

# UIC John Marshall Journal of Information Technology & Privacy Law

---

Volume 17  
Issue 1 *Journal of Computer & Information Law*  
- Fall 1998

Article 2

---

Fall 1998

## Patenting Computer Science: Are Computer Instruction Writings Patentable?, 17 J. Marshall J. Computer & Info. L. 5 (1998)

Allen B. Wagner

Follow this and additional works at: <https://repository.law.uic.edu/jitpl>



Part of the [Computer Law Commons](#), [Intellectual Property Law Commons](#), [Internet Law Commons](#), [Privacy Law Commons](#), and the [Science and Technology Law Commons](#)

---

### Recommended Citation

Allen B. Wagner, *Patenting Computer Science: Are Computer Instruction Writings Patentable?*, 17 J. Marshall J. Computer & Info. L. 5 (1998)

<https://repository.law.uic.edu/jitpl/vol17/iss1/2>

This Symposium is brought to you for free and open access by UIC Law Open Access Repository. It has been accepted for inclusion in UIC John Marshall Journal of Information Technology & Privacy Law by an authorized administrator of UIC Law Open Access Repository. For more information, please contact [repository@jmls.edu](mailto:repository@jmls.edu).

# PATENTING COMPUTER SCIENCE: ARE COMPUTER INSTRUCTION WRITINGS PATENTABLE?

by ALLEN B. WAGNER†

## I. INTRODUCTION

This paper opposes the IBM/PTO<sup>1</sup> proposal to patent (as an article of manufacture) computer instruction fixed on computer readable media (so called *media* or *Beauregard* claims). The juridical issue raised is whether patents are limited to the utilitarian embodiment of inventions (the instructed machine) or may be extended to include mere symbolic expression (the machine instruction) fixed in a tangible medium.

We argue (a) patenting symbolic expression breaches the intellectual property premise prohibiting property interests in mere abstract ideas, by avoiding both copyright merger and patent preemption doctrines, and (b) contrary to the PTO analysis, patents and copyrights are mutually exclusive statutory interests with no overlap in “abstract expression” subject matter.

On the practical side, we contend *media* claims provide an unjust enrichment and competitive advantage to computer manufacturers over software companies by allowing (a) a second compensation demand for an already licensed use (i.e., two payments for one invention embodiment), and (b) hardware industry dominance over independent software development.

---

† © 1998 Oracle Corp. and Allen B. Wagner, all rights reserved, by Allen Wagner, Associate General Counsel, with the dedicated support and assistance of Special Counsel Katja DeGroot. We express our deep appreciation and commend this writing to the many members and friends at KIPO, JPO, EU, EPO, UKPTO and PTO, who selflessly contributed their time, inspiration and questions to our understanding. The views expressed herein are not necessarily those of Oracle Corp.

1. See *In re Beauregard*, 53 F.3d 1583 (Fed. Cir. 1995) (where IBM appealed the PTO denial of such a claim and the PTO conceded the appeal before a court decision was rendered); and the United States Dept. of Commerce, Patent and Trademark Office, Examination Guidelines for Computer-Related Inventions, 61 Fed. Reg. 7,478 (1996) (adopting IBM’s position following the *Beauregard* concession). While we criticize the PTO adoption of IBM’s proposal, this paper expresses no opinion or criticism on any other portion of the PTO *Guidelines*.

In examining the *media* claim issue, we noticed it was the progeny of a deeper continuing uncertainty over the patentability of computer science ingenuity in general. Seeking to understand the former, we were drawn into considering the latter and offer our observations and commentary on it, which now comprises the major portion of this paper. Specifically, using the Cartesian division of reality into objective and subjective realms, we demonstrate how (a) natural science ingenuity lies in the abstract solution to objective problems, thus having measurable physical impact which is used to determine patentability, while (b) computer science ingenuity lies in the abstract solution to abstract problems, having no causal impact upon the objective realm.

We offer the Cartesian divide perspective to understand the continuing juridical confusion over the application of patent law to computer science advancements. We note the critical issue raised is *physical novelty*, which serves to assure the fundamental premises of our intellectual property jurisprudence (i.e., that there be nothing taken from the public domain and that there be no property interest in mere ideas). We suggest *novel utility* in the objective realm as a reasonable alternative standard for patenting computer science ingenuity, while maintaining the same premises.

As part of our explanation we note *novel utility* would resolve the United States Supreme Court decisions (a) placing mere mechanization of mathematical algorithms in the public domain, and (b) equating natural science principles with mathematical algorithms. The question raised is whether computer science ingenuity is to be given parity to natural science ingenuity in patenting as well as preemption; *novel utility* is viewed as a means to do so.

Part II, *Intellectual Property Fundamentals: Can an Idea be Owned?*, reviews the fundamentals, introduces a Cartesian perspective and concludes there is no property interest over mere abstract ideas or scientific principles; though abstract expressions of ideas are copyrightable and utilitarian embodiment of inventive conceptions are patentable.

Part III, *The Abstract Nature of Computer Science*, explores the abstractions of computer science by reviewing the Turing machine, the theory of algorithms and the difference between digital and analog computation, and by comparing computer science ingenuity (abstract solutions to abstract problems) with natural science ingenuity (abstract solutions to concrete problems).

Part IV, *Computer Science and Intellectual Property Law*, serves a dual purpose: primarily, it describes the judicial application of patent law to computer science; but it also contains our commentary on two issues, (a) is a utility limitation necessary and sufficient to patent mere mechanization of process computation, and (b) are mathematical algo-

arithms analogous to scientific principles in patentability as well as pre-emption? A fundamental question arises in this discussion; that is, will the ingenuity of a computer scientist be given parity with the ingenuity of a natural scientist?

Part V, *Are Computer Programs on Media Patentable?*, offers our arguments on the issue we initially set out to discuss.

Finally, Part VI, *Concluding Comments*, offers our closing remarks.

## II. INTELLECTUAL PROPERTY FUNDAMENTALS: CAN AN IDEA BE OWNED?

The institution of property has interested social philosophers in part, at least, because it raises issues of justice. . . . [B]ecause it discriminates between rights and fortune, it invites moral criticism and the demand for justification.

Many of the classical accounts of the origin and function of private property have taken for granted that in nature all things were held 'in common.' This phrase, however, is ambiguous, for it often meant not a system regulating the use of goods by general agreement but a condition where, there being no rules, everything was *res nullius* (a thing belonging to no one) and the concept 'property' was consequently irrelevant. How, then, it was asked, would men come to appropriate the land and its fruits? How could such appropriation be justified? What would be rational grounds for claiming exclusive possession?<sup>2</sup>

### A. OUR COMMON LAW HERITAGE

Property is the *de jure* right to exclude others from use of its subject matter.<sup>3</sup> Under the common law only tangible stuff was the subject of property; that is, you could exclude others from using your stuff but had no right to interfere with the use of other people's stuff. Thus, the *subject matter* and *possessory interest* of common law property were congruent and there were no exclusive rights associated with intangible ideas. Everyone had the positive right to make, use or sell his/her discoveries or inventions.<sup>4</sup> However, this right passed to the public when one permit-

---

2. THE ENCYCLOPEDIA OF PHILOSOPHY, 6 PROPERTY 491 (Macmillan, reprint ed. 1972).

3. See Cal. Civ. Code § 654 (West 1982). Ownership of a *thing* is the right of one or more persons to possess and use it to the exclusion of others. *Id.* In this Code, the *thing* of which there may be ownership is called *property*. *Id.*

4. Bauer v. O'Donnell, 229 U.S. 1, 10 (1912).

The right to make, use, and sell an invented article is not derived from the patent law. This right existed before and without the passage of the law, and was always the right of the inventor. The act secured to the inventor the exclusive right to make, use, and vend the thing patented, and consequently to prevent others from exercising like privileges without the consent of the patentee.

*Id.*

ted the idea's public use.<sup>5</sup>

A common law inventor could secret a new idea to maintain a *de facto* exclusivity; but another who lawfully came upon it possessed the same positive right and ability to dedicate it to the public. Thus, everyone had the right to possess and practice their ideas and no one had the negative right to exclude use by others.<sup>6</sup> This common law heritage continues today as the *trade secret* law (state law) alternative to the federal patent system.<sup>7</sup>

Common law *copyright* was likewise limited to the tangible medium in which an expression was fixed, i.e., transferring the medium containing one copy transferred the right to unlimited reproduction because there was no separate intangible subject matter property.<sup>8</sup>

#### B. INTELLECTUAL PROPERTY'S CARTESIAN DICHOTOMY

Modern *intellectual property* rewards the creative fruit of intellectual curiosity with a separate (subject matter) property status; however, its possessory interest remains physical. There are two fundamental juridical premises applicable to this new *intellectual property*. *First*, nothing may be removed from the public domain; that is, an inventor or author may only get a property interest in what s/he contributes to the public domain. This is generally satisfied by the novelty (subjective and objective) requirement. *Second*, preemption of scientific principle or abstract idea is prohibited; that is, all abstract ideas and scientific principles, as a portion of the storehouse of knowledge, remain available to the intellect and industry of us all.

So, (i) property remains the *de jure* right to exclude use of the *subject* by others, but now (ii) the *subject* may be tangible (for stuff you own) or intangible (to interfere with stuff owned by another); however, (iii) the *possessory interest* is always and only over objective stuff and may not

---

5. See *Wilson v. Rousseau*, 45 U.S. (4 How.) 646, 674 (1846).

At common law, the better opinion, probably, is, that the right of property of the inventor to his invention or discovery passed from him as soon as it went into public use with his consent; it was then regarded as having been dedicated to the public, as common property, and subject to the common use and enjoyment of all.

*Id.*

6. See *Gaylor v. Wilder*, 51 U.S. (10 How.) 477, 494 (1850). "Now the monopoly granted to the patentee is for one entire thing; it is the exclusive right of making, using, and vending to others to be used, the improvement he has invented, and for which the patent is granted. The monopoly did not exist at common law . . ." *Id.* at 494.

7. See generally *Kewanee Oil Co. v. Bicron Corp.*, 416 U.S. 479 (1984); *Sinclair v. Aquarius Electronics Inc.*, 42 Cal. App. 3d 216, 223-25 (1974).

8. See *Pushman v. N.Y. Graphic Soc., Inc.*, 39 N.E.2d 249 (1942). See also 17 U.S.C. § 201 (1984) (separating copyright ownership from ownership of the material object, reversing the common law rule).

preempt principle or idea and may not remove anything from the public domain.

Rene Descarte's division of reality into *subjective* (mental, intangible) and *objective* (physical, tangible) realms (the Cartesian divide) provides a useful context for examining intellectual property whose subject matter is abstract (mental conception or expression) but whose *possessory interest* applies *only* to others stuff.

### C. PATENTS

Patents provide an intellectual property right; that is, a patentee may exclude others from making or using *embodiments* (objective realm) of an inventive *conception* (subjective realm). Embodiment is an objective manifestation, that is, to invest with a physical body.<sup>9</sup> Patents preclude the *embodiment* of inventive conceptions *by others*. That is, while the *subject* of patent protection is an abstract inventive idea, patent's *possessory interest* is limited to its tangible embodiment by others. The patentee's common law right to possess and practice the invention with his/her own stuff remains intact, subject to any prior patent of another.

Patents provide *only* a negative right to interfere with the use of others' stuff when embodying a patented invention. They provide no possessory interest or right over any disembodied conception.

