AIPLA QUARTERLY JOURNAL

VOLUME 31, NUMBER 3 Page 273 SUMMER 2003

COGNITIVE NEUROSCIENCE AS A MODEL FOR NEURAL SOFTWARE PATENT EXAMINATION

Joseph S. Bird[*]

 Mr. Bird is a partner at Bradley Arant Rose & White LLP. He has a J.D., is admitted to the Patent Bar, and expects an M.S. in Vision Science in 2004. The opinions stated herein are those solely of the author, and not of Bradley Arant Rose & White LLP or of its clients.

I. INTRODUCTION

The Patent Office's ("PTO") internal classification system categorizes inventions mostly by structure, in the form of articles or products.[1] This system is sufficient for many inventions but not for software performing functions analogous to the central nervous system ("CNS"). Neural software, also called neural computation or artificial intelligence ("AI"), is written to perform CNS functions and should be categorized using patent classifications to be developed based upon CNS functions which are observed in cognitive neuroscience research. These functions *solve problems* and thus enable survival of the animal in its environment.

Conventional software performs functions analogous to those of machines or real world objects. Neural software performs steps based upon analogies to mental steps.[2] Conventional software patents already suffer from a high level of abstraction. Neural software patents inherit all the existing difficulties of other software patents and, in addition, suffer effects from an even higher level of abstraction.

Problem-solving categories would focus on the underlying functions involved rather than on the specific invention's application. The problem-solving approach is already found in applicable case law governing patent validity.[3] The first step in using new classifications would be to resolve whether a software invention is analogous either to a physical machine, implementing steps from rules, or to a process, like CNS functions.

---

1 *See generally* U.S. PATENT & TRADEMARK OFFICE, PATENT CLASSIFICATION WEB SITE, *available at* http://www.uspto.gov/web/patents/classification/ (last visited June 22, 2003).

2 *See generally* RANDALL C. O'REILLY & YUKO MUNAKATA, COMPUTATIONAL EXPLORATIONS IN COGNITIVE NEUROSCIENCE: UNDERSTANDING THE MIND BY SIMULATING THE BRAIN (MIT Press 2000).

3 *See, e.g.*, Shatterproof Glass Corp. v. Libbey-Owens Ford Co., 758 F.2d 613, 620, 225 U.S.P.Q. (BNA) 634, 638 (Fed. Cir. 1985).

Next, the examiner would identify CNS functions in the claims and then assess novelty, obviousness, written description, enablement, and best mode in light of those functions. CNS analogs, used alongside existing classifications, would improve examination and provide better disclosure of the contents of neural software patents.

II. THE FUNCTION OF A MODEL

Although he was not a lawyer—rather, perhaps, because he was not a lawyer—Allen Newell, a computer scientist, put his finger on the inadequacy of the models for patenting software.[4] According to Newell, case law holding that an algorithm was unpatentable per se[5] reflected a misconception that algorithms are solely mathematical equations: in truth, an "algorithm is an unambiguous specification of a conditional sequence of steps or operations for solving a class of problems."[6] More than fifteen years ago Newell identified the confusion at the core of the issue of software patentability: "The confusions that bedevil algorithms and patentability arise from the basic conceptual models that we use to think about algorithms and their use. That is why I have entitled my remarks, 'The Models are Broken, the Models are Broken.'"[7] Implicit in Newell's remarks was that patent law's problem with software stems from the intangibility of the subject matter. As a result of the intangible nature of software, more tangible models to classify it would be helpful.

4 Allen Newell, *Response: The Models are Broken, the Models are Broken*, 47 U. PITT. L. REV. 1023 (1986).

5 *Id.* (citing Gottschalk v. Benson, 409 U.S. 63, 175 U.S.P.Q. (BNA) 673 (1972)).

6 *Id.* at 1024.

7 *Id.* at 1023.

Humans appear to have used models and analogies throughout history as means for explaining the existence and function of the world. A model may also be described as a metaphor, in the sense that it can be considered a pattern or exemplar.[8] As one scholar stated, "it is clear from many studies of the cognitive process generally, and particularly of creative thought, that the act of thought in its more intense phases is often inseparable from metaphor."[9] Legal theory, as one of the most abstract of human endeavors, needs the most appropriate models in order to be of the most use to society. Patent law especially, as one of the more abstract areas of law covering intangible rights, needs appropriate models when grappling with intangible neural software.

