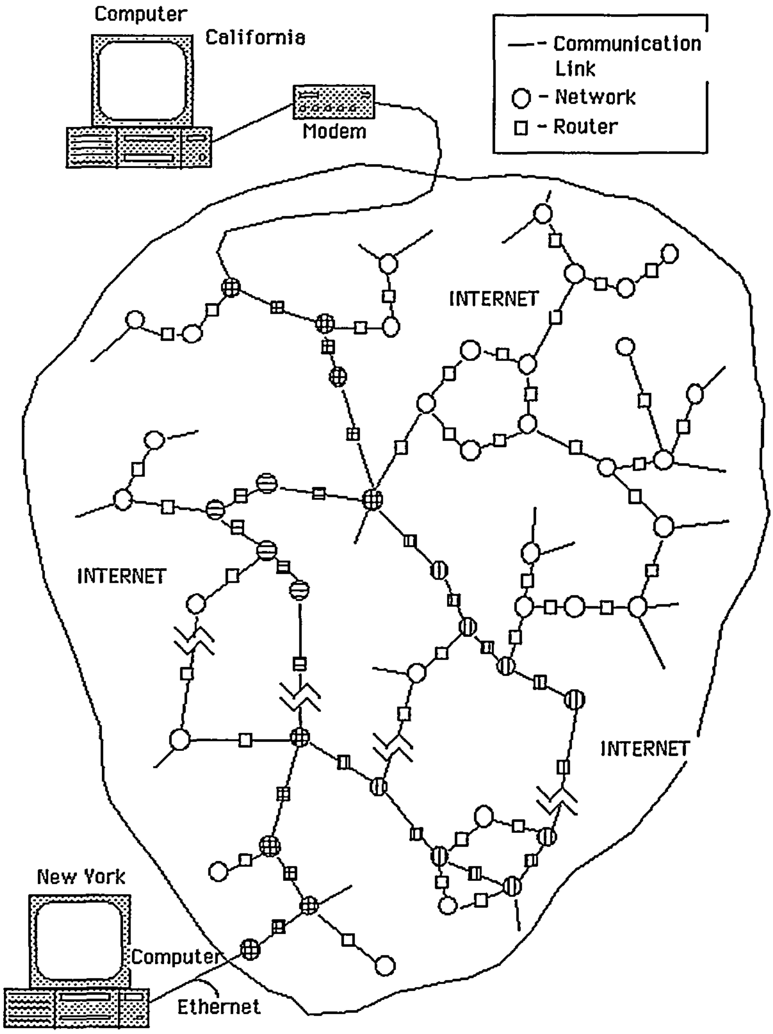
Computer
California
Modem
—- Communication Link
○ - Network
□ - Router
INTERNET
INTERNET
INTERNET
New York
Computer
Ethernet

FIGURE 7

The hacked software is transmitted from New York to California via the Internet, depicted in Fig. 7 by the large centrally-oriented "cloud." Currently the Internet encompasses thousands of computer networks[129] joined together by computers, referred to as "Internet routers." These routers are special purpose computers which regulate the flow of data at each connection point, or node and then determines the best routing for a packet across the Internet by receiving, copying, using, and re-transmitting the data packets. Each network illustrated in Fig. 7 contains one or more computers or sub-networks of computers. Typically, a network or sub-network will contain tens to hundreds of individual computers, but it is not unusual for a single network to comprise of thousands of computers with each computer communicating to another via different electrical and software means.

Internet information is transferred from computer-to-computer and router-to-router along the Internet using a protocol known as transmission control protocol with Internet packing (TCP/IP).[130] Generally, the IP portion of this protocol segments the transmitted software, data structures, and/or data into smaller packets of data. For example, a typical computer program may be time-serially segmented into 1,000 data packets for independent across the Internet. [131] These packets are then individually packaged into a standard packet format for electronic delivery.

The header of a typical IP datagram contains a version number,[132] a header length field,[133] service type field,[134] total length

[129] *See supra* note 6.

[130] For a detailed understanding of the workings of TCP/IP, many texts are publicly available in bookstores or libraries. REGIS J. "BUD" BATES & DONALD GREGORY, VOICE AND DATA COMMUNICATIONS HANDBOOK (1996); DOUGLAS E. COMER, INTERNETWORKING WITH TCP/IP (2d ed., 1995).

[131] Packet switching of data is a simple concept that has been used for decades. Essentially, data packet switching works by breaking down a large file into smaller, more manageable and separate chunks of data. Rather than transmitting a 100 megabyte file as one large file, the file will segmented into thousands of smaller files and transmitted separately in time and by different physical paths, if necessary, to a destination that reassembles the smaller files to recover the 100 megabyte file. If a smaller file is lost, then only the missing smaller file needs to be re-sent and not the entire 100 megabyte file, thereby saving time and resources. *See* David L. Wilson, *A Failure to Connect*, SAN JOSE MERCURY NEWS, Apr. 20, 1998, at 1D.

[132] Identifies the version of IP used to transmit and package the datagrams.

[133] Gives the length of the header in 32-bit quantities so that the receiving end software can easily separate header information from data contained in the datagram.

field,[135] identification field, flags, fragment offset field,[136] time to live field,[137] protocol indicator,[138] header checksum,[139] source IP address,[140] destination IP address,[141] and optional options section.[142]

These datagrams are then presented in a time serial manner to a serial connection of computers, networks, and routers, as illustrated in Fig. 7. These computers further process and package the datagrams, per any hardware limitations, and then physically route the datagrams from New York to California.

A typical voice telephone call uses circuit switched networks which establish a dedicated, end-to-end transmission path for the entire duration of the call. However, packet-switched networks, used to route data over the Internet, use connectionless delivery systems. Unlike a vacationer who plans the exact route to get from home to the vacation spot, the IP datagram knows its starting point and destination but has no idea what route it will use to reach that destination. The exact path used to get each data packet to its ultimate destination is determined "on the fly" at each computerized stop along the information superhighway and may vary as between packets.[143] The result is each TCP/IP datagram, which together

[134] Specifies a priority of the datagram and the type of transport or routing algorithm that should occur on the Internet.

[135] Gives the total length of the datagram in octets (8-bit values) where the maximum size of a datagram is 65,535 octets.

[136] These three fields help further fragment the datagram to accommodate hardware communication protocols and hardware constraints. Since IP datagrams can be 65,535 octets long but some network technologies cannot handle frames of 63,535 octets length, the datagram may require further segmenting to meet hardware limitations. For example, FDDI can only handle 4470 octets per frame while Ethernet can only handle up to 1492 octets per frame.

[137] From Fig. 7, it is clear that some data packets routing along the Internet may travel long and indirect paths. In these cases, packets can get lost or inefficiently routed, needlessly consuming significant bandwidth on the Internet. Therefore, each datagram is given a lifetime on the Internet. Each router subtracts from the lifespan as time goes on. If the lifetime reaches zero before the datagram arrives at the destination, the datagram "self-destructs" and an error is sent back to the source to notify of the "self destruction."

[138] Identifies which protocol formatted the data of the datagram.

[139] Helps identify errors that have occurred in the header during transmission.

[140] Identifies the datagram's source or origination location.

[141] Identifies the datagram's final destination.

[142] Used primarily for network testing and debugging.

[143] Various factors will affect the path the datagram will travel from its current position to its ultimate destination. Such factors include: loading of routers and computers, failures on the Internet, closest paths, and paths with the least amount of noise.

comprise the entire hacked software, may take a different physical route to arrive at the same destination.

Thus in Fig. 7, a first datagram of the hacked software is illustrated as following a path distinguished by vertical graphical hatching, whereas a second datagram of the hacked software follows a path illustrated by horizontal graphical hatching. As shown in Fig. 7, some datagrams of the hacked software will travel the same network links and visit the same routers while journeying from New York to California. Some routers and computers will route a seemingly random subset of the datagrams of the hacked software on different paths, and other routers may direct only a single datagram of the hacked software. Many other computers and routers on the Internet will never see a single datagram of the hacked software. Thus, each computer in each network, and each router along the Internet route from New York to California, may make one or more slavish total copies,[144] random partial copies,[145] time serial copies,[146] repetitive copies,[147] and/or derivative copies of the hacked software,[148] as previously discussed.

The TCP portion of the protocol is the "smarts" of the protocol. The TCP portion tracks the order of the data packets so that they can be properly reassembled in the correct order at the intended destination.

When errors occur or the transmission of a datagram fails, nonacknowledgment signals are sent to the source of the failed datagram. TCP initiates and coordinates the sending of the error signals and the re-sending of this datagram to complete proper transmission

[144] If the patented software invention is small and fits into several thousand bytes of code, a single routed packet may infringe the claims, by itself.

[145] Due to the connectionless nature of TCP/IP paths, each node on the Internet has the capability of making a copy of at least one, or a subset of all, of the datagrams containing the software.

[146] Even if the patented software does not fit into a single datagram, "bottleneck" routers and computers that either feed commonly used wide area networks or feed the inputs and outputs to the receiving computer in California or the transmitting computer in New York are likely to create time serial copies since the entire time serial transmission must pass through these physical connection points.

[147] The transmission may encounter errors which will be corrected via TCP by initiating the re-transfer of erroneous datagrams.

[148] Copies that are encrypted, anonymously remailed, compacted, segmented, and packaged into IP packets are derivatives of the original software.

of the whole hacked software program. Together, TCP/IP and a web of computers, networks, and routers all transmit the hacked software from New York to the hacker in California.

Given that the Internet contains millions of computers and routers, it is not difficult to understand from Fig. 7 that the electronic trek from New York to California may involve copies made by thousands of computers across the country. In addition, due to the connectionless TCP/IP system, some copies may be created in Canada, Mexico, or other countries en route from New York to California thereby involving export and import patent laws. Satellites may be involved in wireless communications (e.g., Iridium communications satellites currently orbit over the earth), especially in the near future as technology rapidly progresses in this area.

Considering Fig. 7 and the above discussion, one could mistakenly assume that if the hacked software encountered 100 computers between New York to California, then 100 unauthorized copies were made. This is not so. Every computer that receives one or more datagrams and any computer executing the software will create many more copies than just one copy per computer. The next few sections demonstrate how and why multiple copies of the hacked code are created.

B. Memory Copies Made Within a Modern Computer Architecture

Fig. 8 illustrates a computer system in a simplified block diagram. On the Internet, a communications packet of data may encounter a desktop computer, such as a Macintosh ci or an IBM x386 machine; or it may encounter supercomputers, IBM mainframes, or other complex computer systems. Regardless of the complexity, all computers encountered by the data packets will contain some microprocessor coupled to some form of computer memory, as illustrated in Fig. 8.