This limitation can be seen (i) in the patent statute subject matter provision: machines, articles and compositions are all tangible objects, and processes or methods have always been confined to the manipulation or transformation of objects,<sup>10</sup> (ii) in the statute's exclusionary right provision: allowing a patentee to prevent embodiment or use by others,<sup>11</sup> and (iii) in the judicial decisions prohibiting patents on (a) a mere abstract idea (i.e., patentable *conception* must comprise a *useful* (objective) implementation of the idea),<sup>12</sup> or (b) a fundamental principle or mathematical algorithm (i.e., patents are limited to a *useful* application of principle or algorithm).<sup>13</sup> Thus, a patent's *possessory interest* covers the

---

9. WEBSTER'S NEW INT'L DICTIONARY 739 (3d ed.1993).

10. 35 U.S.C. § 101 (1984); see *In re Schrader*, 22 F.3d 290, 293-95 (Fed. Cir. 1994) (process patentable if the series of steps transform or manipulate data or signals representing or constituting physical activity or objects); but see *Gottschalk v. Benson*, 409 U.S. 63, 71 (1972) (the contrary is not yet foreclosed).

11. 35 U.S.C. § 271(a) (1984) (prohibiting the unauthorized making, using or selling of a patented invention). See generally 5 DONALD CHISUM, CHISUM ON PATENTS §§ 16.01-16.02 (1997).

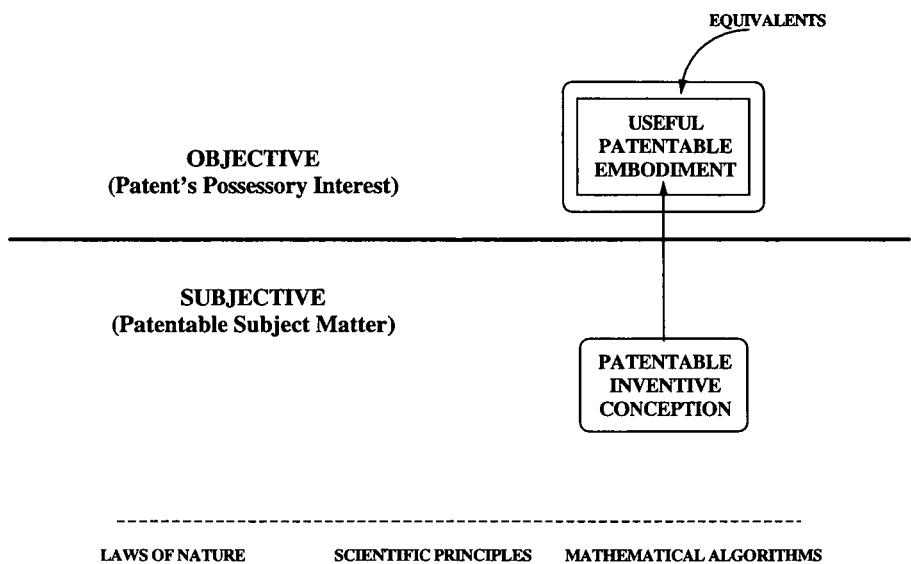
12. See *Rubber-Tip Pencil Co. v. Howard*, 87 U.S. 498, 507 (1874) ("An idea of itself is not patentable, but a new device by which it may be made practically useful is.").

13. See *LeRoy v. Tatham*, 55 U.S. 156, 175 (1852) ("A principle, in the abstract, is a fundamental truth. . ."); *Diamond v. Diehr*, 450 U.S. 175, 185 (1981) ("Excluded from . . . protection are laws of nature, natural phenomenon, and abstract ideas.").

objective embodiment, *by others*, of the invented conception, but may not preempt abstraction or preclude use of any principle or algorithm.

However, since patents are limited to *useful* inventions and the distinction between *subjective* conception and *objective* embodiment is so bright a line, patents enjoy (i) dominion over embodiments using equivalent elements, so long as all elements of the invented conception are present (as claimed or by equivalence),<sup>14</sup> as well as (ii) dominion over subsequent independent development. FIGURE 1 DISPLAYS THIS CARTESIAN VIEW.

Figure 1: PATENTS



D. COPYRIGHT

Copyright by contrast protects an original *expression of an idea* when fixed in a tangible medium.<sup>15</sup> Expression is symbolic representation;<sup>16</sup> that is, a token is being given a meaning beyond its physical na-

14. See Warner-Jenkinson Co. Inc. v. Hilton Davis, 117 S. Ct. 1040 (1997).

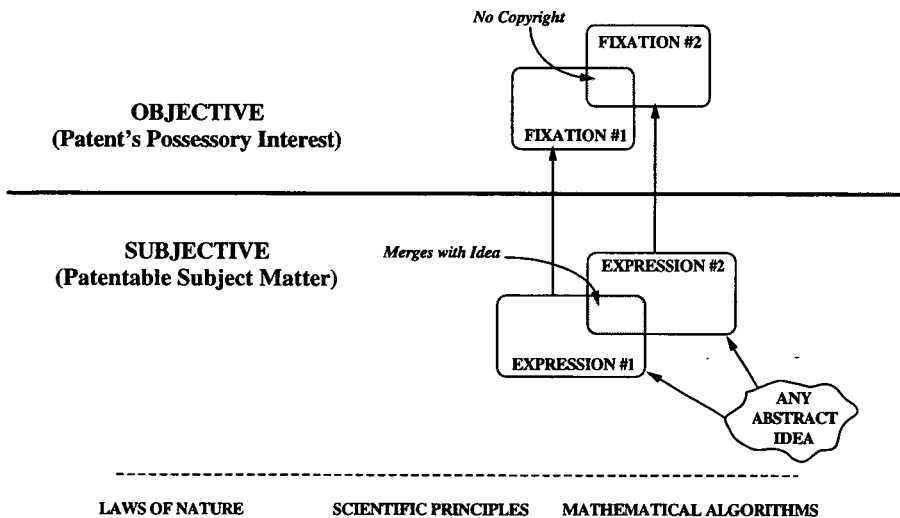
15. See 17 U.S.C. § 102(a) (1984).

16. 17 U.S.C. § 101 (1984) (defining literary works as those *expressed in words, numbers, or other verbal or numerical symbols or indicia*). See also WEBSTER'S NEW INT'L DICTIONARY 803 (3d ed. 1993).

ture. That additional meaning is abstract. So, both an idea and its expression are abstractions (which are why they're often difficult to distinguish). Thus, while the *subject* of copyright protection is abstract expression (independent of the physical medium), its *possessory interest* is limited to the medium in which the symbols are fixed.

To preclude preemption of ideas, the copyright statute expressly excludes *any idea, procedure, process, system, and method of operation, concept, principle, or discovery* from copyright protection.<sup>17</sup> So ideas must be distinguished from their expression. But since the distinction between idea and expression is not a bright line, where alternative expression is prevented, expression merges into idea and copyright is denied.<sup>18</sup> Thus, *copyright is certain only if alternative expression is certain*. FIGURE 2 DISPLAYS THIS CARTESIAN VIEW.

Figure 2: COPYRIGHT



So at this time, in this Cartesian culture, while the *subject* of property may be an abstract expression or conception; ideas, principles and

17. 17 U.S.C. § 102 (b) (1984). See also *Baker v. Seldon*, 101 U.S. 99 (1879) (copyright subsists in a work's expression and not the underlying ideas).

18. This merger doctrine is reflected in 17 U.S.C. §102(a) (1984) (codifying *Baker v. Seldon*, 101 U.S. 99 (1879) (excluding "any idea, procedure, process, system, method of operation, concept, principle, or discovery" from copyright protection)).



mathematical algorithms remain outside property's *possessory interest*, available to the intellect and industry of all, beyond *anyone's* ownership.

### III. THE ABSTRACT NATURE OF COMPUTER SCIENCE

[F]undamentally, computer science is a science of abstraction—creating the right model for a problem and devising the appropriate mechanizable techniques to solve it.

Every other science deals with the universe as it is. The physicist's job, for example, is to understand how the world works, not to invent a world in which physical laws would be simpler or more pleasant to follow. Computer scientists, on the other hand, must create abstractions of real-world problems that can be represented and manipulated inside a [digital process] computer.<sup>19</sup>

From the beginning, some declared computer science a literary art, while others proclaimed it a technical science, but under either view computer science regards modeling the use and operation of a digital process computing device.

Modeling computing devices has been with us since invention of the abacus 5,000 years ago in Babalonia. More recent useful devices include the slide rule (1614), Pascal's digital adding machine (1642), Jacquard's loom (1804), Babbage's analytical engine (1834), Scheutz's working difference engine (1853), the punched card tabulator (1890), Bush's differential analyzer (1930), Philbrick's Polyphemus (the first fully electronic analog computer, 1938), and the Atanasoff-Berry's electronic digital computer (1942).<sup>20</sup> Each device uses some technique of modeling *a logical problem for mechanized computation*.

The modern electronic digital computer<sup>21</sup> represents a distinctive paradigm shift in the nature and role of such devices. Digital computers use a two-digit binary number system to compute functions and decide

19. A.V. AHO & J.D. ULLMAN, FOUNDATIONS OF COMPUTER SCIENCE 1 (1992).

20. See generally TIME-LIFE BOOKS, UNDERSTANDING COMPUTERS, ILLUSTRATED CHRONOLOGY AND INDEX, FIFTY CENTURIES OF COMPUTING (1985).

21. See ANTHONY RALSTON & EDWIN D. REILLY, ENCYCLOPEDIA OF COMPUTER SCIENCE 434 (3d ed. 1993), stating:

*A digital computer* is a machine that will accept data and information presented to it in its required form, carry out arithmetic and logical operations on this raw material, and then supply the required results in an acceptable form . . . . The sequence of the operations required to produce the desired output must be accurately determined and specified by people known as system designers (or analysts) and programmers . . . . The main characteristics of the computer are that it is automatic, general purpose, electronic, and digital.

. . . .

Digital computers are so called because they work with numbers in the form of separate discrete digits. More precisely, they work with information that is in digital or character form, including alphabetic and other symbols as well as numbers.

*Id.*

predicates. To understand the elegant simplicity and distinct nature of digital computing we briefly review the Turing machine and the theory of algorithms, then compare digital to analog computing and, finally, we distinguish the computer science abstract logical model from its computing algorithm and compare computer science ingenuity to natural science ingenuity.

#### A. THE TURING MACHINE AND THE DIGITAL COMPUTER

Alan M. Turing (1912-54)<sup>22</sup> was a brilliant British logician and mathematician whose contribution to computation would be difficult to overstate. In 1935 his interest and attention focused upon mathematical logic and in 1937 he published his celebrated paper introducing the concept of a Turing machine.

A Turing machine . . . consists of (1) a control unit which can assume any one of a finite number of possible states; (2) a tape, marked off into discrete squares, each of which can store a single symbol, taken from a finite set of possible symbols; and (3) a read-write head, which moves along the tape and transmits information to and from the control unit (figure omitted).

The Basic Model: A Turing machine computes via a sequence of discrete steps. Its behavior at a given time is completely determined by the symbol currently being scanned by the read-write head, and by the internal state of the control unit. On a given step, it will write a symbol on the tape, move along the tape one square to the left or right, and enter a new internal state. The new symbol is permitted to be the same as the current symbol; similarly, it is permissible to stay on the same tape square on a given step and/or to reenter the same state. Certain symbol state situations may cause the machine to halt . . .

The program of a Turing machine defines its action for the various state-symbol combinations that are possible. . . .