Understanding the problems presented by neural software patents, though, requires looking to the origins of software. The first computers were mechanical devices which tabulated information embedded in physical media such as punch cards. Software was a new, intangible means for calculating information which had been tabulated previously by mechanical means.[10] The computer acquired through software the ability to run its operations in lieu of physical assistance formerly supplied by external agency. Software originated as intangible instructions supplanting external, physical control of mechanical computers.[11]

From software's origins as a substitute for physical agency in running computer machinery, a very machine-like thing occurred. The content of such conventional software became a series of pre-programmed

---

[8] WEBSTER'S NINTH NEW COLLEGIATE DICTIONARY 762 (1987).

[9] ROBERT A. NISBET, SOCIAL CHANGE AND HISTORY; ASPECTS OF THE WESTERN THEORY OF DEVELOPMENT 5 (Oxford University Press 1969).

[10] GREGORY A. STOBBS, SOFTWARE PATENTS § 1.03 (Aspen Law & Business 2000) (citing CHARLES J. BASHE, ET AL., IBM'S EARLY COMPUTERS (MIT Press 1986)).

[11] *Id.*

"if-then" steps or rules which cannot perform differently than the pre-determined rules.[12] Most software currently created is "conventional" in this sense.[13] Conventional software does not adapt to situations not provided for by the designer, nor can it recognize patterns with partially degraded features. Conventional software is an intangible machine which operates within a tangible computer. Machines are objects constructed by humans as tools, and conventional software meets that definition in a new way by automating functions which previously required physical action. Software originated when little was known about cognitive neuroscience and, until the last twenty to twenty-five years, computer scientists made no effort to design software based upon principles obtained through cognitive neuroscience research. Recent advances in cognitive neuroscience, however, have spawned numerous efforts to design neural software which mimics the mental operations of the CNS. The intangibility of neural software makes it especially difficult to classify. A new set of categories would help the PTO and practitioners keep track of important similarities and distinctions among such inventions.

III. THE IMPORTANCE OF COGNITIVE NEUROSCIENCE

As the many CNS functions are gradually being deciphered, advances in software attempt to mimic these individual functions. Neuro-mimetic software is proliferating and is an area of huge potential growth of inventive activity. Applying traditional patent law elements to neural software presents challenges, but also new opportunities to cope with

---

12 John McCarthy & Patrick Hayes, *Some Philosophical Problems from the Standpoint of Artificial Intelligence*, *in* 4 MACHINE INTELLIGENCE 463 (Bernard Meltzer & Donald Michie eds., 1969); *see also* John Searle, *Minds, Brains and Programs, in* PHILOSOPHY OF ARTIFICIAL INTELLIGENCE (Margaret A. Boden ed., 1990).

13 McCarthy & Hayes, *supra* note 12.

the onslaught of such inventions. And where would one look for better ways to assess such new neuro-mimetic inventions? The CNS itself provides guidance for software inspired by the CNS. Parts III.A to III.C introduce cognitive neuroscience which supports the proposal in part IV that neural software should be classified according to the CNS functions which inspire the software.[14]

A. Synapses and Spikes

The most basic function of the CNS is to receive stimuli from inside and outside the body. The CNS includes the processing centers of the familiar sensory systems of vision, hearing, touch, smell, and taste. Much of the brain is devoted to finding the most useful bits in the avalanche of sensory information.

Neurons are CNS cells. They communicate information to one another at connections known as synapses. Synapses can be electrical or chemical.

At a chemical synapse, a neuron sends a message to another neuron. In the first, or "presynaptic" neuron, a vesicle containing a chemical neurotransmitter fuses with the cell membrane and the neurotransmitter is released into the tiny space between neurons, known as the "interstitial space." These neurotransmitters attach to receptors on the second, or "post-synaptic" neuron, which causes a change in the cell membrane and allows ions to flow from the interstitial space into the post-synaptic cell.