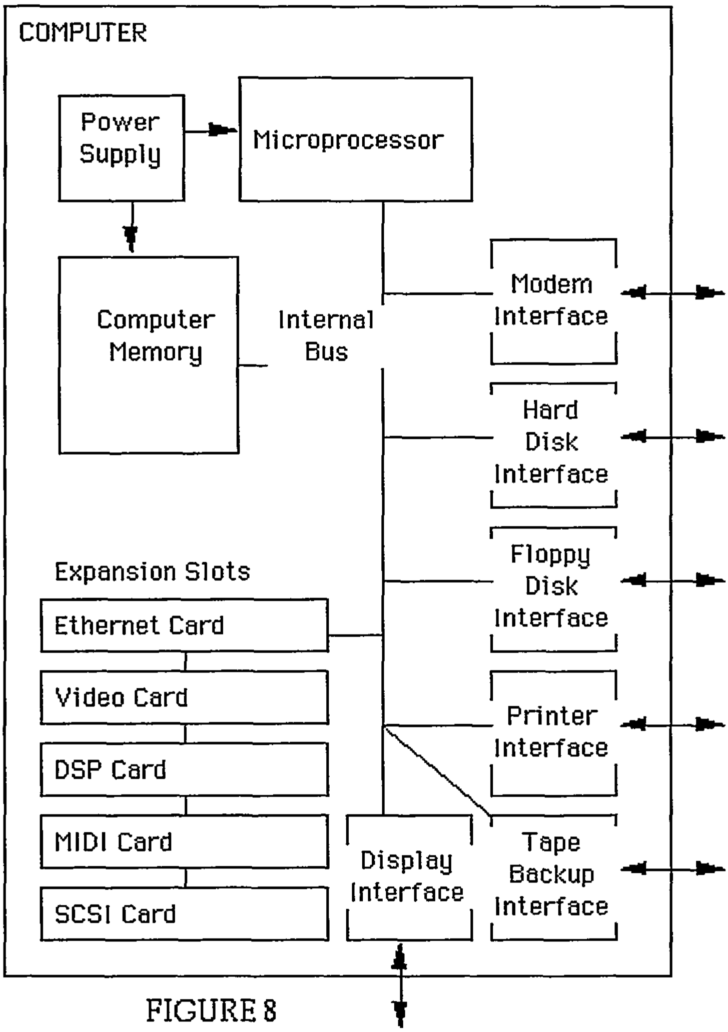

FIGURE 8

FIGURE 8

A distinction must be made between a computer transferring the hacked code as data (which will be done by the routers and most computers on the Internet) and a computer that will execute the hacked program once fully transferred over the Internet.

If the computer in Fig. 8 is a router or a packet switching computer for the Internet data packet (assume router for this scenario), then the hacked program, or portions of the hacked program, will be received by way of the Ethernet card or modem and be copied into the router's memory via a direct memory access (DMA) device, microcontroller, or the microprocessor. Copies will be created and used by the router's internal CPU and memory, and the data packet will then be re-transmitted through a modem or an Ethernet device to the next node. Copies made during these transmission operations may be slavish total copies, time serial copies, random partial copies, or other types, as discussed, *supra*, in Part II.

If the computer of Fig. 8 is *executing* the hacked software, the computer will receive the data via the Ethernet card or the modem as discussed above. A DMA device, microcontroller, the microprocessor, or like device will copy the program to the memory. The hacked program will then be read in pieces by the microprocessor from the memory and executed to perform useful work. During the program execution or through use of the operating system of the computer, the program may be printed, saved to floppy or hard disk, be sent out on the modem, copied on magnetic tape in the course of routine backup, display data on a monitor using VRAM, process sounds or data via a DSP card or sound card, make music on a MIDI card, or communicate with other devices via a SCSI interface, as illustrated in Fig. 8.

In all, one can roughly approximate that each router and packet switching computer which receives with the hacked code will create five copies (Ethernet or modem input copy, microcontroller or DMA transfer to memory copy, resident memory copy, DMA retransfer copy, and Ethernet or modem outgoing copy). If the program is executed, it may create roughly seven copies within the simplified diagram of Fig. 8. One may assume that if 100 router computers are encountered on the Internet, roughly 100x5 copies will be made to route the software with a few more copies being created to execute the program. As will be seen, *infra*, the approximated "500 copies" of hacked software is a very conservative number from the actual number of copies created using modern memory and microprocessor circuits.

C. *Memory Copies Made Within Modern Computer Memory Devices*

Fig. 9 illustrates a simplified, but common, electronic architecture for a typical integrated circuit ("IC") memory component used in a computer microprocessor. Specifically, the circuitry illustrated in Fig. 9 is a DRAM memory array, and the memory access and control circuitry to make the DRAM memory array functional.

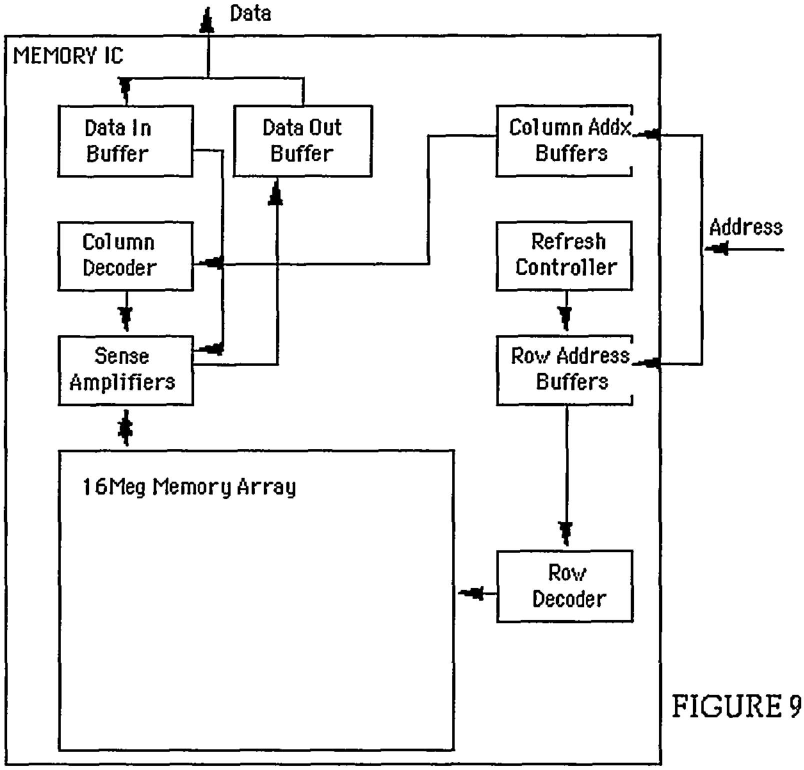

FIGURE 9

FIGURE 9

The DRAM IC in Fig. 9 is a 16 megabyte (16MB) chip which contains approximately sixteen million memory bits. Each bit stores either a "0" (off) or a "1" (on) where collectively a larger group of these bits represent a software program, database, etc. The memory array is written to contain data by providing the data to a Data In Buffer via the Data parallel interface, as shown in Fig. 9. While data is provided for writing the specific addresses identifying the location in the memory array where the incoming data will be stored is presented as Address input. Time serial copies of data are created in the Data In Buffer, while selective time serial copies of the addresses are made in the Column Address Buffers and the Row Address Buffers.

Most memory devices employ at least a two-dimensional or two-level address decoding scheme.[149] The memory array is a two-dimensional arrangement of memory cells across a substrate with each cell storing one bit of digital information. Thus, a Column Address portion and a Row Address portion will uniquely identify a single memory cell among the millions of DRAM cells located on the IC.

The written digital data moves from the Data In Buffer to the Sense Amplifier where the digital data is converted to one or more analog storage pulses. The address information is fed to Row Decoders and Column Decoders which perform the two-level decode and memory cell identification operation. Once the proper cell(s) are identified as the target cells, the data is written from the Sense Amplifier to the appropriate cells via control signals from the Row Decoder and Column Decoder. Therefore, every item of data copied into the memory is saved as data, stored in a buffer, and addressed via Row and Column Address Buffers which create selective time serial copies.

To read the stored data, the reverse operation occurs. Addresses, which identify the cells to read from, are stored in the Row and Column Address Buffers. The addressed cells are accessed via control signals from the Row and Column Decoders in response to the incoming addresses through the Sense Amplifier. The Sense Amplifier converts the analog signal to a digital signal and sends it to the Data Out Buffer for communication to the external world via the Data line.

Every read and write operation creates more than one copy, even in the simplified memory schematic of Fig. 9. Further, after

[149] The use of two-level and multi-level decoding systems are well-known and used in many facets of life. For example, many street maps divide a geographic area into squares, and the reader may locate specific streets or landmarks on the map by using the grid coordinates. Likewise, the postal office uses a multi-level decoding system to deliver mail; to send a letter in the United States, one must provide a street address, name of city, name of state, and zip code. By dividing a large location into two or multiple levels, errors are reduced and the simplicity of use and design are achieved.

the data is stored, the data or voltage begins to deteriorate, as depicted in Fig. 5,[150] so that redundant copies need to be made by the Refresh Controller in Fig. 9 to avoid data loss. Even ignoring the refreshing operation, four or five copies may be generated in every memory IC encountered on the Internet. Thus, the 500 copies approximated in Part III. B., *supra,* are quadrupled to 2,000 copies due to the memory architecture used in common computer systems.[151]

D. Memory Copies Made Within a Modern High-Speed Microprocessor

Fig. 10[152] below illustrates a modern microprocessor[153] of the type commonly used in computer systems. The specific microprocessor illustrated below is a PowerPC™[154] 604 microprocessor which is currently manufactured and distributed by both Motorola and IBM. The PowerPC™ 604 microprocessor is installed in personal computers (PCs) and workstations that may be linked to the Internet. By understanding the high-level workings of the processor in Fig. 10, the reader will also gain an understanding of how other types of modern microprocessors work.

[150] *See supra* Part II.E.

[151] 2,000 copies is, again, a conservative estimate given the complexity of modern memory architectures, pipelining, DMA control, microcontrollers, buffering, etc.

[152] Motorola Inc. owns the copyright to Fig. 10, used herein with permission.

[153] Also referred to as a central processing unit (CPU).

[154] PowerPC is a trademark of IBM.