. . . As is often the case, the algorithm is best thought of as an exercise in symbol manipulation rather than as arithmetic.<sup>23</sup>

Since it is not an actual machine or device, it might better be called a Turing program or concept. It is an abstract mathematical notion of how problems can be solved in a binary number system; that is, a system of only two symbols, zero and one. A square on the tape being read may be a "1," a "0," or blank. In combination with the control unit's internal state (one of a finite number of states), the symbol read on the tape determines (i) whether the machine writes a zero or a one, (ii) whether it shifts left or right, and (iii) what will be the next state of the control unit. The total number of available discrete operations is the product of possi-

---

22. *Id.* at 1394-95.

23. *Id.* at 1397-1401.

ble control unit states times tape states; although the machine only writes a zero or a one and only moves one space right or left. An incredibly simple set of operations.

In practical effect an electronic digital process computer is a Turing machine, that is, it manipulates binary numbers, changing zeros to ones and ones to zeros. However, an electronic computer performs billions of such operations every second; and by breaking complex computation down into a finite set of such simple symbol manipulations, the modern electronic computer provides incredible computation ability.

#### B. THE THEORY OF ALGORITHMS: COMPUTING FUNCTIONS & DECIDING PREDICATES

A computer program is an algorithm for manipulating binary number digits. An algorithm is a precisely stated set of steps to solve a computation.

In the theory of computation, one is mainly concerned with algorithms that are used either for computing functions or for deciding predicates.

A function  $f$  with domain  $D$  and range  $R$  is a definite correspondence by which there is associated with each element  $x$  of the domain  $D$  (referred to as the "argument") a single element  $f(x)$  of the range  $R$  (called the "value"). The function  $f$  is said to be computable (in the intuitive sense) if there exists an algorithm that, for any given  $x$  in  $D$ , provides us with the value  $f(x)$  . . . .

A predicate  $P$  with domain  $D$  is a property of the elements of  $D$  that each particular element of  $D$  either has or does not have. If  $x$  in  $D$  has the property  $P$ , we say that  $P(x)$  is true; otherwise we say that  $P(x)$  is false. The predicate  $P$  is said to be decidable (in the intuitive sense) if there exists an algorithm that, for any given  $x$  in  $D$ , provides us with a definite answer to the question of whether or not  $P(x)$  is true . . . .

The computability of functions and the decidability of predicates are very closely related notions because we can associate with each predicate  $P$  a function  $f$  with a range  $\{0, 1\}$  such that, for all  $x$  in the common domain  $D$  of  $P$  and  $f$ ,  $f(x) = 0$  if  $P(x)$  is true and  $f(x) = 1$  if  $P(x)$  is false. Clearly,  $P$  is decidable if and only if  $x$  is computable . . . .<sup>24</sup>

So, a computer program provides a precise set of instruction for determining which manipulation (from a finite set) to conduct for each step of the process. The instruction is always in the same form, that is, *if* ( $a$ ), *do* ( $x$ ).

The Church-Turing thesis states, any computation solvable by a precisely stated set of instruction (i.e., an algorithm) can be run on a Turing machine (or digital process computer).

---

24. *Id.* at 37-39.

### C. DISTINGUISHING DIGITAL AND ANALOG COMPUTING

A computer may be either digital or analog. The two types do have some principles in common, but they employ different types of data representations and are, in general, suited to different kinds of work. Digital computers are so called because they work with numbers in the form of separate discrete digits. More precisely, they work with information that is in digital or character form, including alphabetic and other symbols as well as numbers.

In a digital machine, the data, whether numbers, letters, or other symbols, is represented in digital form. An analog computer, on the other hand, may be said to deal with a[n analogy] of the problem, in which the variables are represented by continuous physical quantities such as angular position and voltage . . . . Using familiar devices, we could say that a slide rule is an analog device because numbers are represented by linear length. The abacus, on the other hand, is a digital device, because movable counters are used for calculating.

Digital computers differ from analog computers much as counting differs in principle from measuring. Both type of machine employ electric currents, or signals, but in the analog system, a number is represented by the magnitude (e.g., voltage) of a signal, whereas, in a digital computer, it is not the magnitude of signals that is important, but rather the number of them, or their presence or absence in particular positions. Analog computers tend to be special-purpose machines<sup>25</sup> designed for some specific scientific or technical application . . . . In commercial and administrative data processing and for mathematical computation, we are concerned almost exclusively with digital computers.<sup>26</sup>

The input to an *analog* computer is a direct and continuous measurement of a scientific principle's impact upon an objective physical circumstance. The analog computer creates an electronic analogy to the changing physical phenomena. This functional relationship between physical phenomena and the operation of an analog computer is *critical* to our discussion, because it distinguishes a digital from an analog computer; that is, the difference between direct measurement of value and counting or manipulating symbols.

Digital computation is symbol manipulation, it has no causal relationship to objective phenomena or scientific principles. Digital computers change symbolic zeros and ones by flipping the state of a circuit on/off. Thus, while analog computing is directly and functionally depen-

---

25. See *In re Alappat*, 33 F.3d 1526 (Fed. Cir. 1994). When the Court of Appeals for the Federal Circuit suggests a programmed general purpose digital computer is a *special purpose computer* for the functionality of the program instruction it *cannot* mean *special purpose computer* as used to describe an *analog* computer, for digital computing does not become analog computing by mere programming. *Id.* at 1545.

26. See RALSTON & REILLY, *supra* note 21, at 434-35.

dent upon application of scientific principle to physical phenomena, the operation of a digital process computer is completely independent of both.

#### D. ABSTRACT MODELS, ALGORITHMS AND INGENUITY

Computer science creates an abstract logical model of a practical problem expressible as an algorithm to calculate functions and determine predicates. A two-digit binary number system is used to map the algorithm's symbol manipulation (across the Cartesian divide) to the on/off operation of electronic circuit processing. The logical model itself is a *symbolic allegory* distinct from the physical process of a computer executing instructions; that is, no physical analog exists between the logical model and what occurs within the computing device. Unlike natural science, where inventive conception is confined to the objective application of scientific principles, an instructed computer (as such) does not manifest the conceived logical model or the practical use of the calculation result. Indeed, all the usefulness of computer science ingenuity lies hidden in the subjective meaning of the symbols and simply doesn't occur until the computed result is applied to the context of the question solved.

A natural scientist applies physical science to physical phenomena; that is, s/he conceives solutions to concrete (objective) problems. The computer scientist assigns meanings to symbols (the abstract model) and develops the steps (algorithm) of a symbol manipulating process; that is, s/he conceives solutions to abstract problems.

The significant differences between natural science and computer science are: (i) process computation is an abstract principle that *does not* occur in nature, (ii) process computation is independent of the physical form or mechanism used (i.e., symbol manipulation is as accurately accomplished with beer cans and ping pong balls as with CPUs and memories); and (iii) natural science principles are several and immutable, while computer science is premised upon a single mutable principle—the flexible algorithm of instruction.

Human ingenuity in natural science differs from ingenuity in computer science. The question is: what difference does the difference make in applying intellectual property principles? How is ingenuity in modeling process logic and encoding its mathematical expression to be recognized and rewarded? What property can there be in computation logic, in mediating instruction, in an instructed computer, or in applying the computation to a practical use?

## IV. COMPUTER SCIENCE AND INTELLECTUAL PROPERTY LAW

## A. THE ISSUE RESTATED

Computer science provides a gateway to the Information Age, but it also vexes the intellectual property system of our Industrial Revolution. Patents were established to protect the utilitarian application of natural science to industrial technology. Copyrights evolved to protect non-utilitarian (literary/aesthetic) symbolic expressions. Yet, computer science has *both* utilitarian and symbolic aspects.

Natural science ingenuity lies in the selection and application of scientific principles to natural phenomena. The physical changes produced are measured for novelty; ingenuity in selection is evaluated against what is obvious to an artisan; and if new, useful and non-obvious, a patent is available.

Computer science ingenuity lies in modeling a practical question into a process computation dichotomy and in writing a script of instruction. The logical model is allegorical and distinct from the encoded process or its mechanization, that is, there is no physical analog between the logical model and what occurs or exists within a computing device. So, computer science ingenuity does not cause change to physical phenomena, since no causal relationship crosses the Cartesian divide.

As the review below will show, this computer science and natural science difference raises several new patent issues, including:

1. Is encoded machine instruction patentable, *per se*; and if not, is an instructed machine patentable?

2. Since utility does not occur until a computation result is applied to a practical problem, is encoded instruction or instructed machine patentable apart from the context of its use?

3. Assuming a useful *context*, is computer science ingenuity sufficient or is physical novelty always necessary; that is, is computer science merely a permissible but irrelevant adjunct to natural science ingenuity?

4. May the context or *practical use* be mere information processing?

5. Do our property law premises (i.e., nothing removed from the public domain, no property in mere abstract ideas and possessory interests limited to physical stuff) limit the available interests?

At the end of the day, the question is, will the ingenuity of a computer scientist be given parity with natural science ingenuity, and if so, how may our property law premises be maintained?

## B. COPYRIGHT DISAMBIGUATION

Initially, copyright protection for computer programs was uncertain because instruction was accused of being too *useful*. The National Commission on New Technological Uses ("CONTU") and an associated Copy-

right Amendment resolved this issue by recognizing the expressive nature of the program, assuring it copyright protection.<sup>27</sup> But those relying on copyrighted expression eventually found they had inadequate protection,<sup>28</sup> for computer science ingenuity lay in the logical model, not its expressed instruction and copyright provides no interest over subsequent independent development.

### C. PATENT'S EARLY CONTEXT

As suggested by a (Vice President of IBM) member of the President's Commission on the Patent System, a 1966 Commission report, "*To Promote The Progress of . . . Useful Arts*" *In An Age of Exploding Technology*, recommended the patent statute be amended to provide:

A series of instructions which control or condition the operation of a data processing machine, generally referred to as a 'program,' shall not be patentable regardless of whether the program is claimed as: (a) an *article*, (b) a *process* described in terms of the operations performed by a machine pursuant to a program, or (c) one or more *machine* configurations established by a program.

The amendment was never enacted, notwithstanding repeated attempts; however, with such an introduction a struggle over patentability was predicable.

### D. THE UNITED STATES SUPREME COURT COMPUTER SCIENCE DECISIONS

#### 1. *Benson—Liberates Mechanized Process Computing*

Quoting the 1966 Presidential Commission Report, the 1972 United States Supreme Court *Gottschalk v. Benson* decision held a *method* for converting binary-coded decimal numerals into pure binary numerals in a general purpose digital computer, was an unpatentable preemption of a mathematical algorithm.<sup>29</sup> The claimed process was not limited to any

---

27. THE FINAL REPORT OF THE NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS (1978) [hereinafter CONTU] (concluding copyright is the appropriate mechanism for protecting computer software). The Computer Software Copyright Act of 1980 (Act of Dec. 12, 1980, Pub. L. 96-517, § 10, 94 Stat. 3015) incorporated CONTU by amending 17 U.S.C. § 101 (1984) to include a definition of *computer program*, and by adding 17 U.S.C. § 117 (1984), the right to make interim copies. The copyright statute already allowed *literary works* (17 U.S.C. § 102(a) (1984)) defined as *verbal or numerical symbols or indicia, regardless of the nature of the material objects* (17 U.S.C. § 101 (1984)). See H.R. Rep. No. 1476, 94th Cong. 2d Sess. 54 (1976). See also The Computer Software Copyright Act of 1980, Act of Dec. 12, 1980, Pub. L. 96-517, § 10, 94 Stat. 3015, and the accompanying House Report, H.R. Rep. No. 96-1307, Part 2, 96th Cong., 2d Sess. (1980) p.19, (applying the 1976 Act to computer programs).