The rapid movement of ions changes the electrical potential inside the post-synaptic neuron. Neurons can depolarize from their normal, slightly negative, voltage of approximately 70 millivolts to a slightly positive

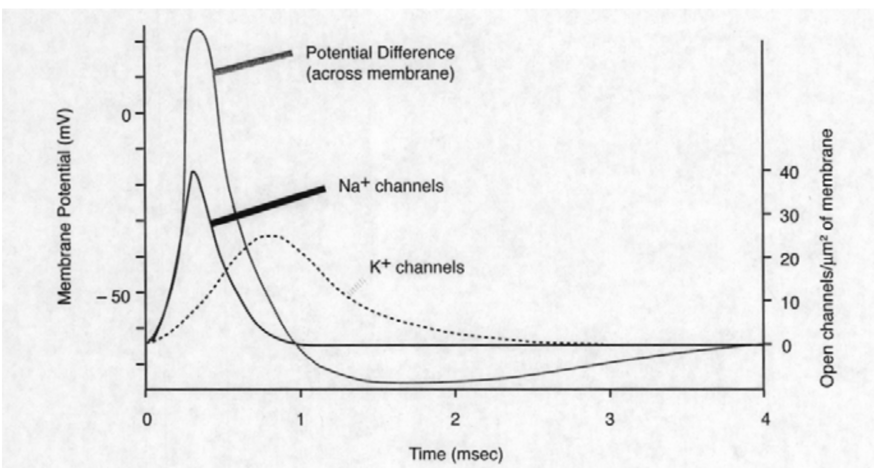
---

[14] *See generally* CHRISTOF KOCH, BIOPHYSICS OF COMPUTATION: INFORMATION PROCESSING IN SINGLE NEURONS (Oxford University Press 1999).

value. When enough positively charged ions flow into a neuron to raise the electrical potential to its "threshold," there is another sudden influx of ions.

An influx of positively charged sodium (Na^+), potassium (K^+), or calcium (Ca^{2+}) ions produces an excitatory post-synaptic potential ("EPSP"). Alternatively, an influx of negatively charged chloride (Cl^-) ions produces an inhibitory post-synaptic potential ("IPSP"), hyperpolarizing the neuron and making it more resistant to depolarization.

This polarization is rapidly transmitted from the neuron's "soma," or body, which contains the cell's nucleus, down the axon. This process, shown in the graph below,[15] is called an action potential, or "spike," and is a unit in the language by which neurons communicate:



15 DAVID WHIKEHART, BIOCHEMISTRY OF THE EYE 152 fig.7-1 (Butterworth-Heinemann 1994).

In an axonal spike, the rapid wave of depolarizing current travels down the axon membrane and causes neurotransmitters to be released into the next post-synaptic space.

The spike is an all-or-nothing event; chemical neurotransmitters are typically only released if there's a spike in the pre-synaptic neuronal axon. Thus, the spike creates a sort of binary code of "off" (not depolarized) or "on" (depolarized).

The other form of connectivity between neurons is an electrical synapse. A direct opening between two neurons, known as a gap junction, allows for the actual transfer of ions from one cell to another. The amount of ion flow through an electrical synapse depends on the concentration of ions near the pre-synaptic cell. Electrical synapses are not binary but, rather, graded like analog circuits and thus have properties different from chemical synapses. The interrelationship between chemical and electrical synapses is important to the dynamics of a system.

As a result of its architecture, an emergent property of the CNS is its plasticity, the ability to adapt to changing stimuli and environments. One of the seminal insights into plasticity was Donald Hebb's postulation, more than fifty years ago, that "[w]hen an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."[16] Increased connectivity between synapses is believed to be synonymous with learning and is often called long-term potentiation ("LTP"); a decrease in connectivity is called long-term depression ("LTD"). How plasticity, LTP and LTD, is implemented synaptically is a major question. Is it accomplished by a change in the number and distribution of neurotransmitter vesicle release

---

16 DONALD OLDING HEBB, ORGANIZATION OF BEHAVIOR: A NEUROPSYCHOLOGICAL THEORY (Wiley 1949).

sites, by an increase in the probability of vesicle release, or by some other method?[17]

B. Computational Neuroscience

Computational neuroscience, the study of CNS information processing, bears no relation at all to the workings of a digital computer. The architecture of the CNS and a digital computer are distinct in that they work on different principles in different media. However, digital computers are useful because they are extremely fast and software can be written which mimics CNS architecture, so that CNS computation can be simulated on a digital computer.[18]

As with any very complex problem, there are differing hypothetical solutions for how the CNS encodes information. One theory, known as "rate coding," hypothesizes that information is encoded according to the rate at which spikes occur. "[T]he number of spikes in a fixed time window

---

[17] It was originally believed that spikes traveled only from the soma down the axon, but more recent findings show that spikes also originate in the dendrites. These findings have many implications for synaptic plasticity. *See, e.g.*, KOCH, *supra* note 14 at 428-51; Bartlett W. Mel, *Information Processing in Dendritic Trees*, 6 NEURAL COMPUTATION 1031 (1994). For implications in synaptic plasticity, see, for example, Rajesh P.N. Rao & Terrence J. Sejnowski, *Spike-Timing Dependent Hebbian Plasticity as Temporal Difference Learning*, 13 NEURAL COMPUTATION 2221 (2001).