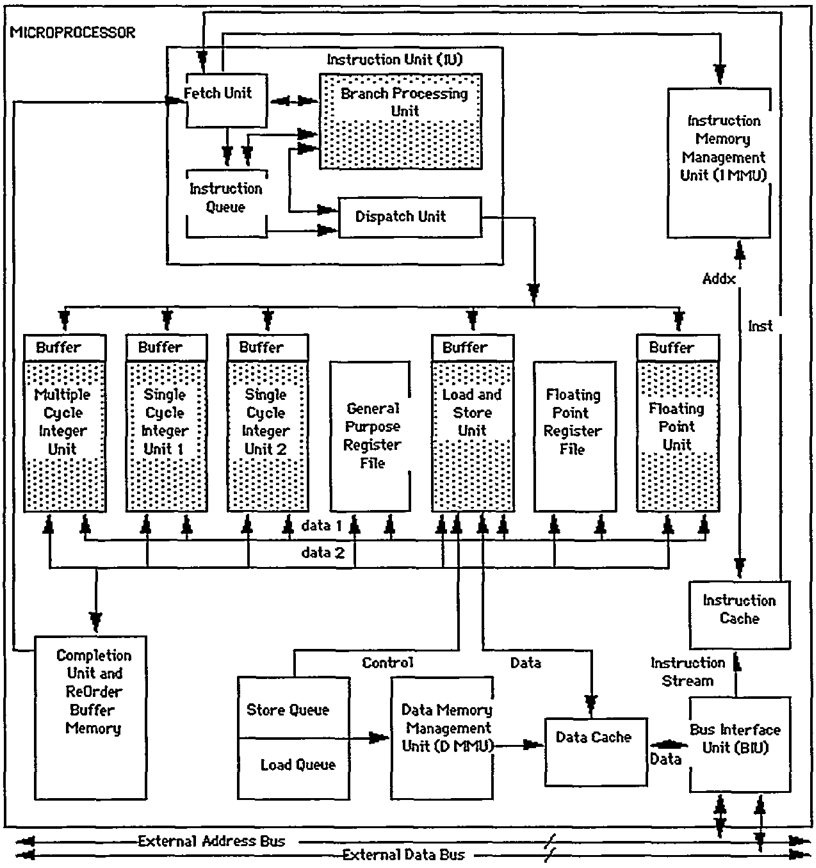

FIGURE 10

In Fig. 10, the components which perform computer instruction execution are shaded while the non-shaded components are "overhead" and simply provide information to the shaded units. The high-level operation of a modern CPU is not complicated. A Bus Interface Unit (BIU) is illustrated in Fig. 10 as part of the microprocessor. The BIU accesses memory[155] and peripherals external to the microprocessor using an address bus, a data bus, and control signal. Most data and computer instructions provided to and from the microprocessor are routed through the BIU; the BIU is the microprocessor's connection to the external world. The BIU creates time serial copies of the program and data structures stored in external memory when transferring this information to and from the shaded units depicted in Fig. 10.

The BIU in Fig. 10 splits data and computer executable instructions read from external memory into two separate electrical paths within the microprocessor. Instructions are cached/stored in an Instruction Cache (I-Cache) while data is cached/stored in a Data Cache (D-Cache), where the I-Cache and D-Cache are separate memory units within the microprocessor.[156] Many other microprocessors have a unified cache where both data and instructions are stored in the same internal cache memory. The caches in Fig. 10 will make selective copies or selective time serial copies of the software and data from external memory.

Data from the D-Cache is provided to and from the shaded areas of Fig. 10 through use of a Store Queue, a Load Queue, and a Data Memory Management Unit (DMMU). The Store and Load Queues maintain temporary copies of stores[157] and loads[158] which are to occur in the D-Cache, thereby performing a buffering-type function.[159] The DMMU is the control unit or special purpose CPU which regulates the read, write, and control of the data in the D-

[155] Similar to memory devices, as discussed *supra* in Part II.

[156] For example, a computer instruction like "ADD x,y" commands a computer to add the number "x" to the number "y." The computer would place the ADD instruction into the I-Cache and the data "x" and "y" into the D-Cache.

[157] A "store" operation takes data from within the microprocessor (e.g., Pentium™ or PowerPC™ 604) and writes the data to a memory device external to the microprocessor (e.g., 32 Megabyte DRAM SIMMs).

[158] A "load" works in reverse from a store operation. A load takes data from external memory and provides it internally to the CPU; therefore, loads and stores provide a bi-directional path for computer code and data between the memory devices and the CPU.

[159] Buffers are used so that different units may run at different speeds without performance or data loss. If a device provides five items which need to be processed within a time N and the receiving unit can only process one item in time N, then a buffer stores the remaining four items until the receiving unit can process them.

Cache. Additional copies of computer data may be made in the DMMU and the Load and Store Queues.

Instructions are provided to the shaded components in Fig. 10 through a Fetch Unit. The Fetch Unit is a special purpose CPU section responsible for reading instructions from external memory via the BIU and I-Cache. Once the Fetch Unit receives executable instructions, the instructions are queued in the Instruction Queue to provide some buffering between the shaded components in Fig. 10 and the Fetch Unit. The Dispatch Unit monitors the shaded units in Fig. 10 and ensures that these units are busy executing instructions from the Instruction Queue and not lying dormant for too long. If more useful work can be done by a newly-dormant unit, the Dispatch Unit sends it another instruction to process. Therefore, these units are also busy making copies of the software code.

In summary, the BIU, D-Cache, DMMU, Load Queue, and Store Queue provide data to the shaded units in Fig. 10. The BIU, I-Cache, Fetch Unit, Instruction Queue, and Dispatch Unit provide executable computer instructions to the shaded components in Fig. 10. Each unit makes one or more type of electronic copies as they operate.[160]

In Fig. 10, two banks of registers are illustrated. The General Purpose Register File contains registers[161] for storing data as are temporarily needed for the shaded components in Fig. 10. The Floating Point Register File contains floating point data used by the Floating Point Unit.

Fig. 10 shows six units which are used to execute computer instructions. Many of the shaded components in Fig. 10 are capable of operating parallel in time so that more than one instruction can be processed at any one overlapping time period. Computer code from memory is fetched and executed sequentially through memory addresses unless a branch instruction is executed, sending the program execution to a newm non-sequential location.[162] The Branch Proc-

[160] *See* discussion *supra* in Part II.

[161] Registers are mini-RAM elements that store 8-bit, 16-bit, 32-bit, 64-bit or other small portions of software or data.

[162] Computer instructions stored in addresses 1 through 10 in memory are typically executed sequentially from 1 to 10 unless some computer instruction tells the fetching and executing operation to go to another address/location of memory.

essing Unit predicts and functions to execute these branch instructions which change the flow and location of instructions fetched from memory by the Fetch Unit. Fig. 10 illustrates three Integer Units which handle integer operations.[163] Fig. 10 illustrates a Floating Point Unit which processes floating point instructions[164] which manipulate floating point numbers.[165] A Load and Store Unit in Fig. 10 initiate read and write operations to external memory through the D-Cache.

Fig. 10 illustrates a microprocessor which can be used to transfer data on the Internet or execute code once the hacked code has been fully downloaded from the Internet. When the processor operates to route or move software as data on the Internet, the data paths within the microprocessor will make copies of the hacked software. However, code being executed though the instruction path of the microprocessor will not be infringing code but will simply be the code needed to route and process the hacked software on the Internet. The microprocessor's goal when operating in this routing mode is to receive the hacked program as data from an incoming/input source (e.g., a modem, FDDI input, an Ethernet link, etc.) and move the data to the computer's output line to transmit the data towards its final destination while ensuring the transmitted data's integrity.

In essence, when the microprocessor is functioning as an Internet router, the hacked program traverses the following path when received as microprocessor input in order to determine the proper output destination, proper routing, and error information: (1) into the BIU; (2) into the D-Cache; (3) into the Load Queue; (4) through the Load and Store Unit; (5) to the General Purpose Registers; or (6) to

[163] Integers are number that contain no values right of a decimal point. For example, 12, 350, and -24 are integers, whereas 3.14159 is not an integer but a floating point number. Integers are represented in computers in a simple manner and can be processed easily by special purpose hardware Integer Units. Integer operations include: shift right, shift left, add, subtract, XOR, AND, OR, multiply, and divide.

[164] Floating point instructions include: add, subtract, divide, multiply, sine, cosine, and arc-tangent.

[165] Floating point numbers are represented in a computer by a more complex format than integers. A floating point number is stored as a mantissa, a sign, and an exponent. Therefore, the number -1.123×10^{5} is stored in three segmented values where the mantissa is 1.123, the sign bit is negative, and the exponent is 5. Due to the three fields of a floating point value, floating point operations are harder to process than integer operations and require their own special hardware.

the Integer Units for at least one copy and interactive processing step among all the previously mentioned resources. On the way back out of the microprocessor to the output of the computer, the microprocessor creates roughly the same six copies in reverse. Therefore, even ignoring the repetitive copies made in the processing phase of the operation, roughly twelve copies are made in order to route the data though the microprocessor when performing Internet routing of the Internet datagrams.

E. Unauthorized Software Copies Created on the Internet

Compounding these (*twelve*) copies with the 100 router computers encountered by the hacked software, while traveling from New York to California via the Internet, results in a total of 1200 copies. Adding this number to the current number of copies from memory within each of the 2,000 copies made in memory of a microprocessor, we have 3,200 illicit copies created from New York to California (100 routing computers multiplied by four copies in the computer, multiplied by five copies in the computer main memory, plus 100 x twelve copies made by the microprocessors while routing).

In addition to these copies, many microcontrollers perform data movement operation within the computers other than the main microprocessor. Assume that these microcontrollers are ¼ as complex as the main microprocessor and that each one of these devices makes roughly three copies.[166] Many microcontrollers may be encountered by the hacked software in one computer. To be conservative, assume only two microcontrollers or special purpose peripheral CPUs are encountered on each of the 100 router computers - one for incoming datagrams and one for outgoing datagrams. This adds 600 more copies for a total of 3,800 potential copies from New York to California.

The above 3,800 hacked copies made across the globe assumes that the only microprocessor operating or transferring the data within the computer was the central microprocessor and two microcontrollers in at least one peripheral device. This is usually not the reality

[166] *See* HC11 MicroController Databooks available from Motorola, Inc.

of computer systems. Every function in the computer, memory access, printer storage, serial interface buffering, modem storage and control, Ethernet card storage and control, are all done by microcontrollers which are special purpose microprocessors that make electronic copies, but usually not to the extent of the main microprocessor. Thus, the 3,800 potential illicit copies are an overly cautious estimate.