28. See, e.g., *Lotus Dev. Corp. v. Borland Int'l Inc.*, 116 S. Ct. 804 (1996) (menu command hierarchy non-copyrightable method of operation).

29. *Gottschalk v. Benson*, 409 U.S. 63 (1972) (Douglas, J.) (no dissent, 3 abstentions).

art or technology, apparatus or machinery, or particular end use, and purported to cover any use of the *method* in any general purpose digital computer.<sup>30</sup>

The Court recited several precedents holding (i) scientific truths, mathematical expressions, mere ideas and natural phenomena were unpatentable;<sup>31</sup> and (ii) *transformation and reduction of an article "to a different state or thing," is the clue to patentability of a process claim that does not include particular machines.*<sup>32</sup> Yet the Court tempered the latter (but not the former) cases by concluding:

It is argued that a process patent must either be tied to a particular machine or apparatus or must operate to change articles or materials to a "different state or thing." We do not hold that no process patent could ever qualify if it did not meet the requirements of our prior precedents . . . . What we come down to in a nutshell is the following.

It is conceded that one may not patent an idea. But in practical effect that would be the result if the formula for converting BCD numerals to pure binary numbers were permitted in this case. The mathematical formula involved here has no substantial practical application except in connection with a digital computer, which means that if the judgment below is affirmed, the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself.<sup>33</sup>

## 2. *Benson & Mechanized Process Computation*

*Benson* included only *process* claims to an abstract algorithm (although one claim did include *shift registers*),<sup>34</sup> but does it apply to a *machine* programmed with an algorithm?

*Benson's* preclusion of patenting mathematical algorithms seems reasonably premised. Consider removing just one mathematical formula from all others. The probability any problem would be denied reasonable solution seems small. Indeed, perhaps several formulas could be removed without denying solution to any problem; but how many before probabilities intersect and new practical uses are denied process computing? Addressing this very issue the Court held every known or unknown process computing algorithm is analogous to a principle of natural science, patentable only in non-preemptive *practical uses*. Thus, the Court placed all computing algorithms into the storehouse of knowledge available to the intellect and industry of all.

---

30. *Id.* at 64.

31. *Id.* at 67-69.

32. *Id.* at 69-72 (emphasis added).

33. *Id.* at 71-72.

34. *Id.* at 73-74.



But if an algorithm provides faster computer processing, shouldn't its computer embodiment be patentable as an improved computing machine, or would that constitute preemption? Some argue computing is advanced by an algorithm no less than when a new circuit design is patented. However, the principles of operating a computer are unchanged by the algorithm being processed, that is, there is no change in physical processing speed, efficiency or function, only a change in the computational steps taken to determine a value. Furthermore, circuit designs do not preempt principle or algorithm. Such patents protect only an invented circuit, stated in terms of the physical laws applied to its electric components, *not* as an abstract calculating logic or all other circuit use of it. By contrast, a digital computer patented in terms of an algorithm would prevent use by all computer architectures (known or later developed), in all computer languages, for all utilities, that is, it preempts computer use of the algorithm.

Theoretically there are endless ways to a mathematical value, one algorithm is only one way; but it is a specific way, demonstrating a computational truth in abstract relationship. Since much of *objective* reality may be logically modeled by computing truths, patenting an algorithm based upon its application to one *objective* circumstance unjustly removes it from the public domain, if doing so denies use to other circumstances. An algorithm may conveniently express a truth of an invention, but it is not the substance of the invention until the context of its *practical use* is revealed; and that is the most an inventor can be said to provide.

The algorithm concern expressed in *Benson* was focused upon a computer program. However, as Justice Douglas acknowledged, the *only practical use* of such algorithms is in a mechanical computing device; and, patenting that mere mechanized performance of an algorithm was the objectionable preemption. So, merely programming a computing device and calling it a novel patentable machine<sup>35</sup> seems patently inconsis-

---

35. Some suggest *In re Alappat*, 33 F.3d 1526 (Fed. Cir. 1994), stands precisely for that, however, the facts in *Alappat* belie that claim. While some of the court's language is ambiguous; see, e.g.:

[C]laim 15 would read on a general purpose computer programmed to carry out the claimed invention . . . . We have held that such programming creates 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.

*Id.* at 1545.

[T]he proper inquiry . . . is to see whether the claimed subject matter as a whole is a disembodied mathematical concept . . . which . . . represents nothing more than a "law of nature," "natural phenomenon," or "abstract idea."

*Id.* at 1544.

Nonetheless, Claim 15 was limited to a rasterizer use of the programmed computer. *Id.* at 1542-43. The circumstance the court might not have considered (and the facts did

tent with *Benson*, the only Supreme Court decision dealing with actual preemption. Mere mechanization is preemptive, the programmed device needs a *practical use* limitation.

### 3. *Computer Science Embodiment and Utility*<sup>36</sup>

If the physical *embodiment* of an algorithm in a computing device is unpatentable without a utility limitation, the algorithm preemption concern in patent law may be analogized to the preemption of idea concern in copyright law.<sup>37</sup> As we saw, both an idea and its copyrightable expression are *subjective* abstractions in the Cartesian divide and their distinction is often difficult; so copyright's doctrine of merger requires the availability of *alternative expression* to assure ideas are not preempted. Likewise, an unpatentable mechanized algorithm and its patentable *practical use* are both *objective* and their distinction may be difficult, so patenting could be premised on the nature and availability of *alternative practical uses*.

---

not provide) is mere embodiment of an algorithm in a computing device with no limitation to a *practical use*; admittedly, the above language can be read to suggest Claim 15 covers a mere programmed computer, as such, even when not designed for use as a rasterizer, but then it would appear inconsistent with *Benson*, the only Supreme Court decision factually dealing with preemption.

36. The utility requirement of patent law arises from the "*useful Arts*" limitation of the Constitution. U.S. CONST., art. 1, § 8, cl. 8, provides Congress the power: "To promote the progress of science and useful arts, by securing for limited times to authors and inventors the exclusive right to their respective writings and discoveries." *Id.* The prevailing view of this *balanced* provision is that *science* pertains to *authors/writings*, while *useful arts* pertain to *inventors/discoveries*. See Karl B. Lutz, *Patents and Science*, 32 J. PAT OFF. SOC'Y 83 (1950); Giles S. Rich, *The Principles of Patentability*, 42 J. PAT OFF. SOC'Y 75, 76-80 (1960). The *useful Arts* had a unitary meaning including: "... the so-called industrial, mechanical and manual arts of the 18th century, which of course stemmed back to the useful arts of antiquity insofar as knowledge thereof had been handed down or rediscovered." *Id.* See also Robert I. Coulter, *The Field of the Statutory Useful Arts*, 34 J. PAT OFF. SOC'Y 487, 496 (1952).

However, the term was intended to specifically *broaden* the limited British statutory phrase *new manufactures*, to assure *new processes* were included. See Karl B. Lutz, *Patents and Science*, 32 J. PAT OFF. SOC'Y 83, 86 (1950):

The British Statute of Monopolies (1623) permitted patents for "new manufactures" only . . . . [B]y the year 1787 it was being recognized even in Great Britain that the phrase "new manufactures" was an unduly limited object for a patent system, since it seemed to exclude new processes. [Which was] resolved in the United States Constitution by broadening the field from "new manufactures" to "useful arts" . . . .

*Id.*

While *practical use* meant any *technical* application of scientific principle, utility was significant only in its absence; digital process computing, however, may seek utility as a means to confine a patent's scope to prevent preemption.

37. See *Parker v. Flook*, 437 U.S. 584, 589 (1978) (Stevens, J.) (3 dissensions) (seemingly pointing toward this parallelism when discussing patentable process versus unpatentable principle).

So long as *practical use* meant the application of a scientific principle, physical embodiment and novelty were assured, and utility was significant only in its absence (i.e., any use suffices for natural science ingenuity). Physical novelty assures the premises of our intellectual property (i.e., nothing removed from the public domain and no property in mere abstract ideas).

Computer science ingenuity however does not assure physical novelty or embodiment; and indeed, as *Benson* shows, mere embodiment of an algorithm is preemptive, absent a practical use limitation. So for computer science ingenuity, utility is a limitation (not just as a threshold, as in natural science). If an instructed computer is preemptive, a *practical use* limitation confines a patent and resolves the preemption concern. And, by limiting patentable computer science ingenuity to a new *use* of an algorithm, the absence of physical novelty is resolved. Thus, requiring computer embodiment and a novel utility assures the premises of our intellectual property, even though the ingenuity is abstract.

#### 4. *Flook—Rejects Computer Science Ingenuity*

The Court's 1978 *Flook* decision held a *method* for updating alarm limit values for the catalytic conversion of any hydrocarbon was unpatentable where the only novel feature was the mathematical algorithm used to calculate the values.<sup>38</sup> While calculating updated alarm limits was a new step, the Court held all known or unknown mathematical algorithms were in the public domain and as such were to be ignored. So *Flook* required natural science ingenuity and rejected conventional post-solution activity even though limited to the physical realm and to a specific (*albeit* broad) range of end use (thus, it was not preemptive).

*Flook* would refuse every patent premised on computer science ingenuity; that is, it demands traditional physical novelty. Also, *Flook* established the inadequacy of mere post-solution activity; that is, the use must be a more traditional *useful art*, not mere information processing.

#### 5. *Diehr—An Uncertain Revelation*

In *Diehr*<sup>39</sup> (1981) the Court again analogized mathematical algorithms to unpatentable scientific principles.<sup>40</sup> *Diehr* held an old *process* for curing rubber which included measuring mold temperature but now

---

38. *Id.* at 584.

39. *Diamond v. Diehr*, 450 U.S. 175 (1981) (Rehnquist, J. ) (4 dissensions).

40. *Id.* at 187. Referring to *Benson*, the Court stated: "We defined 'algorithm' as a 'procedure for solving a given type of mathematical problem' and we concluded that such an algorithm, or mathematical formula, is like a law of nature, which can not be the subject of a patent." *Id.*

using a prior known mathematical formula to compute the precise time to open the mold, was traditional patentable subject matter deserving examination.

*Diehr* distinguished *Flook*, by noting *Flook* claimed an algorithm's use in the abstract without limitation to the objective elements of a patentable process, whereas *Diehr* was so limited. However, by stating, "[i]t is now commonplace that an application of a law of nature or mathematical formula to a *known* structure or process may well be deserving of patent protection"<sup>41</sup> and holding, "the 'novelty' of any element or step in a process, or even of the process itself, is of no relevance in determining statutory subject matter."<sup>42</sup>

*Diehr* cast considerable doubt on *Flook's* requirement for the physical novelty of natural science. Yet *Diehr* favorably cited *Flook* for the inadequacy of *insignificant post-solution activity*. So while a use may be old and absent physical invention, it apparently requires more of a traditional *useful art* than mere *insignificant post-solution activity*.

To restate, *Flook* asked if computerizing an old hydrocarbon catalytic conversion process with a *new formula* might be patentable if preemptive, and was told *no*, computer science ingenuity was irrelevant. Three years later, *Diehr* asked if computerizing a traditional rubber curing process with an *old formula* might be patentable and was told *yes*, if stated as a traditional physical process. In *Flook*, said *Diehr*, the process was too abstract, lacking sufficient physicality.