[18] Analog computing is also an important technology, and lends itself well to neural computation, in that analog functions more closely represent functions within the CNS than do digital computing functions. *See* ANALOG VLSI: CIRCUITS AND PRINCIPLES at xvii, xix, 1-3 (Shih-Chii Liu et. al. eds., MIT Press 2002)

following the onset of a static stimulus represents the intensity of that stimulus."[19]

An assumption underlying the theory of rate coding is that the rate of spikes is determined by the sum of the pre-synaptic inputs being relayed to the post-synaptic cell, where a "switch" determines whether the sum of the stimulation is sufficient to cause the post-synaptic neuron to spike. This is the "integrate-and-fire" assumption, under which the total EPSPs and IPSPs in a certain time period are summed, or integrated, to determine whether or not a spike will occur.[20]

Another important model of synaptic activity is "coincidence detection" which postulates that information is encoded because of the precise timing of pre-synaptic and post-synaptic activity. Synchronous firing causes encoding.[21] Thus, according to coincidence detection, individual EPSPs, rather than large numbers of them, cause information to be encoded.

Coincidence detection is supported by the observation that rate coding is essentially impossible at the very high speeds in which some

---

[19] FRED RIEKE, ET AL., SPIKES: EXPLORING THE NEURAL CODE 7 (MIT Press 1997).

[20] This theory was originally advocated by SIR CHARLES SHERRINGTON, INTEGRATIVE ACTION OF THE NERVOUS SYSTEM (C. Scribner's Sons 1906). It has been elaborated by numerous scientists. *See, e.g.,* Michael Shadlen & William T. Newsome, *Noise, Neural Codes, and Cortical Organization*, 4 CURRENT OPINIONS IN NEUROBIOLOGY 569 (1994).

[21] *See* MOSHE ABELES, CORTICONICS: NEURAL CIRCUITS OF THE CEREBRAL CORTEX (Cambridge University Press 1991); Peter Konig et al., *Integrator or Coincidence Detector? The Role of the Cortical Neuron Revisited*, 19 TRENDS IN NEUROSCIENCES 130 (1996).

sensory information is processed.[22] There are substantial differences between the theories of rate coding and coincidence detection. One theory or some combination of both may explain the experimental data. It is possible that different CNS functions use rate coding and coincidence detection in intricate combinations in order to perform the phenomenal feats of the CNS.

C. CNS Functions

The CNS is a massively parallel network. There are innumerable "hard-wired" connections dedicated to performing functions necessary for survival, and these circuits perform their functions at speeds breathtaking by current digital computer standards.

The patterns of connections in the CNS suggest a broad outline of how CNS functions are implemented. For example, the general pattern of connections between Area V1, the primary entrance of visual information into the cortex, to Areas V2, MT, MST and others, suggests that visual images are generally deconstructed by numerous areas of the brain dedicated to various kinds of features or motion.

A viewer can easily recognize a familiar face in approximately one second. The maximum speed of most neurons is about 100 spikes per second. This gives rise to the "100 step rule" which postulates that most CNS functions can be performed in 100 steps or less.

Thus, the CNS makes up for its lack of speed, compared to digital computers, with massively parallel connections, which allow dedicated circuits to perform functions almost instantaneously. The number and complexity of these parallel circuits is shown by the fact that the CNS contains perhaps 100 billion neurons, each of which synapses onto at least several and possibly thousands of neurons. The functions of the CNS can

---

[22] *See, e.g.*, KONIG ET AL., *supra* note 21.

be identified and discussed as properties of the neurons and neuronal assemblies themselves.

As an example, the most important CNS sensory function is vision. A list of CNS vision-related functions includes:

Contrast Sensitivity	Feature Binding
Lightness Constancy	Depth Perception
Image Stabilization	Motion Perception
Edge Detection	Contextual Modulation
Spatial Acuity	Attention
Detection Acuity	Vestibulo-Ocular Reflex
Localization Acuity	Saccade
Resolution Acuity	Fixation
Color	Visual Memory
Orientation Tuning	Vergence

Seeing a colorful airplane in flight across the sky requires collaboration of many CNS functions, including image stabilization, edge detection, motion perception, orientation tuning, and rapid eye movements, called saccades. Each of the CNS visual functions represents an important area of research in visual neuroscience, and there are many other sensory, motor, and cognitive functions under investigation. Each function represents a field for the advance of cognitive neuroscience and, eventually, neural computation.