Assuming that one thousand users now access and download the hacked code from the site where the hacked code was posted, then 3,800x1,000 copies have been made resulting in the creation of 3,800,000 infringing copies, worldwide. If several hackers operate in parallel with each other, hacking many different types of software on the Internet, then it is easy to see how millions or even billions of infringing copies can be made on the Internet in just one day. Such mass infringement may involve the Internet-enabling entities who may be named in the lawsuit under direct or contributory infringement theories.

Now that the vast extent of these electronic copies are understood, it is important to determine if these electronic copies will create patent infringement liability for Internet-enabling entities or if patent law will provide some type of protection.

IV. SOFTWARE PATENT DIRECT INFRINGEMENT ARGUMENTS MAY BE PUT FORTH IN AN ATTEMPT TO OBTAIN A REMEDY

There are no software patent cases directly on point which held that Internet-enabling entities will either be liable as direct infringers or sheltered from direct infringement liability in a patent law context. However, some copyright precedents indicate that direct infringement of patent claims may succeed in certain patent infringement circumstances.[167] One copyright infringement example is demonstrated in *Sega Enterprises, Ltd. v. MAPHIA*.[168]

[167] Generally, while copyright law and patent law differ in substantial ways, copyright law and patent law are both intellectual property legal doctrines that are subjected to the same policy discussions and policy implications. Thus, the underlying and related policies addressed in copyright law may be applicable to patent law cases, especially those cases of first impression where copyright policy is discussed. *See* Sony Corp. of Amer. v. Universal City Studios, 464 U.S. 417, 439 (1984) (explaining that it is appropriate to refer to patent case law in copyright cases because "of the historic kinship between patent law and copyright law);

In *Sega*, the defendant ran a computer bulletin board service[169] ("BBS") which allowed users to upload and download illegal copies of software programs. Many of the software programs uploaded and downloaded to the defendant's bulletin board via bulletin board users were infringements on *Sega*'s copyrights. The court in *Sega* stated, "Sega has established a likelihood of success on the merits of showing a prima facie case of direct and contributory infringement by Defendants' operation of the MAPHIA bulletin board."[170] Although the Court in *Sega* seemed to focus on the culpability, knowledge, intent, and level of inducement by the defendant,[171] the mere fact that the Court still found direct infringement, which does not require intent, knowledge, inducement, etc., suggests a willingness by courts to find direct infringement by Internet service providers. Although the defendant in *Sega* acted with volition and was clearly culpable, the holding indicates that direct infringement liability may be placed on the Internet-enabling entities by passively providing the means by which directly infringing Internet copies are created.[172]

Another copyright case that may affect the imputing of direct infringement liability is *Frena*.[173] In *Frena*, the court found direct

Atari Games Corp. v. Nintendo of Amer. Inc., 24 U.S.P.Q.2d (BNA) 1015 (Fed. Cir. 1992); Data Gen. Corp. v. Grumman Sys. Support Corp., 32 U.S.P.Q.2d (BNA) 1385 (1st Cir. 1994); Pro-CD Inc. v. Zeidenberg, 39 U.S.P.Q.2d (BNA) 1161, 1166 (7th Cir. 1996); Image Tech. Servs. v. Eastman Kodak Co., 44 U.S.P.Q.2d (BNA) 1065 (9th Cir. 1997). For a recent example where a software copyright claim and patent software claim were filed together and considered in unison, see Cabinet Vision v. Cabinetware, 44 U.S.P.Q.2d 1683, 1684 (Fed. Cir. 1997).

[168] *Sega*, 857 F. Supp. 679 (N.D. Cal. 1994) (finding direct and contributory copyright infringement by computer bulletin board service company and individual in control of bulletin board who knew and facilitated the creation of unauthorized copies of video games when such illegal copies were uploaded to the bulletin board by unknown users and subsequently downloaded by users to make additional copies).

[169] A "computer bulletin board system" which is an older term for a modem or Internet accessible computer server.

[170] *Id.* at 687.

[171] *Id.*

[172] *See also* Columbia Pictures Indus., Inc. v. Aveco, Inc., 800 F.2d 59 (3d Cir. 1986) (holding that by renting rooms to the public to view video cassettes obtained from any source, defendant has infringed the owner's exclusive rights to authorize public performances of the copyrighted work). *But see* Sony Corp. v. Universal City Studios, Inc., 464 U.S. 417, 435 (1984) (finding that where public policy and the public good of enabling copying may require that none of direct or contributory infringement be found). Note that the Court in *Sega* also found both contributory infringement and direct infringement of copyright.

[173] Playboy Enters., Inc. v. Frena, 839 F. Supp. 1552 (finding that electronic bulletin board service operator was liable for the transfer of copied, digitized photographs through his bulletin board despite the operator's lack of knowledge of the transfer).

infringement by the defendant and held "[t]here is no dispute that Defendant Frena supplied a product containing unauthorized copies of a copyrighted work. It does not matter that Defendant Frena claims he did not make the copies itself [sic]."[174] The findings in *Frena* suggest that even if an Internet-enabling entity committed no act having any nexus to their user's infringement, a court may still be inclined to find direct infringement. Further, without requiring volition of any kind, the *Frena* court stated:

> There is irrefutable evidence of direct copyright infringement in this case. It does not matter that Defendant Frena may have been unaware of the copyright infringement. Intent to infringe is not needed to find copyright infringement. Intent or knowledge is not an element of infringement, and thus even an innocent infringer is liable for infringement.[175]

While patent law and copyright law differ in some significant manners, the same Internet public policy will undoubtedly influence both patent and copyright law on the Internet. Therefore, one may study recent copyright case laws and the public policy voiced in these opinions to gain insight as to what direction the patent infringement doctrine on the Internet will develop.[176] Given this language from *Frena*, it seems that a passive Internet-enabling entity may be found liable through patent law even if infringement occurred without the entity's knowledge, intent, committing an act, or having volition with some reasonable nexus to the infringing activity.

It is not the author's contention that *Frena* and/or *Sega* were decided wrongly. In fact, with the apparent culpability and knowledge of the defendants in both of these cases, it seems that a finding of some sort of infringement was necessary to deter this type of wrongful conduct. The author mere notes that the courts deciding *Sega* and *Frena* seemed to go out of their way to find direct infringement where contributory infringement, vicarious liability, or inducement would have served the same purpose. By stretching to find direct infringement, a broad reading/interpretation of either

[174] *Id.* at 1556.

[175] *Id.* at 1559.

[176] *See* discussion *supra* Part I. (or page 105)

Frena or *Sega* by any other federal court could result in very passive and unculpable defendants being held liable under a direct infringement strict liability theory in both a copyright context and a patent context via Internet activity.

Due to this case law, the original software owners harmed by Internet hacking may attempt to use direct infringement to impose liability on Internet-enabling entities involved in the infringing transfer of the patented material, even absent both volition and intent/knowledge. This type of claim is especially likely to occur when the hacker is judgment proof or concealed by anonymous remailing, encryption, and/or like technology manufactured, owned, or operated by the Internet-enabling entity. The software owner can argue that 35 U.S.C. § 271(a) grants the patent owner has the right to exclude others from making, using, selling, offering for sale, or importing the patented software. Though of these five elements, only three -making, using, and importing - may be problematic for an Internet-enabling entity. Additionally, the ability to "export" TCP/IP Internet datagrams to foreign countries may result in liability under 35 U.S.C. § 271(f).

A. The Right to Exclude Others From "Making"

The software patent owner may argue that the memory devices, processors, computers, telecommunication sites, and networks that together comprise the Internet all made infringing copies[177] of the hacked software in order to transfer the hacked software across the U.S. or globe. The integrated circuits (ICs), computers, and memory/storage devices on the Internet were designed, owned, and operated for the purpose of making automatic and infringing copies of the patented software or database. The software patent owner will also argue that the Internet-enabling entities need not have any intent or knowledge and need not even commit the actual act or volition of making the copies as in *Sega* and/or *Frena* to be found strictly liable as direct infringers.[178]

[177] *See* discussion *supra* Part III.

[178] *See* Kewanee Oil Co. v. Bicron Corp., 416 U.S. 470 (1974) (holding that patent law forbids independent creation of the patented matter and a finding of infringement needs no intent or knowledge); Thurber Corp. v. Fairchild Motor Corp., 269 F.2d 841, 845 (5th Cir. 1959) (finding that infringement does not depend on the infringer's good faith or innocent

The Internet-enabling entities will argue that the article of manufacture claims, the Alappat means-plus-function claims, and the data structure claims require two main elements in order to be directly infringed. The first element required is the storage medium, machine, computer, memory/storage, and/or a like hardware element required under 35 U.S.C. § 101 in order to create an enforceable software claim.[179] The other required element in order to directly infringe the claim is the software resident within the hardware element.

The Internet-enabling entities will argue that they only provided the hardware device, and that the hacker provided the software whereby the only viable theory under which the providers can be held liable is under a contributory theory for contributing and controlling the providing of only a portion of the infringing entity. This situation of "making" of the infringing copies seems best suited for the theory of contributory infringement which was developed to handle the infringer that provided a smaller fraction of a total patented combination. It seems inequitable, given the large damages and ease of creating large damages on the Internet, that an Internet-enabling entity that had no knowledge/intent to infringe and had no human volition or act associated with the infringing act, other than enabling automatic mechanical reproduction or manufacturing, selling, and maintaining Internet equipment, should be found directly liable. Some act of infringement with some non-mechanical nexus

mind-set); Metal Film Co. v. Melton Corp., 316 F. Supp. 96, 111 (S.D.N.Y. 1970) (holding that neither lack of knowledge of the patent nor lack of intent to infringe is a defense for the issue of patent infringement); Playboy Enters, Inc. v. Frena, 839 F. Supp. 1552, 1555 (M.D.Fla. 1993); Sega Enters., Ltd. v. MAPHIA, 857 F. Supp. 679 (N.D. Cal. 1994).