#### 6. *Are Mathematical Algorithms Analogous to Scientific Principles?*

In *Benson* the Supreme Court analogized mathematical algorithms to scientific principles for purposes of preemption and placed them in the public domain, but it has yet to articulate their relationship or the impact of their differences.

Scientific principles act only within and directly upon the *objective* realm. They are known *only* by their measured and consistent impact upon physical phenomena. Since *objectively* determined, they are not *subjectively* mutable and ingenuity is limited to conception of their *objective* application.

Process computing, by contrast, does not occur in nature.<sup>43</sup> It acts only upon the *subjective* meanings assigned to tokens, is confined only by its logic or numeric system and is *subjectively* mutable; indeed, computer science ingenuity is limited to abstract models and has no physical im-

---

41. *Id.* (emphasis added).

42. *Id.* at 188 (emphasis added).

43. Conversation with John Serle, Author, *The Philosophy of Mind*, Professor of Philosophy and Linguistics, Univ. of Calif. at Berkeley (source on file with the author).

pact until a computing device executes an algorithm of instruction and applies the result to some other useful context.

Application of a scientific principle to a physical structure or process is well known in patent law; it produces an *objective* measurable event, used to judge novelty and obviousness. Applying a new algorithm, however, may change computation process to enhance speed or provide *new and useful* information without changing physical *structure, process or result*.<sup>44</sup> By analogizing mathematical algorithms to scientific principles did the Court intend parity in patentability as well as preemption?

*Flook* assumed the formula was *novel* but the process was old<sup>45</sup> and held:

The process itself, not merely the mathematical algorithm, must be new and useful. Indeed, the novelty of the mathematical algorithm is not a determining factor at all. Whether the algorithm was in fact known or unknown at the time of the claimed invention, as one of the "basic tools of scientific and technological work" [citing *Benson*], it is treated as though it were a familiar part of the prior art.<sup>46</sup>

*Flook* rejects computer science ingenuity as a basis for patentability. Not surprisingly, it expressly limits patents to the application of scientific principles and requires more than mere *post-solution* activity.

In *Diehr* both the formula and process were old and novelty was limited to continuous temperature measurement and computing cure time.<sup>47</sup> While limited to determining patentable subject matter where a scientific principle was present but unchanged, *Diehr* held:

[W]hen a claim containing a mathematical formula implements or applies that formula in a structure or process which, when considered as a whole, is performing a function which the patent laws were designed to protect (e.g., transforming or reducing an article to a different state or thing), then the claim satisfies the requirements of § 101.<sup>48</sup>

The example given by the Court was the traditional application of natural science, however, *Diehr* suggested physical novelty was irrelevant: "The "novelty" of any element or step in a process, *or even of the process itself*, is of no relevance in determining whether the subject matter of the claim falls within the § 101 categories of possibly patentable

---

44. The cases are divisible by the question of physicality. Some (e.g., *In re Arrhythmia*, 958 F.2d 1053 (Fed. Cir. 1992)) find a patentable *process or machine* incidentally applying a mathematical algorithm. Others reject patentability for lack of any material embodiment (e.g., *In re Meyer*, 688 F.2d 789 (C.C.P.A. 1982); *In re Grams*, 888 F.2d 835 (Fed. Cir. 1989); *In re Warmerdam*, 33 F.3d 1354 (Fed. Cir. 1994)) (representing no more than mere manipulation of ideas). But what if there is embodiment without any change in scientific principle?

45. *Parker*, 437 U.S. at 588.

46. *Id.* at 591-92.

47. *Diamond*, 450 U.S. at 177-81.

48. *Id.* at 192.

subject matter."<sup>49</sup> So, for a claim limited to a *function that the patent laws were designed to protect* the *Flook* requirement of physical novelty appears to be overruled, *sub silentio*, by *Diehr*.

*Diehr* did continue an uncertain shadow of *Flook* by citing it as precedent precluding patents where (a) there is *insubstantial post-solution activity* or (b) a formula is merely limited to a *particular technological environment*. So, practical use is significant to the patentability of computer science ingenuity. If application to a traditional *useful art* is required, but physical novelty is not, then computer science ingenuity in hastening a known structure or process to a known result, may be patentable; but patentability of computer science ingenuity in enhancing insignificant post-solution activity (such as displaying manipulated information on a computer screen, as in today's Information Age) would remain speculative.

Yet, if the premises of intellectual property are assured, there is no apparent reason to deny the recognition and reward of patenting to computer science ingenuity. As discussed, requiring mechanization and a novel use limitation assures those premises, allowing the analogy between natural science principles and computer science algorithms in both preemption and patentability; if application of computer science is viewed as a *useful art* and the *Flook* shadow removed.

#### E. THE COURT OF CUSTOMS AND PATENT APPEALS AND COURT OF APPEALS FOR THE FEDERAL CIRCUIT ALGORITHM-CENTRIC CASES

The Court of Customs and Patent Appeals (CCPA) and its successor the Court of Appeals for the Federal Circuit (CAFC) decisions followed the Supreme Court's algorithm-centric analysis.

(a) In 1978 pre-*Flook*, *Freeman*<sup>50</sup> held patentable a *process* and *machine*, computer implemented typesetting system, using a conventional typesetter but not stating a mathematical algorithm. The court asked two questions: does the claim recite an algorithm, and if so, is it wholly preempted?

(b) In 1979 *Bradley*<sup>51</sup> held patentable a computing *machine* switching system for multi-programmed operation using firmware microcode including a mathematical algorithm that was not part of the claimed invention. The court distinguished how a computer works (claimed) from what it does with real world data (not claimed), and was sustained on appeal by a four to four vote of the Supreme Court.

---

49. *Id.* at 188-89 (emphasis added).

50. *In re Freeman*, 573 F.2d 1237 (C.C.P.A. 1978).

51. *In re Bradley*, 600 F.2d 807 (C.C.P.A. 1979), *aff'd sub nom. Diamond*, 450 U.S. 381 (1981).

(c) In 1980 *Walter*<sup>52</sup> held unpatentable the *process* and *machine* computational unscrambling of reflected seismic waves, comprising a mathematical exercise in the abstract with *no substance apart from the calculations involved*.<sup>53</sup> Following *Flook*, mere improved calculation was insufficient for patenting; i.e., there must be a new and useful structure or end. The court changed the *Freeman* test to ask if an algorithm is applied to structural physical elements or limits process steps; or is merely solved, even if post-solution activity or a preamble field of use limitation is present.

(d) In 1982 *Taner*<sup>54</sup> held patentable an improved *process* of seismic exploration using simulated seismic wavefronts to determine subsurface formations that included a calculation. Seismic signals were viewed as *physical apparitions* and the algorithm was used to transform the physical signals.

(e) In 1982 *Abele*<sup>55</sup> held the *process* and *machine* for calculation and display (by gray scale shading) of data values in a field, unpatentable where applied to any data (notwithstanding the display step) and patentable when limited to X-ray attenuation data, since the latter required specific (process) steps beyond mere data gathering. It was a *conventional CAT scan process* and addition of an algorithm did not render a statutory process non-statutory. The court modified the *Freeman-Walter* test to ask only if a claim with an algorithm is *otherwise statutory*; i.e., applied in any manner to physical elements or process steps and limited by more than a field of use or non-essential post-solution activity.

(f) In 1982 *Meyer*<sup>56</sup> held unpatentable a *process* and *machine* used to test for probable malfunctions in any complex system; the literally described algorithm was preemptive, where not applied to physical elements or their process steps.

(g) In 1989 *Grams*<sup>57</sup> held unpatentable a *process* to test complex systems; where the literally described algorithm was preempted and the only physical step was gathering data.

(h) Also in 1989 *Iwahashi*<sup>58</sup> held patentable a *machine* used to calculate auto-correlation coefficients for use in pattern recognition; a computer combination of means interrelated by an algorithm operation, where at least one element was a specific component (ROM) did not claim every means, so it was not viewed as a *process*.

---

52. *In re Walter*, 618 F.2d 758 (C.C.P.A. 1980).

53. *Id.* at 769.

54. *In re Taner*, 681 F.2d 787 (C.C.P.A. 1982).

55. *In re Abele*, 684 F.2d 902 (C.C.P.A. 1982).

56. *In re Meyer*, 688 F.2d 789 (C.C.P.A. 1982).

57. *In re Grams*, 888 F.2d 835 (Fed. Cir. 1989).

58. *In re Iwahashi*, 888 F.2d 1370 (Fed. Cir. 1989).

(i) In 1992 *Arrhythmia*<sup>59</sup> held patentable a *process* and *machine* for analyzing electrocardiograph signals to determine heart activity; using a computer operation to convert input signals to different output signals, a physical thing was transformed; even though all the mathematical procedures were previously known in the abstract.

(j) In 1994 *Schrader*<sup>60</sup> held unpatentable a *process* for competitively bidding on a plurality of related items; the literal algorithm was preempted where bids were not physical, mere manipulation of data constitutes no physical change, effect or result; but the court acknowledged patentability where physicality is involved.<sup>61</sup> The algorithm was a well known optimization procedure.

There were others, but the above case law fairly represents the fundamental principles and approaches taken by the intermediate appellate court under the Supreme Court decisions, until 1994.

#### F. THE 1994 COURT OF APPEALS FOR THE FEDERAL CIRCUIT SHIFT TO A NON-ALGORITHM PERSPECTIVE

After twenty-two years of algorithm-centric analyses, 1994 was a watershed year for a shift in the CAFC perspective on computer science patentability:

(a) First, a divided *in banc Alappat*<sup>62</sup> held an instructed general purpose computer effectively becomes a special purpose computer for performing the instructed functions,<sup>63</sup> which may support patentability, and questioned the validity of *any* algorithm-centric consideration of patentability beyond a claim for a mere *disembodied mathematical concept*.<sup>64</sup>

59. *In re Arrhythmia*, 958 F.2d 1053 (Fed. Cir. 1992).

60. *In re Schrader*, 22 F.3d 290 (Fed. Cir. 1994).

61. *Id.* at 293-95 (acknowledging a computer process is patentable if the series of steps transform or manipulate data or signals representing or constituting physical activity or objects).

62. *In re Alappat*, 33 F.3d 1526 (Fed. Cir. 1994).

63. *Id.* at 1526, 1545 (1994), where the court stated:

We have held that . . . 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. . . . Consequently, a computer operating pursuant to software may represent patentable subject matter, provided, of course, that the claimed subject matter meets all of the other requirements of Title 35 . . . .

*Id.* (emphasis in original).

64. *Id.* at 1544.

A close analysis reveals . . . the Supreme Court never intended to create an overly broad, fourth category of excluded from § 101. Rather, at the core of the Court's analysis . . . lies an attempt . . . to explain a rather straight forward concept, namely, that certain types of mathematical subject matter, standing alone, represent nothing more than abstract ideas until reduced to some type of practical application . . . . Given the foregoing, the proper inquiry . . . is to see whether the



(b) Second, *Warmerdam*<sup>65</sup> held unpatentable an abstract *process* for generating a data structure representing the shape of physical objects as hierarchies of bubbles, but held patentable a claim to *any* machine containing in its memory *any* data representing an object in one of the described prior art bubble hierarchies. After accounting for the mathematical algorithm issue, the court concluded the concern of its originators was over *abstract ideas* and could be satisfied by a *physicality* requirement; thus, a mere manipulation of mathematical constructs or abstract ideas (i.e., no physical transformation or reduction) in a *process* was unpatentable, while the machine was physical and definite (enough). The data structure was held abstract like the process.