D. *Neural Software*

Neural computation is a broad label applied to efforts to mimic the synaptic action of CNS functions through software. Computer scientists and others work from neurobiological data on rate coding and coincidence detection to design digital software which seeks to perform biologically observed functions. This field is as diverse and difficult to define as the CNS itself.

The potential applications for neural computation are infinite. For a robot, however, a task as simple as recognizing an obstacle in a pathway and avoiding it has proven daunting.[23]

A notable cluster of theories is called connectionism, or parallel distributed processing ("PDP"). The basic components of connectionism are neural networks or neural nets: software implemented algorithms modeled after biophysically observed properties of neurons. Processing units or nodes transform inputs (pre-synaptic) into outputs (post-synaptic) which represent neural information processing. A node in a neural net is an approximation of a synaptic alteration (weighting) of pre-synaptic spikes. Each node has an activation value which corresponds roughly to the strength of the hypothesis that what that unit stands for is present in the perceptual input. Nodes are connected and transmit information to one another by implementing neural concepts, such as LTP and LTD. Thus, mutually consistent nodes tend to excite one another, and mutually inconsistent nodes tend to inhibit one another.[24] The patterns of information are not stored in these PDP models. Rather, what is stored is the connection

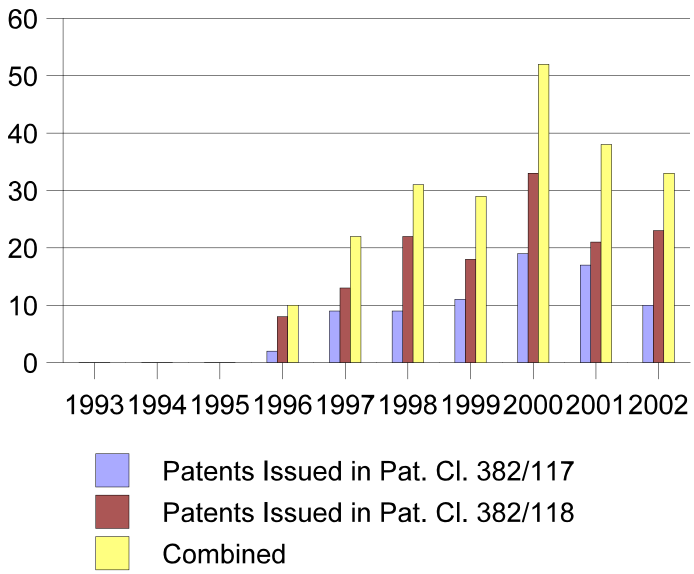
---

[23] Jeffrey Dean, *Animats and What They Can Tell Us*, 2 TRENDS IN COGNITIVE SCIENCES 60, 62 (1998).

[24] DAVID E. RUMELHART & JAMES L. MCCLELLAND, PARALLEL DISTRIBUTED PROCESSING: EXPLORATIONS IN THE MICROSTRUCTURE OF COGNITION 21 (MIT Press 1986); *see also* Alan F. Murray, *Pulse-Based Computation in VLSI Neural Networks, in* PULSED NEURAL NETWORKS 87 (Wolfgang Maas & Christopher M. Bishop eds., 1999).

strengths between the nodes, so that a representation of a real world object is comprised of a group of connections between nodes.[25]

Neural computation is a rapidly expanding field. Patent applications for neural computation inventions are proliferating and, given the subject's complexity and intangibility, the field could become an unmanageable area of patents without careful planning. A review of patents issued by year[26] in relevant patent classifications 382/117 and 382/118, "Image Analysis, Using a Characteristic of the Eye" or "Using a Facial Characteristic", respectively, shows a large increase in the numbers of patents issued beginning in 1993:



[25] RUMELHART & MCCLELLAND, *supra* note 24, at 31.

[26] Figures are from the PTO's web database. Figures for 2002 are annualized based on data through February 2002. *See* http://www.uspto.gov.

These data represent a marked increase in inventive activity in computer vision, and presage a much larger increase in such activity. What follows is a suggestion for how the PTO can better cope with the rising tide of patent applications in neural computation.