[179] Decades of federal case law have engrained hardware limitations into software claims for the sole purpose of allowing software claims to surpass the patentable subject matter hurdle of 35 U.S.C. § 101. The addition in the claim of well known hardware, structure, memory, storage, computer components, etc. does little to overcome art rejections under 35 U.S.C. § § 102-103. Therefore, it seems ironic that the hardware limitations in a software claim, which were added or included to the claim to satisfy decades of engrained 35 U.S.C. § 101 formalities, may now incur liability to the hardware manufacturers. One should ask, would it be best to dispense with the form-over-function hardware limitations in the claim and allow the claiming of the software absent a "tangible" medium to avoid imposing contributory liability on hardware manufacturers? See Gottschalk v. Benson, 409 U.S. 63 (1972); Parker v. Flook, 437 U.S. 584 (1978); Diamond v. Diehr. 450 U.S. 175 (1981); *In re* Freeman, 573 F.2d 1237 (C.C.P.A. 1978); *In re* Walter, 618 F.2d 758 (C.C.P.A. 1980); *In re* Abele, 684 F.2d 902 (C.C.P.A. 1982), for historical law on the development of hardware limitations in software claims.

to the infringing activity should be required in order to directly infringe.[180]

Therefore, finding direct infringement by Internet-enabling entities under the theory of excluding the automatic "making" of infringing copies would seem to be misapplication of the concept of direct infringement of patent claims, as was noticed in *Netcom* in an Internet copyright context.[181] A more rational approach would be to utilize a theory of contributory infringement if the right to exclude the "making" of copies was the right relied upon by the original software owner. Once you consider the fact that it is desirable to: (1) avoid or reduce the high quantity of damage and infringing activity that can occur on the information superhighway in a very brief period of time without any intent or knowledge on the part of the Internet-enabling entities; (2) maintain the cost of Internet access within a feasible range for further U.S. economic and technological growth; and (3) avoid any chilling effect on new products and future innovations related to the information superhighway, then it is very likely that any theory of direct infringement under the "making" element of 35 U.S.C. § 271(a) should fail for passive nonvolitional Internet-enabling entities.

B. The Right to Exclude Others From "Using"

In addition to claiming that the Internet-enabling entities infringed due to the "making" of infringing copies, the original soft-

[180] Eastman Oil Well Survey Co. v. Sperry-Sun Well Surveying Co., 131 F.2d 884, 887 (5th Cir. 1943) (holding that liability is based upon "what the defendant is doing" under direct infringement); Orthokinetics, Inc. v. Safety Travel Chairs, Inc., 806 F.2d 1565, 1578 (Fed. Cir. 1986) (finding defendants personally liable for *acts of direct infringement*) (emphasis added). "Act" may also include omission or failure to act after one is given knowledge of infringement. Suppose one told an Internet entity that infringing software was on his service and he chose to do nothing for two years? *See* Black & Decker (U.S.), Inc. v. Home Prod. Mktg., Inc., 929 F. Supp. 1114, 1121 (N.D. Ill. 1996) (concluding that either acts or omissions accompanied by knowledge can result in direct infringement). It seems that although strict liability does not require intent or knowledge of infringement, it does require that the defendant do some conscious act or volition with some nexus to the infringement. It seems suspect that a defendant who had no knowledge and performed only an automatic act that required no volition would be liable for massive damage on the Internet, especially when the Internet provides substantial public benefit over and above infringing activities.

[181] *See infra* Part IV. B. for a more detailed discussion of *Netcom*, 907 F. Supp. at 1368 (finding no direct infringement for copyright law on the Internet when no volitional act is found on the part of defendant; automatic copies should not result in direct infringement liability).

ware owner may bring suit relying on the right to exclude others from "using" of the infringing work. The original software owner may argue that it is immaterial as to how the work was made, and that after the work was made on the hardware via the Internet, the Internet-enabling entities used the copy to facilitate Internet transfer. The original software owner will argue that 35 U.S.C. § 271(a) uses the disjunctive form of "or" that implies that one is liable for infringement by "using" even if sale, offer for sale, importation, or making are absent.[182] The original software owner will state that "use" is broadly defined, and arguably, need not be for infringing software execution, but need only be used in Internet transferring in order to constitute infringement. Sometimes, mere storage or backup with the ability to use, even absent any real use, may arguably constitute "using" under 35 U.S.C. § 271(a).[183] The patent owner will also argue that the "use" need not be the contemplated use of software execution and need only be any use of the software, such as Internet transmission.[184]

The defenses and policy against any finding of direct infringement for both "making" and "using" in this patent infringement hypothetical are best discussed with respect to *Netcom*.[185] Even though *Netcom* is a copyright infringement case, the policy considerations and remarks made therein are very insightful and well-suited to strict liability patent law as it should apply to the Internet.

[182] For the historical creation of this line of thinking, see Beedle v. Bennett, 122 U.S. 71 (1887). *See also* Aro Mfg. Co. v. Convertible Top Replacement Co., 377 U.S. 476, 484 (1964) (holding that unauthorized use, without more, constitutes infringement); Roche Prods., Inc. v. Bolar Pharmaceutical Co., 733 F.2d 858, 861 (Fed. Cir. 1984) ("It is well-established that the use of a patented invention, without either manufacture or sale, is actionable.").

[183] *See* Olsson v. United States, 25 F. Supp. 495 (Ct. Cl. 1938); Hughes Aircraft Co. v. United States, 717 F.2d 1351 (Fed. Cir. 1983) (having a system act as a backup mode or to provide an extra yet unused measure of safety constitutes "use").

[184] *See* American Standard, Inc. v. Pfizer Inc., 722 F. Supp. 86, 105 n.10 (D. Del. 1989) ("one cannot escape infringement by merely taking the claimed structure and using it in a way that differs from the description contained in the preferred embodiment."); E.I. DuPont de Nemours & Co. v. Mallinckrodt, Inc., 833 F.2d 1022 (Fed. Cir. 1987) (entitling patentee to any use of the patented structure even if the use was previously unknown to him). Some case law refers to this theory as the "all uses" theory. *See* General Talking Pictures Corp. v. Western Electric Co., 304 U.S. 175, 181 (1938); Treemond Co. v. Schering Corp., 122 F.2d 702, 703, 706 (3d Cir. 1941).

[185] *Netcom*, 907 F. Supp. 1361 (N.D. Cal. 1995).

In *Netcom*, the court found that the service providers and Internet-enabling corporations enabled a system that operated without human intervention (i.e., there was no volitional act or intent or knowledge by the defendants). Since there was no human intervention on the defendant's part, the defendant in *Netcom* could not be the one who created the copies via an infringing act. Even though intent is not an issue in strict liability, there needs to be an act or some element of volition with causation to the infringing copy which is totally lacking where a defendant's system is merely used to automatically or mechanically create a copy by a third party.

The court in *Netcom* also concluded that the defendant BBS operator correctly distinguished its situation from another case, *MAI Systems Corp. v. Peak Computer Inc.*,[186] which found violation of copyright law when Peak Computer transferred copyrighted software into RAM. The defendant in *Netcom* pointed out that it had not taken any affirmative action directly resulting in any copying of plaintiffs' works, other than installing and maintaining a system whereby software automatically forwards messages received from subscribers onto the Usenet, and temporarily and automatically stores copies on its system. The court further stated that "Netcom's actions, to the extent that they created a copy of plaintiffs' works, were necessary to having a working system for transmitting Usenet postings to and from the Internet." The defendant in *Netcom* did not initiate the copying or act with volition, whereas the defendants in *MAI* infringed the plaintiff's copyright by taking some affirmative action to create the infringing copy.[187] Therefore, the court reasoned in *Netcom* that the system incidentally creating temporary copies of plaintiffs' works does not mean Netcom has caused the copying.[188]

[186] MAI Sys. Corp. v. Peak Computer, Inc., 991 F.2d 511 (9th Cir. 1993).

[187] MAI held copyrights to an operating system (OS) that it sold to third parties. Peak Computer was providing competitive maintenance and support services for the computers and software, including the MAI OS, that were in possession by the third parties. To service their clients' machines and/or the MAI OS, the personnel from Peak loaded and executed MAI's OS from the computer's memory. Peak's employees would actually perform the affirmative act of making the infringing copy of the MAI OS by switching on the client's computer, thereby loading the software into the computer's RAM. The court found that this act of turning on the computer was a sufficiently affirmative and volitional act by Peak to impute liability for direct infringement.

[188] *Netcom*, 907 F. Supp. at 1368-69.

Furthermore, the *Netcom* court stated that "contributory infringement is more appropriate for dealing with BBS liability, first, because it focuses attention on the BBS-users relationship and the way imposing liability on BBS operators may shape this relationship, and second because it better addresses the complexity of the relationship between BBS operators and subscribers."[189] This rationale seems to be equally applicable to patent law when considering whether to use a contributory infringement analysis versus direct infringement analysis.

In addition, the *Netcom* court declared that any holding to the alternative, where direct liability is possible, would result in "unreasonable liability" that could jeopardize the entire Internet or raise transactional costs so high that the Internet would be economically crippled in the U.S.[190] This increased cost and chilling effect on the Internet would result regardless of whether copyright or patent infringement is asserted by the plaintiff.

Another rationale considered by the court in *Netcom* was that "if an entity provided only the wires and conduits — such as the telephone company, it would have a good argument for an exemption if it was truly in the same position as a common carrier and could not control who or what was on its system."[191] The Internet-enabling entities can be likened to common carriers as they are passive entities that merely manufacture, provide, own, or operate the system that enables useful public Internet communication. This benefit to the public should not be hampered by massive liability under direct infringement, especially when no intent/knowledge and no volition are found on the part of the defendant.

The court in *Netcom* cited and distinguished *Sega*. The court in *Sega* found that the defendant's knowledge of the infringing activities, encouragement to infringe, direction and provision of the facilities for infringement through his operation of the BBS constituted direct and contributory infringement, even though the defendant did

[189] *Id.* at 1369.

[190] *Id.*

[191] *Id.* at n.12 (quoting Information Infrastructure Task Force, *Intellectual Property and the National Information Infrastructure: The Report of the Working Group on Intellectual Property Rights* 122 (1995)).

not know exactly when files were on the BBS.[192] It is clear that a very large majority of the Internet-enabling entities provide no encouragement, incentive, or direction promoting any patent infringement over the Internet. In addition, no permanent archives are maintained by a large majority of the Internet-enabling entities.[193]

Netcom found that there would be no purpose served by holding liable those who have no ability to control the information to which their subscribers have access.[194] If you enjoin an Internet-enabling company, how can they stop the infringement when they have no direct control over the use of their provided, manufactured, or owned resources?[195] What infringement deterrent is achieved by assigning massive damage liability to an entity that cannot cost-effectively avoid subsequent infringing acts? The court in *Netcom* recognized that billions of bits of data flow through the Internet every day, these bits are necessarily stored on servers throughout the network, and it is impossible to screen out infringing bits from noninfringing bits while keeping transaction costs reasonable.[196] In addition, under *Netcom*, vicarious liability cannot bootstrap a defendant into direct infringement liability since there is no financial gain to the Internet-enabling companies by virtue of the hacking and infringement.[197]

[192] *Id.* at 1370.

[193] *Id.* at 1372.