(c) Third, in *Lowry*<sup>66</sup> the court held patentable a computer memory *article* containing a specific data structure; after finding *data structures impose a physical organization on the data . . . are specific electrical or magnetic structural elements in a memory . . . [and] are physical entities that provide increased efficiency in computer operation*, the court quoted its pre-*Benson Bernhart*<sup>67</sup> holding that programming a computer physically changes its memory elements, and held the PTO failed to show the data structure *lack[ed] a new and unobvious functional relationship with the memory*.<sup>68</sup> The court also held the *printed matter doctrine* is limited

---

claimed subject matter as a whole is a disembodied mathematical concept . . . which in essence represents nothing more than a "law of nature," "natural phenomenon," or "abstract idea."

*Id.*

65. *In re Warmerdam*, 33 F.3d 1354 (Fed. Cir. 1994).

66. *In re Lowry*, 32 F.3d 1579 (Fed. Cir. 1994).

67. *In re Bernhart*, 417 F.2d 1395, 1400 (C.C.P.A. 1969). "[I]f a machine is programmed in a certain new and unobvious way, it is physically different than the machine without that program: its memory elements are differently arranged." *Id.*

68. In *Lowry*, the court apparently misconstrued computer science by suggesting there may be a new and non-obvious *functional relationship between physical memory and a logical data structure*. But no *functional* relationship ever crosses the Cartesian divide. In *re Lowry*, 32 F.3d 1579 (Fed. Cir. 1994). Computer science uses an arbitrary, abstract (and quite simple) mathematical concept (developed by Alan Turing) and a mechanical device that conveniently allows a mapped correlation between physical *process* and mathematical *computation*. This correlation of *process* and *computation* occurs at the interface of the divide but never crosses it. No causal relationship (*ergo*, no *functional* relationship) ever crosses the Cartesian divide.

Further, computer science rests upon a *consistent* and unchanging *functional* relationship between physical memory and the digital signals stored upon it; significantly, physical organization or location of data symbols is *totally irrelevant* to the computation and to its hidden *practical use* meaning. So, computer science ingenuity is unavailable and unrecognizable, until that contextual meaning is made known and binds to the process as its utility limitation.

Digital process computers manipulate symbolic digits; no physical analog to a circumstance exists because a two position logic is mapped to an on/off circuit. Any novelty associated with *Lowry's* data structure is purely logical (i.e., in the abstract meaning of the sequential processing of binary numbers) not physical. *Lowry's* data structure may be in-

to human activity.

(d) Fourth, in *Trovato*<sup>69</sup> the court reviewed the algorithm—centric two step test of *Freeman-Walter-Abele* and the abstract idea manipulation test of *Warmerdam*, to hold unpatentable the *process* and *machine* claims of a logical system for calculating the shortest distance between two points in physical space; absent any described specific physical process or apparatus, the court held it a manipulation of abstract ideas. However, on July 25, 1995 an *en banc* court issued a *per curiam* order vacating its judgment, setting aside the PTO decision and remanding *Trovato* for further consideration in light of *Alappat* and the pending PTO Guidelines for computer inventions.<sup>70</sup>

#### G. TRANSITION OBSERVATION

Since *Benson*, we've sought to understand how a process algorithm relates to its computer implementation and application to *a function the patent laws were designed to protect*. But such efforts often only renewed uncertainty in (i) whether computer mechanization alone is sufficient to cover all uses, (ii) where between abstract algorithm and practical use, patentability attaches, and (iii) which practical uses were sufficient, which too much?<sup>71</sup>

*Media* claims for computer instruction fixed on a computer readable medium (our target theme) regards patenting *expression* versus *embodiment*; which is a question beyond the mathematical algorithm preemption issue or computer science versus natural science ingenuity raised by the *process* and *machine* patents discussed above. However, the *media*

---

novative, but its patent application is an attempt to establish ownership over use of an abstract logical construct.

69. *In re Trovato*, 42 F.3d 1376 (Fed. Cir. 1994).

70. *In re Trovato*, 60 F.3d 807 (Fed. Cir. 1995).

71. What is a function the patent laws were designed to protect? See *Alappat*, 33 F.3d at 1550-68; Chief Judge Archer (in dissent) suggests the majority's opinion will lead to a proliferation of *mathematical patents* couched in terms of *special purpose machines* that will be easy to obtain because there is *no prior art* to judge them:

One might invent or discover a new and useful product or process that includes as an element therein digital electronics performing mathematics . . . . One might invent or discover a mode of operation of a digital electronic device, capable ultimately of being used to perform mathematics . . . . Or, one might discover a particular mathematic operation and claim the use of digital electronics to perform the mathematical operation . . . . This last category, however, is at best newly discovered mathematics which is not being 'implement[ed] or applie[d] . . . in a structure or process which, when considered as a whole,' [Diehr citation] (emphasis added), represents an invention or discovery of a machine or process(as in the case of Diehr) for which one may obtain a patent pursuant to § 101.

*Id.*

The concern Judge Archer expresses in the last category is the absence of a utility limiting the mechanized computation to a particular *practical use*. *Id.*

claim proposal is a progeny of the algorithm struggle and understanding its origin puts its issue in context.

## V. ARE COMPUTER PROGRAMS ON MEDIA PATENTABLE?

### A. THE PTO PROPOSAL

Both *Lowry*<sup>72</sup> and *Beauregard*<sup>73</sup> sought article of manufacture patent claims for computer science ingenuity, but there their similarity both starts and ends. *Lowry* patented a computer component, a memory holding specific structured data. The court held the memory was a patentable machine component.<sup>74</sup> In *Lowry* the machine had been instructed, the algorithm executed and the result stored in memory for later access; *Lowry* was an *instructed machine* component, not mere *machine instruction*.

*Beauregard* sought an *article* claim for computer readable media containing computer instruction for filling a rectangle. It was appealed after *Lowry* but before *Lowry's* decision. Based on the ruling in *Lowry* the PTO conceded the *Beauregard*<sup>75</sup> appeal, disregarding the embodiment versus expression distinction altogether. Subsequent PTO *Guidelines* reversed its prior opinions<sup>76</sup> and concluded *patent and copyright*

72. *In re Lowry*, 32 F.3d 1579 (Fed. Cir. 1994) (holding data structured on a memory used by an application program was patentable).

73. See *In re Beauregard*, 53 F.3d 1583 (Fed. Cir. 1995).

74. *Lowry* allowed the computing machine claim with no further *practical use* limitation. *In re Lowry*, 32 F.3d 1579 (Fed. Cir. 1994). The lack of a *practical use* is troubling; absent it, *Lowry* suggests mere mechanization of a computing process is enough to preempt all subsequent mechanized use. *Id.* But that is the algorithm aspect of *Lowry* and for the *Beauregard* (article of manufacture) discussion it is more significant to note *Lowry* pertained to an *instructed machine* and not mere *machine instruction*. *Id.*

75. *In re Beauregard*, 53 F.3d 1583 (Fed. Cir. 1995). In summary, the Court of Appeals for the Federal Circuit (CAFC) granted a motion by the Commissioner of the U.S. Patent and Trademark Office (PTO) to dismiss an appeal from the Board of Patent Appeals and Interference's order rejecting computer product claims on the basis of the printed matter doctrine. *Id.* Because there was no case or controversy, the Court of Appeals for the Federal Courts vacated and remanded the case. *Id.* In his motion, Commissioner Lehman stated that "computer programs embodied in a tangible medium, such as floppy diskettes, are patentable subject matter under 35 U.S.C. § 101 and must be examined under §§ 102 and 103." The PTO memorialized its position in its advisory guidelines directed to patent examiners. See *Examination Guidelines for Computer-Related Inventions*, 61 Fed. Reg. 7,478 (1996). Note, there was no decision on the merits in *Beauregard*.

76. See *Guidelines to Examination of Programs*, 829 OFF. GAZ. PAT. & TRADEMARK OFFICE 865 (Aug. 16, 1966); *Guidelines to Examination of Applications for Patents on Computer Programs*, 855 OFF. GAZ. PAT. & TRADEMARK OFFICE 829 (Oct. 22, 1968); *Guidelines to Examination of Applications for Patents on Computer Programs*, 858 OFF. GAZ. PAT. & TRADEMARK OFFICE 18 (Jan. 7, 1969); and *Patentable Subject Matter: Mathematical Algorithms and Computer Programs*, 1106 OFF. GAZ. PAT. & TRADEMARK OFFICE 5 (Sept. 5, 1989).

*share authority over symbolic expression, by reasoning:*

Judicial denial of patent protection for scientific principle *and* abstract idea based on a preemption concern, is absolute for scientific principles *only*.

Neither a scientific principle nor an abstract idea is patentable as such; however, a practical use of either may be patentable.

So if a *practical use* of an abstract idea is patentable, then its disembodied instruction (expressed on a tangible media) is patentable, because patents provide control over the making of an invention and *functionally descriptive* computer instruction serves that purpose.<sup>77</sup>

That incredible shift in patent's paradigm, from *useful embodiment* to *symbolic expression*, conceded in *Beauregard*, *sub silentio*, and expressed in the PTO *Guidelines*, was done with *no mandate* of court or Congress. Yet it impacts the traditional patent—copyright<sup>78</sup> distinction dramatically and breaches the premise against exclusive property over mere abstract ideas.

The *Guidelines'* *media-article* patent gives *de jure* property over *all* expression instructing any computing device use of the algorithm. Since all alternative expression is precluded, the patent grant is essentially a

---

77. See Examination Guidelines for Computer Related Inventions, 61 Fed. Reg. 7478 (determining whether the claimed invention complies with 35 U.S.C. § 101 (1984) and the appended flow chart), state:

Courts have expressed a concern over 'preemption' of ideas, laws of nature or natural phenomena . . . . In fact, such concerns are only relevant to claiming scientific truth or principle. Thus, a claim to an 'abstract idea' is non-statutory because it does not represent a practical application of the idea, not because it would preempt the idea.

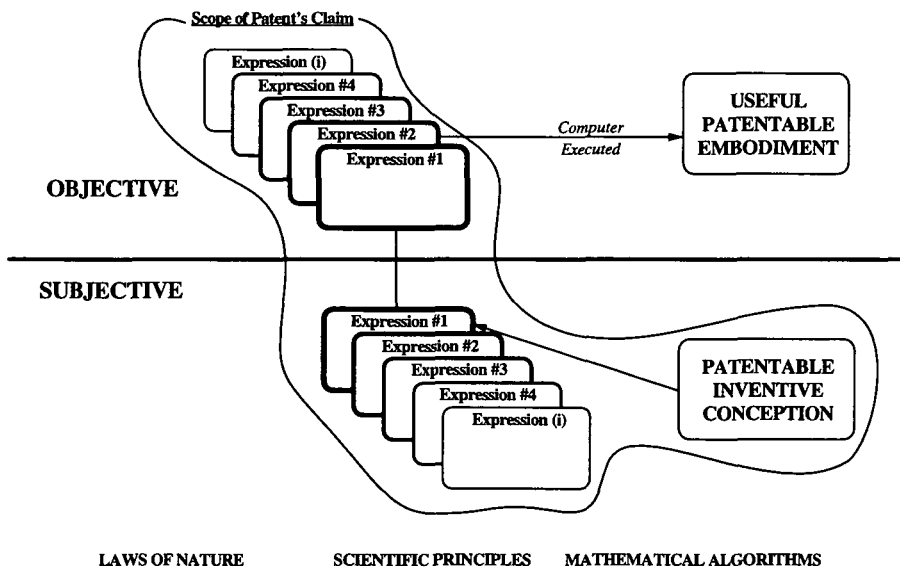
*Id.*

Thus, once a practical application is demonstrated (here by patenting a programmed computer or its process), it is permissible to separately patent the "functionally descriptive" machine instructions as an article of manufacture, if fixed in a computer readable medium.

78. This merger of copyright and patent principles in the *Guidelines* is recognized by at least one other commentator. See Ronald S. Laurie, *Patenting Content: The Expanding Role of Patent Protection for Internet-Based Information Products* 37, Bar Association of San Francisco Annual Computer Law Institute, San Francisco (July 14, 1996). See also Hollaar, *Justice Douglas Was Right: The Need for Congressional Action on Software Patents*, 24 AIPLA Q. J. 283 (1996). Professor Hollaar points out that the *Guidelines* incorrectly distinguish *functional descriptive material* from *non-functional descriptive material*. *Id.* Because repository memory cannot hold "functional" information, there is no logical difference between computer instruction and any other kind of data, including music. *Id.* (The *Guidelines* define "data structures and computer programs which impart functionality when encoded on a computer-readable medium" as statutory subject matter). See *Guidelines*, *supra* note 1. As noted by one commentator, software is merely a "set of instructions defining functions to be carried out by computer hardware . . . it can carry out no physical functions because it lacks any capacity to do so." See Amicus Curaie Brief, *In re Beauregard*, 53 F.3d 1583 (Fed. Cir. 1995).

property interest over the abstract idea itself.<sup>79</sup> Significantly, patenting such expression circumvents *both* copyright merger and patent preemption prohibitions! FIGURE 3 DISPLAYS THIS CARTESIAN VIEW.

Figure 3: COMPUTER INSTRUCTION PATENTS



Additionally, in *Beauregard* the claim does not appear limited to a *useful art*, even though directed to filling a polygon one line at a time.<sup>80</sup> A claim over all computing device use of an algorithm for a computer

79. Where an idea can effectively be expressed only in one way, the idea and expression are deemed to have *merged*, such that the entire work is held unprotectable. See *Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 446 F.2d 738, 742 (9th Cir. 1971) (idea and expression of bee-shaped pin indistinguishable).

80. See *In re Beauregard*, 53 F.3d 1583 (Fed. Cir. 1995) (emphasis added):

Representative Claim 22: An article of manufacture comprising: a computer usable medium having *computer readable* program code means embodied therein for causing a polygon having a boundary definable by a plurality of selectable pels on a graphics display to be filled, the *computer readable* program code means in said article of manufacture comprising:

*computer readable* program code means for causing a computer to effect, with respect to one boundary line at a time, a sequential traverse of said plurality of selectable pels of each respective boundary line;

*computer readable* program code means for causing the computer to store in an array during said traverse a value of an outer pel of said boundary of said plurality of selectable pels for each one of a plurality of scan lines of said polygon; and

operation, in effect, is a patent over that functionality if not limited to some *practical use*. *Benson* is unequivocal, if abstract logic is patentable, mechanization is necessary but not sufficient; to avoid preemption, the instructed machine must be limited to a *practical use*.

Many who struggle to protect computer science ingenuity believed the PTO got it right, at last.<sup>81</sup> But did it, or did the pendulum merely swing from too little to too far or for the wrong reason?

## B. EMBODIMENT VS. EXPRESSION

The United States Constitution, statutes and case law distinguish patentable invention from copyrightable writing.<sup>82</sup> Each pertain to abstract ideas but neither provides a possessory interest over ideas. Copyrights prevent copying of original *expressions* of ideas, fixed in a tangible media.<sup>83</sup> Patents prevent making, using or selling *useful embodiments* of inventive ideas.<sup>84</sup> Each is limited: copyright to symbolic expression and patent to utilitarian embodiment.

---

*computer readable program code means for causing a computer to draw a fill line, after said traverse, between said outer pels having said stored values, for each said one of said scan lines.*

*Id.*

So Beauregard's employer (IBM) patented every possible computation process, usable on any computing device, to fill any polygon in that manner; and it sought to get a patent on every possible computer instruction to do so.

81. See, e.g., Robert C. Laurensen, *Computer Software 'Article of Manufacture' Patents*, 77 J. PAT. & TRADEMARK OFF. SOC'Y 811 (1995); Victor Siber, *Narrowing the Gap Between Technology and the Law - Computer Program Related Inventions*, Presented at NCIPLA, Akron, Ohio (Sept. 13, 1997); Keith Stephens & John Sumner, *Software Objects: A New Trend in Programming Software Patents*, 12 COMPUTER & HIGH TECH. L. J. 1 (1996); D.C. Toedt, *Software as 'Machine DNA': Arguments for Patenting Useful Computer Disks Per Se*, 77 J. PAT. & TRADEMARK OFF. SOC'Y 275 (1995); Keith E. Witek, *Developing a Comprehensive Software Claim Drafting Strategy for U.S. Software Patents*, 11 BERKELEY TECH. L. J. 363 (1996).

82. U.S. CONST. art. 1, § 8, cl. 8 is a *balanced* sentence pertaining to the two distinct subjects of copyright and patents. Congress was granted (i) power to promote the progress of science by securing to authors the exclusive right to their writings; and (ii) power to promote the progress of *useful arts* by securing to inventors the exclusive right to their discoveries. See Karl B. Lutz, *Patents and Science: A Clarification of the Patent Clause of the US Constitution*, 32 J. PAT. & TRADEMARK OFF. SOC'Y 83 (1950). The clause was a *consolidation of two proposals, which got packaged together*. Giles S. Rich, *The Principles of Patentability*, 42 J. PAT. & TRADEMARK OFF. SOC'Y 75 (1960). The Copyright Act expressly provides copyright protection does not extend to *any idea, procedure, process, system, method of operation, concept, principle, or discovery*. 17 U.S.C. § 102(b) (1984). See also *Baker v. Selden*, 101 U.S. 99 (1879) (copyright subsists in a work's expression and not the underlying ideas). 35 U.S.C. § 101 (1984) (providing the statutory subject matter for patents: process, machine, manufacture, or composition of matter).

83. 17 U.S.C. § 102(a) (1984).

84. 35 U.S.C. § 271(a) (1984).

Computer programs instruct a Turing machine's computation performance. They are not the computation itself, that is, machine instruction is not the instructed machine, nor does it constitute any physical part or component of the actual computation, no more than a menu is a part of a meal, or a highway sign the destination. Software is disembodied symbolism distinct from the instructed use. It does not lose its abstract nature when fixed in a tangible medium. Its expression is properly copyrightable. Mere instruction (even if fixed in a medium) is not patentable, until reduced to a mechanization of the invention.

If Beauregard was first to invent using computer readable media to transport instruction, a patent may justly preclude all such media use for that purpose. But since the only difference he offered is a change in symbolic meaning or content, there is no advancement in the art of such media and he is properly limited to copyrighted expression, even if a machine so instructed were patentable. Patents pertain to embodiment, not expression.

Recording computer instruction upon *computer readable media* to instruct a computer is a skill now within the reach of any novice and involves no inventive faculty. Thus, *media article* claims lack patent merit. While concurrent *process, machine* and *article* patent claims are permissible,<sup>85</sup> patentability of each claim format is based upon its own merits.<sup>86</sup> The PTO *Guidelines* tacitly admit there is no invention in *media article* claim formats, by conditioning such claims upon their machine or process claims satisfying the utility, novelty and non-obviousness patent requirements.<sup>87</sup>

---

85. See *In re Tarczy-Hornoch*, 397 F.2d 856 (C.C.P.A. 1968) (holding a process claim otherwise patentable should not be rejected merely because the application discloses apparatus which will carry out the recited steps, whether or not the apparatus is separately patentable).

86. See *Nestle-Le Mur Co. v. Eugene. Ltd.*, 55 F.2d 854, 856-57 (6th Cir. 1932) (and cases cited therein), where the court held:

The question of law thus presented may perhaps be stated as follows: Where one discovers a new and useful process for accomplishing a given result, is the obvious mechanical or electrical device, obvious to anyone to whom the proposed method is disclosed, patentable apart from the process? We are constrained to the opinion that it is not.

*Id.*

See also *Whitman v. Andrus*, 194 F.2d 270, 273, citing *Nestle-Le Mur*; *Guidelines to Examination of Applications for Patents on Computer Programs*, 855 OFF. GAZ. PAT. & TRADEMARK OFFICE 829 (Oct. 22, 1968); *Guidelines to Examination of Applications for Patents on Computer Programs*, 858 OFF. GAZ. PAT. & TRADEMARK OFFICE 18 (Jan. 7, 1969), citing the *Nestle-Le Mur* and *Whitman* opinions.

87. See Examination Guidelines for Computer Related Inventions, 61 Fed. Reg. 7478 (determining whether the claimed invention complies with 35 U.S.C. § 101 (1984) and the appended flow chart).

The PTO recites the often quoted talisman that *anything under the sun, made by man* is patentable, but the PTO includes within it, the *made invention and any useful instruction on its making*,<sup>88</sup> without expressing concern over either patent preemption or copyright merger doctrines. But instruction is not the *thing made* nor does invention exist *under the sun* until instruction is executed. Case law is in accord with this embodiment constraint.<sup>89</sup>

Most significantly, the PTO *Guidelines* lack any juridical explanation or citation to support removing *practical abstract ideas* from within the scope of preemption protection. There is no premise supporting the PTO's distinction between scientific principle and abstract idea. Indeed, the Supreme Court uses either and *both* in referring to mathematical algorithms.<sup>90</sup>

### C. THE CAMSHAFT FALLACY

Some argue computer instruction fixed in a medium is similar to a machine camshaft,<sup>91</sup> but that is an inapt analogy for patent law analysis. A camshaft embodies a necessary<sup>92</sup> mechanical element of a machine. Its patentability rests upon the functionality provided by its physical manifestation; that is, the application of natural science. Contrariwise, computer instruction is symbolic language that must be interpreted, translated or read and reduced to practice before any innovated functionality can show up.<sup>93</sup> When instructed, a digital computer may be an alternative to a determined (camshaft) machine, but computer in-

---

88. Examination Guidelines for Computer Related Inventions, 61 Fed. Reg. 7478, 7481 (1996).

89. Even the recent Court of Appeals for the Federal Circuit decisions require *physicality* by sufficient physical *structure, embodiment, or activity*. See, e.g., *In re Warmerdam*, 33 F.3d 1354, 1360 (Fed. Cir. 1994) (method for generating a data structure nonpatentable because *only practical, (sic) embodiment* of the claimed method steps was purely mathematical); *In re Alappat*, 33 F.3d 1526, 1544 (Fed. Cir. 1994) (claims recite specific machine, not a *disembodied mathematical concept*); *In re Schrader*, 22 F.3d 290, 292-93 (Fed. Cir. 1994) (claimed process steps must be applied to or limited by physical elements or process steps).

90. In *Gottschalk*, 409 U.S. at 71 (the mathematical algorithm was analogized to an abstract idea; *Flook* analogized the mathematical formula to a phenomenon of nature; while *Diamond v. Diehr*, 450 U.S. 175 (1972) said it was like a law of nature).