[194] *Id.*

[195] The whole point of Internet service providers (ISPs) controlling or monitoring the content of their systems is a mess and a "no-win" situation for them. Libel law suggests that if the ISP controls or monitors the content of its service, it may be found liable for the libel. *See* Stratton Oakmont, Inc. v. Prodigy Servs. Co., 1995 WL 323710 (N.Y. Supp.); Cubby v. Compuserve, 776 F. Supp. 135 (S.D.N.Y. 1991); United States v. Thomas, 74 F.3d 701 (6th Cir. 1996). This may inadvertently encourage an ISP should not monitor the activity on its system to avoid certain liability exposure. However, copyright law on the Internet suggests that strict liability may punish the ISP that does not attempt to look for and remove the infringing content off of its service. *See* Religious Tech. Ctr. v. Netcom, 907 F. Supp. 1361 (N.D. Cal. 1995); Playboy Enters., Inc. v. Frena, 839 F. Supp. 1552 (M.D. Fla. 1993); Sega Enters. Ltd. v. MAPHIA, 857 F. Supp. 679 (N.D. Cal. 1994). Patent law may find that if the ISP controls the content of its service, it may have enough control to come closer to a finding a vicarious liability and may have the needed volition or lack of volition accompanied with knowledge to find direct liability. Therefore, if an ISP monitors content, it is in a liability situation, but if it decides not to monitor content, the ISP is still exposed to liability.

[196] Screening methods may provide futile if the sender uses technologies such as encryption, compression, etc.

[197] *Netcom*, 907 F. Supp. at 1375 (defendant is liable under vicarious liability doctrines for the actions of a primary infringer where the defendant: (1) has the right and ability to control the infringer's acts; and (2) receives a direct financial benefit from the infringement).

In essence, the *Netcom* court was swayed by the fact that: (1) the defendant took no affirmative action or committed no volitional act to facilitate the creation of the copy; (2) no human intervention was needed since the copies were automatic; (3) the copies are needed in order to maintain a functional Internet that is very beneficial to the public; (4) the liability imposed on Internet-enabling entities would be massive, repetitive, and crippling to the Internet industry; (5) contributory infringement is more appropriate for providers since it requires knowledge and some contributory volition; (6) passive providers of a communication conduit and system should not be liable in a manner similar to the fact that a phone company is not liable for slander; (7) Internet-enabling companies do not provide encouragement, incentive, or direction promoting any copyright infringement and therefore inducement theories provide some protection; (8) very few, if any, entities on the Internet keep a permanent archive of the infringing file which could be problematic upon marking or notice to the archiver followed by inaction to remove the infringing copies; (9) an injunction or court order cannot be effective since the Internet-enabling entities have no direct control over the system or its use; (10) it is impossible to screen infringing work from non-infringing work even if a close and vigilant watch is maintained at great cost; and (11) vicarious liability does not apply for many Internet-enabling entities since there is no financial gain to most of the Internet-enabling entities from the infringement and there usually is an extreme lack of ability to control the use of their resources. These arguments seem equally persuasive and insightful in patent law, as well as for copyright law, as these insights pertain to the Internet.

Therefore, it seems reasonable that direct patent infringement by virtue of automatic "making" and/or "using" should not be found against mere passive Internet-enabling entities that have committed no volitional acts having a direct nexus to the infringement. By not allowing direct infringement claims to succeed against passive Internet-enabling entities in a manner similar to *Netcom* and *Sony*, theories of inducement, contributory infringement, direct infringement involving volitional acts, vicarious liability, and the like still provide ample disincentive to knowing and volitional Internet infringement and will result in imposing liability on all culpable Internet infringers as clearly seen in *Sega* and *Frena*.[198] In general, inducement, contributory infringement, direct infringement involving volitional acts, vicarious liability, and the like will protect passive Internet-enabling entities while finding culpable Internet-enabling entities liable in a well-balanced Internet liability system.

C. Importing TCP/IP Datagrams on Internet Lines

When transmitting Internet datagrams using the TCP/IP connectionless protocol, Internet datagrams may be imported into the U.S. and exported out of the U.S. to effectuate global Internet communications. The importation and exportation of these datagrams and the liability that may result under patent law is the subject of the next two sections.

Liability for importation of the infringing piece of software or data structure is provided by 35 U.S.C. § 271(a)[199] and/or 35 U.S.C. § 271(g).[200] Assume that a hacker copies an infringing program from Germany to the United States. Due to the connectionless nature of the TCP/IP Internet protocol, different parts of infringing software or database can come into the country through different states and finally create a 100% slavish total copy in one or more U.S. states. Under 35 U.S.C. § 271(a), the importation analysis is similar to the "using" and "making" direct infringement analysis discussed, *supra*, in Part II since in order to import an object from the Internet, a copy must be "made" or "used" before, during, and

[198] *See* discussion *supra* Part IV-IV. A.

[199] 35 U.S.C. § 271(a) provides: Except as otherwise provided in this title, whoever without authority makes, uses, offers to sell, or sells, any patented invention, within the United States or imports into the United States any patented invention during the term of the patent therefor, infringes the patent. 35 U.S.C. § 271(a) (1988).

[200] 35 U.S.C. § 271(g) provides: Whoever without authority imports into the United States or offers to sell, sells, or uses within 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, offer to sell, sale, or use of the product occurs during the term of such process patent. In an action for infringement of a process patent, no remedy may be granted for infringement on account of the noncommercial use or retail sale of a product unless there is no adequate remedy under this title for infringement on account of the importation or other use, offer to sell, or sale of that product. A product which is made by a patented process will, for purposes of this title, not be considered to be so made after —

> (1) it is materially changed by subsequent processes; or (2) it becomes a trivial and nonessential component of another product.

35 U.S.C. § 271(g) (1988).

after the importation occurs. Thus, passive entities having no volition or nexus to the infringement should not be liable for this infringing importation, similarly to that examined in the *Netcom* decision.

Regarding 35 U.S.C. § 271(g) some hurdles are placed into the statutory language that should give some protection to Internet-enabling entities. Under 35 U.S.C. § 271(g), infringement may be avoided if the product being imported into the United States was either (1) materially changed by a foreign process; or (2) "becomes a trivial and nonessential component of another product." Given the technological working of the Internet, including data packeting, compression, encryption, telecommunication protocol alteration, etc., one may attempt to argue that the imported product was materially changed by one or more foreign processes before importation.[201] Therefore, the exception under § 271(g)(1) for imported objects which have been materially changed may give some protection to Internet-enabling entities involved in passive importation.

While § 271(g)(1) offers some protection, it seems unlikely that § 271(g)(2) will offer any protection. It is improbable that a necessary fraction of the code imported into the United States via a TCP/IP datagram which is required for program proper execution and transmitted for the express purpose of being put together with other TCP/IP datagrams coming over the Internet could be called trivial, or nonessential, to the software program. Consequently, the § 271(g)(2) "nonessential component" exception will not include any TCP/IP datagram which enters the United States.

In any event, it is likely that importation and exportation of TCP/IP Internet packets over the Internet could create liability problems for Internet-enabling entities, even if passively involved, due to the lack of a volitional act and the courts' willingness to cast a broad net for direct infringement, as found in both *Sega* and *Frena.*

[201] *See* Eli Lilly and Co. v. American Cyanamid Co., 82 F.3d 1568 (Fed. Cir. 1996) (finding that if a patentee has a patent on X and X is "materially changed by subsequent processes" before it is imported, then no infringement will be found); Bio-Technology Gen. Corp. v. Genentech, Inc., 80 F.3d 1553, 1559 (Fed. Cir. 1996) (noting that the "materially changed" exception of 35 U.S.C. § 271(g) requires that "at a minimum, that there be a real difference between the product imported, offered for sale, sold, or used in the United States and the products produced by the patented process"), *cert. denied,* 117 S. Ct. 274 (1996).

D. Exporting TCP/IP Datagrams on Internet Lines

Various Internet copies, such as the time serial copy or the 100% slavish total copy may result in direct infringement due to exportation of TCP/IP datagrams out of the U.S. on Internet communication lines. It was unclear throughout patent law history whether time serial copies themselves constitute infringement under either direct or contributory infringement. This issue was first faced by the Supreme Court in *Deepsouth.*[202]

In *Deepsouth,* the defendant manufactured all the parts of a claimed invention. The defendant then individually boxed each of the parts and shipped them to a foreign location where a completed product containing all of the parts was put together. This situation is similar to the time serial memory device since a time serial memory device manufactures or copies all of the parts of the program one after another in a same location but never puts them together to form the entire program during one overlapping point in time.[203] Assembling the entire program after Internet transmission completes is left to another device at the receiving end (analogous to the foreign country in *Deepsouth*).

The Fifth Circuit found infringement in the case of *Deepsouth;*[204] however, the Supreme Court later reversed the FifthCircuit's decision, holding that no infringement occurs when the parts of the invention are created while the total combination of the parts into a complete unit does not occur within the physical domain of the patent's protection (i.e., the United States and its territories).[205]

[202] Deepsouth Packing Co. v. Liatram, 406 U.S. 518 (1972) (holding that making pieces of a patented device and then exporting them to be put together into the total claimed device was not infringement under U.S. patent laws).

[203] Even though the defendant in *Deepsouth* did not form the whole claimed unit, the defendant had possession of all the pieces at one time. One distinguishing feature is that the time serial device does not have possession of all the pieces at one time as did the defendant in *Deepsouth*. However, it seems clear that if one infringes a patent by making the N components of the claimed device in one day (N being a finite positive integer), the patent infringer could not avoid liability by making and shipping each of the components to a foreign country in non-overlapping and separate time periods. The final result and damage is the same, and this slight factual distinction should not absolve one of liability where infringement would otherwise be found.

[204] *See* Liatram Corp. v. Deepsouth Packing Co., 443 F.2d 936 (5th Cir. 1971), *rev'd sub nom.* Deepsouth Packing Co. v. Laitram Corp., 406 U.S. 518 (1972).

[205] 35 U.S.C. § 100(c) (1988).

Many federal cases after *Deepsouth* followed the Supreme Court's line of reasoning where a combination patent containing many elements is infringed only where the operational whole of the assembly is created in the United States.[206] The Supreme Court's *Deepsouth* decision left a big hole in the patent law whereby foreign infringement could not be stopped even if the parts were completely manufactured in the United States. However, aware of the loophole and the tension created by *Deepsouth*, the Federal Circuit in *Paper Converting Machine Co. v. Magna-Graphics Corp.*[207] found that the manufacture of the parts of a machine, in a manner similar to time serial copies on the Internet, constituted infringement even though the parts were not completely pieced together into a final total unit. The Court in *Magna-Graphics* expressly stated, "If without fear of liability a competitor can assemble a patented item past the point of testing, . . . [n]othing would prohibit the unscrupulous competitor from aggressively marketing its own product and constructing it to all but the final screws and bolts."[208]

Furthermore, in response to the hole created by *Deepsouth* and the extensive dissent and firm reluctance to comply with *Deepsouth* in the lower courts, Congress passed 35 U.S.C. § 271(f)(1)[209] and 35 U.S.C. § 271(f)(2).[210] Since the passage of 35 U.S.C. § 271(f)(1) and 35 U.S.C. § 271(f)(2), creating the pieces of an invention for presentation to someone else anywhere external to the United States for

[206] *See* Decca Ltd v. United States, 640 F.2d 1156 (Ct. Cl. 1980); Lang v. Pacific Marine and Supply Co., 895 F.2d 761 (Fed. Cir. 1990).

[207] Paper Converting Mach. Co. v. Magna-Graphics Corp., 745 F.2d 11 (Fed. Cir 1984).

[208] *Id.* at 19.

[209] 35 U.S.C. § 271(f)(1) provides: 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 of 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) (1988).

[210] 35 U.S.C. § 271(f)(2) provides: 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 . . . , 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 the patent if such combination occurred within the United States, shall be liable as an infringer. 35 U.S.C. § 271(f)(1) (1988).

assembly into the final device is infringement in the United States.[211] It seems highly possible that an additional application of 35 U.S.C. § 271(f) will be to attempt to make the creation of time serial copies of patented invention infringement when exportation to another country occurs due to the Internet transmission. If one is transferring data from Japan to Germany on the Internet, it is possible that only a fraction, say 70%, of the program will pass through the United States due to the connectionless manner in which the TCP/IP protocol operates.[212] Even in this case, if the 70% was a "substantial portion" of the patented invention, liability may be found under 35 U.S.C. § 271(f)(1). Even worse, a single TCP/IP datagram may be viewed as being a "component" which is "specially adapted" to be pieced together with the rest of the program communicated elsewhere over the Internet. Therefore, even if one TCP/IP datagram crosses the United States, liability could be found under 35 U.S.C. § 271(f)(2) since the one datagram may be "any component . . . that is especially made or . . . adapted for use in the invention." This may also mean any state in which a time serial copy is made will subject a defendant to subject matter jurisdiction in that state. Even if the time serial copy is not direct infringement via the above discussion, direct infringement by virtue of an inevitable final slavish total copy will open the door for contributory infringement involving any time serial copy on the Internet.[213]

V. Vicarious Liability and Inducement

Theories of vicarious liability and inducement can be used to find culpable and/or volitional Internet-enabling entities liable while offering justified protection to passive Internet-enabling entities. If

[211] Wahpeton Canvas Co. v. Bremer, 893 F. Supp. 863, 872 (N.D. Iowa 1995) (finding that *Deepsouth* arguments are ineffective today on summary judgment due to 35 U.S.C. § 271(f)). *See also* ROBERT PATRICK MERGES, PATENT LAW AND POLICY 740 (1992) ("This section [§ 271(f)] overruled *Deepsouth Packing Co. v. Laitram Corp.*, which had held that exporting components of a patented combination for quick assembly overseas was not infringement.").

[212] *See* discussion *supra* Part III.A. for a more detailed discussion of the connectionless TCP/IP protocol.

[213] Deepsouth Packing Co. v. Liatram Corp., 406 U.S. 518, 526 (1972) (holding that if the defendant's conduct was intended to lead to the unauthorized use of the patented device inside the United States, its production and sales activity would be subject of injunction as an induced or contributory infringement).

the Internet enabling entity induced the infringement or exercised control over the user and obtained financial benefit as a direct result of the infringing activity, then a finding of direct infringement liability would be justified. In contrast, it is unreasonable to expose passive Internet enabling entities to direct infringement liability as suggested by *Sega* and *Frena*. Therefore, the use of vicarious liability and inducement may obviate the need for any finding of direct infringement due to Internet transmissions on the part of Internet-enabling entities, while still deterring infringement and ensuring that plaintiffs obtain a proper remedy.[214]

The theory of inducement is not likely to cause too much controversy on the Internet. If a defendant solicits for infringing material, advertises his services knowingly for infringing activity, or instructs an individual to infringe, few would take issue with a finding of liability.[215] However, vicarious liability may pose a few problems. Many Internet service providers exercise control over the use and content of their service. These same Internet service providers may charge service fees on a per access basis, a per download basis, a per upload basis, and/or a time of use basis. When using one or more of these payment forms, the argument can be made that defendant service provider has benefited directly from the infringing activity. On the other hand, one may argue, as was discussed above for direct infringement via *Netcom*, that total 100% policing and control over the Internet is not technically feasible or cost effective and that the level of control for Internet service providers should not be enough to expose service providers to liability.

Thus, while few may object with imposing liability on Internet-enabling entities who actively induce infringement, some may object to the use of vicarious liability as a dominant liability theory for the

[214] *See* Sony Corp. v. Universal City Studios, 104 S. Ct. at 787 (1984) (noting the "historic kinship between patent law and copyright law" in using patent standards for vicarious liability in copyright case); Shapiro, Bernstein & Co. v. H. L. Green Co., 316 F.2d 304 (2 Cir. 1963) (even in the absence of an employer-employee relationship one may be vicariously liable if he has the right and ability to supervise the infringing activity and also has a direct financial interest in such activities)

[215] 35 U.S.C. § 271(b). *See also* Water Techs. Corp. v. Gartner, 850 F.2d 660, 668 (Fed. Cir. 1988) ("[A] person infringes by actively and knowingly aiding and abetting another's direct infringement. Although section 271(b) does not use the word "knowing," the case law and legislative history uniformly assert such a requirement.").

transmission of hacked software over the Internet. Vicarious liability may hold an Internet-enabling entity liable for patent infringement by virtue of the payment structure used that entity. Imposing liability under this situation may cause such Internet-enabling entities to change the payment relationship with their users in an attempt to avoid vicarious liability without actually deterring the infringing activity. Likewise, an Internet-enabling entity may be found liable for a user's infringing activity simply by the manner in which that entity controls user access to system resources. This scenario also creates a less than perfect result since Internet-enabling entities would then purposely take less active roles in policing their systems to avoid vicarious liability. Thus, vicarious liability, while offering a slightly better Internet liability structure than a broadly interpreted, direct infringement theory, may adversely affect growth on the Internet.

VI. SETTLED CONTRIBUTORY INFRINGEMENT LAW PROVIDES PATENT INFRINGEMENT PROTECTION TO NONVOLITIONAL INTERNET-ENABLING ENTITIES

In addition to the relatively reasonable finding of infringement, or noninfringement, under the theories of vicarious liability, inducement, and direct infringement with volitional acts, it is reasonable that contributory infringement should also be used to find culpable Internet-enabling entities liable for patent infringement while sheltering passive Internet-enabling entities from liability. This section shows that well-settled contributory infringement law provides ample protection to the nonvolitional Internet-enabling entity so that massive patent infringement liability will most likely never result for the nonvolitional defendant. If a passive Internet-enabling entity can avoid a finding of direct infringement and force a plaintiff to prove vicarious liability, inducement, and/or contributory infringement, the passive Internet-enabling defendant is in a much more secure position whereas the volitional Internet-enabling entity is likely to be found liable in either circumstance.

A. Requirements of Contributory Infringement

35 U.S.C. § 271(c) is the federal statutory authority for patent contributory infringement, and it provides:

> Whoever offers to sell or sells within the United States or imports into the United States a component of a patented machine, manufacture, combination or composition, or a material or apparatus for use in practicing a patented process, 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 not a staple article or commodity of commerce suitable for substantial noninfringing use, shall be liable as a contributory infringer.

Federal case law interpreted § 271(c) as requiring the complaintant to show that: (1) direct infringement has occurred; (2) the defendant has contributed a material part to the combination infringing the patent; (3) knowledge on the part of the defendant that its component was used in the infringement of a patent or especially designed, made, or adapted for infringement of the patent; and (4) the part contributed by the defendant was not "a staple article" or commodity of commerce suitable for substantial noninfringing use. [216] The sections below outline how a nonvolitional Internet-enabling entity can use: (1) the "knowledge" requirement, and (2) the "substantial non-infringing use" or "staple article" requirement of contributory infringement to avoid liability under contributory infringement.

B. Material Component Requirement and Lack of Direct Infringement Are Not Likely to Avoid Internet Liability

The direct infringement requirement offers little shelter since there will most likely be direct infringement somewhere within the United States unless the Internet transmission begins in a foreign country and is destined for a foreign country. If the transmission is from a foreign country to a destination foreign country, then the TCP/IP Internet protocol may not result in one computer or memory

[216] Marsh-McBirney, Inc. v. Jennings, 22 U.S.P.Q.2d (BNA) 1621, 1625 (C.D. Cal. 1991); C.R. Bard, Inc. v. Advanced Cardiovascular Sys., Inc., 911 F.2d 670, 673 (Fed. Cir. 1990); Preemption Devices, Inc. v. Minn. Mining & Mfg. Co., 803 F.2d 1170, 1174 (Fed. Cir. 1986).

structure in the United States containing an entire 100% memory copy of the entire software or data structure that is claimed. Therefore, the direct infringement requirement[217] of a contributory infringement finding is not extensively addressed herein, since it offers little protection to nonvolitional Internet-enabling entities.

In addition, the "material" requirement of contributory infringement will offer little aid since "material" typically is interpreted to be any element "recited in the claim."[218] If either the hardware or software in a software patent claim is provided by the defendant, then the defendant will most probably have contributed a material component to the act of infringement.

In summary, the "direct infringement" requirement and "material" requirement offer little aid to the passive Internet-enabling entity looking to avoid liability for a hacker's activities.

C. Lack of Knowledge or Intent - The First Line of Defense From Liability

Contributory infringement under 35 U.S.C. § 271(c) also requires that the defendant have some knowledge. Based upon the existing case law, it is uncertain whether or not the "knowledge" requirement mandates that the defendant know his component is being used in a particular manner or know that this particular use of his component is infringement of a patent.[219] In either case, many Internet-enabling entities do not have any knowledge whatsoever as to how their components, services, and computers are being used at any given time. Many Internet-enabling entities may not even know that their products or services are being used on the Internet. Many more entities will not know when, how, where, or why the infringing ac-

[217] Carborundum Co. v. Molten Metal Equip. Innovations, Inc., 72 F.3d 872, 876 n.4 (Fed. Cir. 1995) ("Absent direct infringement of patent claims, there can be neither contributory infringement nor inducement of infringement" [under 35 U.S.C. § 271].); Hodosh v. Block Drug Co., 833 F.2d 1575, 1578 n.9 (Fed. Cir. 1987) (direct infringement is a prerequisite for contributory infringement); Met-Coil Sys. Corp. v. Korners Unlimited, Inc., 803 F.2d 684, 687 (Fed. Cir. 1986) ("Absent direct infringement of the patent claims, there can be neither contributory infringement . . . nor inducement of infringement.").