91. See e.g., Victor Siber & Marilyn Smith Dawkins, *Claiming Computer-Related Inventions as Articles of Manufacture*, 35 IDEA 13, 21 (1994).

92. Note, unlike a camshaft, computer instruction fixed in a computer readable medium is removable from the machine/computer. The computer readable medium is merely a transport carrying instruction to the machine.

93. Computer instruction expresses how to make an invention and is not yet any part of the machine's physical manifestation; it must be *read* and *executed* before becoming operative. The PTO acknowledges the expressive nature of computer instruction by distinguishing between *functional descriptive material* and *non-functional descriptive material*.



struction is always disembodied expression compared to a camshaft embodiment.<sup>94</sup>

#### D. THE CALL FOR JUSTICE

The only justification offered for *media* claims is given by the computer hardware industry; that is, the ease of their enforcement against the software industry.<sup>95</sup> Calling for Justice, they say they merely seek a direct action against such programmers because they *enable* infringement. But patentees can already sue programmers for inducement or contributory infringement if the computer use of the accused programs infringe a patent. Inducement<sup>96</sup> and contributory<sup>97</sup> infringement do require an infringing *use* of the accused software, but otherwise they are essentially the same as a direct infringement action.

Creating a direct infringement action against program instruction writing is both unnecessary and unjustified. Such a change would shift the patent paradigm from embodiment to expression, placing it adrift in the murky waters of abstraction and should be avoided. Extending patent's scope into the abstract domain of expression is inconsistent with our juridical fundamentals and (as shown below) unjustly enrich the hardware industry over independent software development.

#### E. MEDIA-ARTICLE CLAIMS AND THE SOFTWARE INDUSTRY

Hardware manufacturers already license their patented *uses* to their, and their cross licensees, customers. A direct infringement action against software developers would allow their demand of a *second* payment for a use already licensed. It also threatens the software industry's independence.

---

*See* Examination Guidelines for Computer Related Inventions, 61 Fed. Reg. at 7,481 (1996).

94. *In re Lowry*, 32 F.3d 1579 (Fed. Cir. 1994) (making an apparent distinction). The claims in *Lowry* recited a memory for storing a data structure for use by a computer application. *Id.* at 1581. The court held the claims statutory after finding the requisite structure was present (the electronic structural arrangement of the data elements in the memory constituted the result of the executed instruction). *Id.* Compare with *In re Warmerdam*, 33 F.3d 1354 (Fed. Cir. 1994) (where a claim to a data structure *per se* was held non-statutory). *Beauregard* claims recite the media for physical structure. However, the media does not contribute any physicality to the instructed machine, nor is there any invented functional relationship between the computer readable media and the instruction. *Beauregard* articles serve only to communicate instruction to a computer, that is, they are "read."

95. *See, e.g.*, Victor Siber & Marilyn Smith Dawkins, *Claiming Computer-Related Inventions as Articles of Manufacture*, 35 IDEA 13, 21 (1994); Amicus Curiae Briefs submitted in *In re Beauregard*, 53 F.3d 1583 (Fed. Cir. 1995).

96. 35 U.S.C. § 271(b) (1984).

97. 35 U.S.C. § 271(c) (1984).

### 1. *Common Law Inherency, Cross Licensing & Double Payment*

When a manufacturer sells a computer containing all the physical elements of a process patented by the manufacturer, there is no patent infringement if a purchaser so used it, and so no inducement or contributory infringement in providing instruction. A computer purchaser acquires the seller's entire right to enjoy all the beneficial use of the process computing capability *inherent* in the computer design and characteristics, *except* those expressly withheld.

Since a patent provides only the right to interfere with the use of others' stuff and a seller transfers all right and interest in the sold item, a seller cannot later object to a purchaser's use of the machine as sold, here a digital process computer capable of performing any computation algorithm. This *doctrine of inherency* estopps sellers from later challenging their purchaser's title over the goods sold.<sup>98</sup> Such a seller would also be estopped from asserting *later* obtained patents against the sold device; simply put, you cannot take back what was granted for consideration.<sup>99</sup>

Some argue, sale of a general purpose computer does not include patented uses unless the computer has no other beneficial use, or the seller knew of the intended use. Those defenses however apply to where something is added or the stuff sold is physically reconstituted before infringement is found. Neither support a seller derogating title over use

---

98. The doctrine of inherency is a part of every transfer of property interest or sale of goods. See *Sparks v. Hess*, 15 Cal. 186, 196 (1860) (quoting *Whitney v. Olney*, 29 Fed. Cas. 1074 (1831)). "[T]he good sense of the doctrine on this subject is, that under the grant of a thing, whatever is parcel of it, or of the essence of it, or necessary to its beneficial use and enjoyment, or in common intendment is included in it, passes to the grantee." See also *Trask v. Moore*, 24 Cal. 2d 365 (1944). "As a general rule, a conveyance of property carries with it by implication all incidents rightfully belonging to, and essential to the full enjoyment of, such property at the time of conveyance." *Id.* Thus, when a device is sold, all the seller's right, title and interest over its inherent (known or unknown) incidents and beneficial uses are transferred to the purchaser, except those expressly reserved. (Compare bailment, where title remains in the bailor and the bailee may use the thing only as expressly permitted (any other use being a conversion or trespass of the bailor's property)).

The doctrine of patent exhaustion is a distinct patent law perspective, getting to the same place. See HARMON, PATENTS AND THE FEDERAL CIRCUIT 210 (3d ed. 1994). "The first authorized sale of an article embodying a patented invention exhausts the patent rights in that article. In other words, the purchaser of a patented invention has the right to use it . . . without interference from the patentee . . . . A purchaser . . . is free to use or resell the products." *Id.* Any unreserved patent right of the seller to preclude a beneficial use of a computer's inherent design and structure is both exhausted and inherent in title to the sold device. *Id.*

99. See *AMP, Inc. v. United States*, 389 F.2d 448 (C.C.P.A. 1968): "[W]here the owner of a patent grants to the licensee the right to use a patented machine, the grant carries with it, by necessary implication, a license under any other patent of the licensor which would be infringed by operation under the grant." *Id.*

of the inherent computation capability of a general purpose computer. If the computer is only a part of an infringed patent then this defense is unavailable, but so long as the accused infringement is physically inherent, the seller can not derogate the title.

Additionally, hardware manufacturers typically cross license their patents and include a net royalty payment based on relative sales and patent portfolios.<sup>100</sup> Such arrangements typically include each entity's *entire patent portfolio* and grant rights "to make, use, lease, sell and otherwise transfer Licensed Products and to practice any method or process involved in the manufacture or use thereof."<sup>101</sup> A purchaser of such a licensee may practice those licensed uses free from any cross licensor interference.<sup>102</sup>

If hardware manufacturers receive compensation for their patents in their sales and in cross licensing, are they entitled to a *second* compensation from the programmers who enable the licensed use, where only *one embodiment* occurs? Such instructions do not cause patent infringement; indeed, they enable licensed use. Patenting mere instruction provides an unjustified windfall to hardware producers; *two* payments for one invention embodiment.

## 2. Independent Software Development

Computer program *media* patent claims threaten independent software ingenuity and development. Hardware producers encouraged independent instruction when machine sales mattered. Increased importance of patents and software ingenuity has the hardware industry seeking leverage over independent software development.<sup>103</sup> But computer sales and cross licensing may prevent an inducement or contributory infringement claim.

---

100. A company like IBM, for example, enters such cross-licenses and receives valuable royalty (now over \$1 billion annually, nearly doubled from 3 years ago), as well as valuable cross-licenses, without providing any product or service. Clearly it could not later challenge the licensed customer's use.

101. Quoted from the grant clause of IBM's standard cross-license agreement (emphasis added).

102. See *Bloomer v. Millinger*, 68 U.S. 340 (1864) (the law on this subject has been known for over 100 years).

[When a patentee] has himself constructed a machine and sold it, or authorized another to construct and sell it, or to construct and use and operate it, and the consideration has been paid to him for the right, he has to that extent parted with his monopoly, and ceased to have any interest whatever in the machine so sold or so authorized to be constructed and operated.

*Id.*

103. See Ira Sager, *Big Blue is Out to Collar Software Scofflaws*, Bus. Wk., Mar. 17, 1997, at 34.

By making computer instruction on any computer readable medium a direct infringement, the hardware industry can demand a *second* payment as well as cross licenses to software industry patented technology, while giving up nothing (it hadn't already sold) in return. They will likely get the licenses to the software industry technology and a payment for their trouble, if they get such *media-article* patent claims.

But once established, when will their demand be satiated? How much payment will they seek? There is no limit to what a patentee may charge, so long as all takers are treated equally. In such a squeeze, will independent software development survive?

## VI. CONCLUSION

Applying a scientific principle causes physical change while processing an algorithm does not. Computer science modeling is an infinitely flexible abstract principle whose only use or manifestation is within a man-made *objective* device; while a scientific principle is a rigid concept, fixed by the natural workings of *objective* reality. In the world of physical science, we discover a principle and apply it to manipulate circumstance to a *useful* end, the principle remaining available for other uses. In the digital world, we discern a specific logical solution and mathematical truth to a problem in symbol manipulation leaving the algorithm available for other uses.

Digitization is the principle of flexibility, in the form of a flexible principle; a man-made abstract construct that *only* manipulates symbols, but can do so at incredible speeds, because it is so tightly bound in a highly structured, very formal, mathematical process. The machine is purely incidental and completely irrelevant. It's worth repeating, computation is conducted as accurately with beer cans and ping pong balls as with CPUs and memories. Respectfully we note scientific principle is primarily concerned with the mechanization of CPUs and memories, or beer cans and ping pong balls; yet, where computer science resides, neither one effects the process at all. It doesn't matter to the meaning of abstract calculation.

Innovation in the natural sciences is in the thought of applying immutable scientific principles to things, while computer science ingenuity is in the thought of applying mutable logic to mathematical symbols. In a sense, process computing is a *subjective* mutable principle, seeking an *objective practical use*. The two ingenuities may be brought together under one protection mechanism, patents.<sup>104</sup> Though there is likely at-

---

104. *Gottschalk v. Benson*, 409 U.S. 63 (1972) (patent law always presents opportunities, hear Justice Douglas' pointer: "It is argued that a process patent must be tied to a particular machine or apparatus or must operate to change articles or materials to a 'differ-

tendant uncertainty and change under any such arrangement, that is, we need to develop a further jurisprudence of the *utility* requirement.

If *objective* novelty in the application of natural science were required, then computer science ingenuity in hastening a known structure or process to a known result or in enhancing mere post-solution activity would be unpatentable. The juridical patent issue we face is do we give computer science ingenuity parity to natural science ingenuity and if so, how? Numerous powerful arguments attend both sides of that consideration and they should all be thoroughly thought through, no doubt; but the proposed *Beauregard, media-article* patent claims on expression is an unnecessary, misguided and unjustified breach of the utilitarian embodiment limitation. There is no *reason* for such patents. They will only confound Justice and the independence of the software industry for decades.

---

ent state or thing.' We do not hold that no process could ever qualify if it did not meet the requirements of our prior precedents." *Id.* at 71.

But while offering to remain open, Douglas closed noting: "If these programs are to be patentable, considerable problems are raised which only committees of Congress can manage . . . . The technological problems tendered in the many briefs before us indicate to us that considered action by the Congress is needed." *Id.* at 73.