[218] Conner Peripherals, Inc. v. Western Digital Corp., No. C-93-20117, 1993 WL 645932, at *8 (N.D.Cal. 1993).

[219] *See* Freedman v. Friedman, 242 F.2d 364 (4th Cir. 1957); Buxton, Inc. v. Julen, Inc., 223 F. Supp. 697 (S.D.N.Y. 1963); *Aro Mfg.*, 377 U.S. at 482-83.

tivity occurred. Certainly, few if any of the Internet-enabling companies will know of a specific hacking incident either during its occurrence or before it occurs in order to have the requisite knowledge.

The total lack of any knowledge on the part of Internet hardware manufacturers, Internet service providers, Internet resource operators, Internet software vendors, and telecommunication corporations ought to result in no liability under theories of contributory infringement. However, an Internet-enabling entity that is volitionally involved in the copying and solicitation of illegal copies will most likely have the requisite intent and be found liable under a contributory theory. Therefore, the intent requirement of contributory infringement, like elements of vicarious liability and inducement, is better suited than a broad interpretation of direct infringement for properly determining liability on the part of the volitional Internet-enabling entities while protecting passive Internet-enabling entities from unjustified liability.

D. Staple Good, Non-Infringing Use, and Lack of Special Making or Adaptation - The Second Line of Defense

In addition to lack of knowledge or intent, the defendant may argue that the memory devices, protocols, software, and CPUs on the Internet are either staple goods of commerce or have substantial non-infringing use.[220] One way to prove a defendant's contribution is a staple contribution is to demonstrate that the use of the good is incidental or non-critical to the infringement of the claims.[221] This ar-

[220] Oak Indus., Inc. v. Zenith Elecs. Corp., 726 F. Supp. 1525, 1538-39 (N.D. Ill. 1989) ("If the practice of the patented method is incidental and necessary to the practice of the unpatented methods, the device is a staple and there can be no contributory infringement. If . . . the practice of the patented method is not necessary or incidental to practice of the unpatented methods, a jury could find that the device *as a whole* is not staple and the seller could be liable for contributory infringement.").

[221] For case law on "staple" goods, *see Oak Indus.*, 726 F. Supp. 1525 (N.D. Ill. 1989); Sony Corp. of Amer. v. Universal City Studios, 464 U.S. 417, 426 (1984) (listing common, consumable office supplies as staple goods); Oxy Metal Indus. Corp. v. Quin-Tec, Inc., 216 U.S.P.Q. (BNA) 318, 324 (E.D. Mich. 1982). Typically, only consumables or inexpensive material that have wide application are found to be staple. *See* Eversharp, Inc. v. Philip Morris, Inc., 256 F. Supp. 778, 786 (E.D. Va. 1966) (holding that razor blades in shaving apparatus are staple products); Haskell v. Lever Bros. Co., 243 F. Supp. 601, 614 (S.D.N.Y 1965) (determining that soap is a staple product); Hodosh v. Block Drug Co., 833 F.2d 1575, 1578 (Fed. Cir. 1987) (holding that if defendants' goods are staple articles, the Court must determine whether the allegedly infringing device constitutes a material part of the invention); Haworth,

gument is not actually a decent defense since memory, microprocessors, and like technology are crucial to the operation of the Internet. The Internet will not function without these components. Alternatively, a defendant may argue his components are consumable. Staple goods are oftentimes viewed as being consumable goods having a low cost and short lifetime.[222] Many memory ICs and CPUs are not repaired after use, but instead are thrown out and replaced once they cease to function. Furthermore, some memory devices and microcontrollers do cost less than one dollar. Therefore, some CPUs and memory devices on the Internet may be very similar to consumable products while other memory and CPU devices may not be easy to classify as a consumable good. However, it is improbable, though, that all memory and CPUs on the Internet are consumable since there lifetime is typically many years in length and their cost is a large portion of most computer systems.

Rather, it is highly probable that a typical U.S. home contains tens or hundreds of CPUs and memory devices. Automobiles alone may contain tens of processors and many memory devices. Further increasing this household total are almost all home appliances, some children's toys, stereo systems, TVs, computers, furnaces, air conditioners, home security systems, VCRs, video cameras, remote controls, lighting systems, and so on. As memory and CPUs become more of an integral part of our society, it is even more likely that these items will be considered staple or commonplace by virtue of lower cost, substantial non-infringing use, consumable nature of replacement, high degree of alternative devices, and so forth. Therefore, an argument along the lines of "staple good" or lack of "special making or adaptation" may aid in avoiding liability under patent theories of contributory infringement in addition to the lack of knowledge/intent.

Inc. v. Herman Miller, Inc., 37 U.S.P.Q.2d (BNA) 1080, 1088 (W.D. Mich. 1994) (assessing whether a product is a staple article of commerce by the quality, quantity and efficiency of the suggested alternate uses).

[222] *See*, for example, *Sony Corp.*, 464 U.S. at 426 (1984) (listing common, consumable office supplies as staple goods).

The courts are more inclined to find no liability if the contributed component is used in many other non-infringing manners.[223] Very few memory devices and microprocessors are designed especially for use on the Internet. Many memory devices and processors are used for computers not connected to the Internet, for automotive applications, for medical devices, for consumer electronics such as stereos, cellular phones, microwave ovens, etc., for children's toys, for industrial manufacturing, and the list goes on and on. Even if a processor or memory was specifically designed only for use on the Internet, the Internet is not used only for infringing activity. Even a CPU or memory device specifically designed for Internet will be used for Internet transmissions that are noninfringing. For every hacked piece of software stolen or illegally used on the Internet, many more legal and noninfringing programs, e-mail messages, and data are transmitted.[224] Furthermore, memory devices and CPUs in the Internet computers could be replaced by nearly an infinite number of other memory devices and CPUs manufactured by other legal entities and still remain fully functional since memory and CPUs are usually not specially adapted for use only on the Internet. These substantial non-infringing uses are very tangible and are not "far-fetched", "illusory", or "theoretical" non-infringing uses.[225]

Therefore, the staple products limitations and substantial non-infringing use criterion of 35 U.S.C. § 271(c) should provide substantial protection from liability for nonvolitional or passive Internet-enabling entities.

VII. Conclusion and Summary

The Internet and modern computer systems are high speed micro-factories which have caused a stir in the intellectual property

[223] *See* Mirafi, Inc. v. Murphy, 14 U.S.P.Q.2d 1337, 1349 (Fed. Cir. 1989) (discussing other substantial, non-infringing uses for invention).

[224] *Cf.* Sony Corp. of Amer. v. Universal City Studios, 464 U.S. 417, 788 ("[A] sale of an article which though adapted to an infringing use is also adapted to other and lawful uses, is not enough to make the seller a contributory infringer. Such a rule would block the wheels of commerce." (quoting Henry v. A.B. Dick Co., 224 U.S. 1, 48 (1912))).

[225] *See* Preemption Devices v. Minnesota Min. & Mfg. Co., 630 F. Supp. 463, 471 (E.D. Pa 1985) (stating that plaintiff has the burden of proving contributing infringer knows that the component part is made especially for an infringing use *and* that the component is not suitable for another substantial, noninfringing use), *vacated in part by* 803 F.2d 1170 (Fed. Cir. 1986).

community. Modern computers and the Internet are capable of making and distributing millions of copies of data and/or software programs to millions of end users in very brief periods of time. Modern technology often allows judgment-proof entities to inflict massive amounts of damages in a short period of time via the mass-transmission of these copies. The technology itself may even hide the human hacker/infringer from legal detection via anonymous remailing, encryption, compression, and/or the like. Furthermore, unlike other creations of copies outside of cyberspace, every infringing transmission on the Internet and every infringing copy of software made on a computer system involves the use of equipment, software, services, etc., of hundreds of commercial entities, many of which are fortune 500 companies. Who is to bear the loss when the human hacker/infringer cannot pay for the damage done in cyberspace? In other words, should the Internet enabling entities[226] bear the risk of the loss or should the data/software owners shoulder the risk of loss for massive Internet infringement?

Copyright law has, in the past few years, been forced to deal with this situation as pictures, art, text, data bases, and like copyrighted work, have become common passengers on the Internet. By liberally applying direct infringement theories articulated in *Sega* and *Frena*, it seems that courts have taken the opportunity to shift the risk of loss for copyright infringement onto the Internet-enabling entities. However, case law, including *Netcom* and *Sony,* suggest the opposite policy, where direct infringement should not apply to Internet-enabling entities. *Netcom* suggests that the lack of intent and volition on the part of an Internet-enabling entity should shield the passive Internet-enabling entity from direct liability.

As the battle rages on in copyright law on the Internet, a new war threatens to emerge. Recent CAFC patent law decisions, such as *Alappat*, *Beauregard*, *Lowry*, *Warmerdam*, and *Trovato*, have resulted in the recent issuance of new types of software patent claims that could be infringed by software or database transmission over the Internet. In addition to the issuance of these new claims, the technological advances in both the size and speed of the Internet have

[226] *See supra* note 19 and accompanying text.

enabled more frequent software communications, data transmissions, and patent-protected commercial transactions over the Internet than ever before. Due to these changes in the law, technology, and market, plaintiffs harmed by wrongful acts committed on the Internet may soon decide to plead patent infringement in addition to or in lieu of copyright infringement in the near future. The lack of Internet patent infringement case law, the uncertainty as to the true course of the law in the patent-similar strict liability copyright arena, and the lack of the troublesome fixation requirement in patent law may be especially attractive to a plaintiff seeking redress for infringing software transmission or data base transmission on the Internet.

The courts have yet to decide whether any Internet-enabling entities will be liable under direct infringement or shielded from direct infringement by a lack of volition. It seems unreasonable that patent law would follow a broad application of the holding in *Sega* and *Frena* and find nonculpable and nonvolitional Internet-enabling entities liable for the potentially massive damage created via a single uncontrolled transmission over the Internet. The theories of contributory infringement, vicarious liability, and inducement along with the policies outlined in *Netcom* and *Sony*, seem best suited to patent infringement liability on the Internet, so that volitional and/or culpable Internet-enabling entities are deterred from infringement or found liable for infringement while nonvolitional or passive Internet-enabling entities are not chilled from further Internet innovation.