IV. COGNITIVE NEUROSCIENCE AS A MODEL FOR NEURAL SOFTWARE PATENT EXAMINATION

Software, standing alone, first won unequivocal status as patent eligible subject matter under 35 U.S.C. § 101 in *AT&T Corp. v. Excel Communications, Inc.*[27] In that case, the Federal Circuit directly addressed the previous physicality requirement which allowed software to be patent eligible subject matter only when applied to a physical transformation of matter.[28] The Federal Circuit eliminated the physicality requirement for software-related patents.[29] Thus, software's status as patent eligible subject matter, by itself and without a physical transformation of matter, is secure.

Software's new, unequivocal status as patent eligible subject matter means that the courts have done little or nothing to explicate software's novelty, obviousness, enablement, best mode, or written description, the traditional requirements of patent law. Perhaps the only certainties now are that nothing is certain, and that we are beginning a search for rules governing software patents. The Federal Circuit has implicitly

---

[27] AT&T Corp. v. Excel Communications, Inc., 172 F.3d 1352, 1361, 50 U.S.P.Q.2d (BNA) 1447, 1451 (Fed. Cir. 1999).

[28] *Id.* at 1358, 50 U.S.P.Q.2d (BNA) at 1452

[29] *Id.* at 1357, 50 U.S.P.Q.2d at 1450; *see also* John W. Bagby, *Business Method Patent Proliferation: Convergence of Transactional Analytics and Technical Scientifics,* BUS. LAW. 423, 436-39 (2000).

acknowledged this.[30] As Professor Newell said, "fixing the models is an important intellectual task. It will be difficult. The concepts that are being jumbled together—methods, processes, mental steps, abstraction, algorithms, procedures, determinism—ramify throughout the social and economic fabric."[31] Making further progress will require "sustained intellectual labor."[32] Moreover, he made this observation before the onslaught of neural software which creates additional complexity.

For a number of years before *AT&T v. Excel*, members of the public debated whether software should be patent eligible subject matter, and some of this debate was captured in PTO public hearings on software patents in 1994.[33] These hearings reflected no consensus about the desirability of software patents. Subsequently, the PTO acknowledged public debate over the patent eligibility of software and software-implemented business methods by issuing documents demonstrating the rationale for such patents. Later, the PTO's Examination Guidelines for Computer-Related Inventions ("Guidelines") stated that software is section 101 subject matter not only when involved in a physical process, but also in a process that is merely *useful*.[34] The PTO "determined that additional training materials were needed to address how to apply the Guidelines in the areas of business, artificial intelligence and mathematical processing applications. Each of these three areas has shown a high growth rate and increased examining

[30] *AT&T Corp.*, 172 F.3d at 1357, 50 U.S.P.Q.2d at 1450.

[31] Newell, *supra* note 4, at 1035.

[32] *Id.*

[33] The transcript of these hearings is reproduced in GREGORY STOBBS, SOFTWARE PATENTS app. A (2001)[hereinafter 1994 Hearings].

[34] Examination Guidelines for Computer-Related Inventions, 61 Fed. Reg. 7,478, 7,481 (Mar. 29, 1996).

complexity."[35] Following issuance of the Guidelines, the PTO published training materials illustrating how to apply them.[36] The PTO has also issued materials concerning software-implemented business method patents, including the rationale for their status as section 101 subject matter.[37]

Issuance of these apologetics and special examination procedures demonstrates uncertainty about how the software patent regime will be made to work. Given the riddles presented by software patent applications and the rising workload from the accelerating rate of filing, the PTO has done a commendable job in coping. Software patents are being issued in large numbers,[38] however, and many believe that using a "business as usual" approach for software patents will be unsatisfactory.[39] Some argue for an alternate system for protection of software as a whole.[40] Others desire a separate system for protection of software-implemented business methods because, they believe, business is different from more "traditional" areas of

---

35 U.S. PATENT & TRADEMARK OFFICE, EXAMINATION GUIDELINES FOR COMPUTER-RELATED INVENTIONS: TRAINING MATERIALS DIRECTED TO BUSINESS, ARTIFICIAL INTELLIGENCE AND MATHEMATICAL APPLICATIONS, *available at* http://www.uspto.gov/web/offices/pac/compexam/exam comp.pdf (last visited July 2, 2003).

36 *Id.*

37 *See, e.g.,* U.S. PATENT & TRADEMARK OFFICE, WHITE PAPER: AUTOMATED FINANCIAL OR MANAGEMENT DATA PROCESSING METHODS (BUSINESS METHODS): EXECUTIVE SUMMARY, *available at* http://www.uspto.gov/web menu/busmethp/index.html (last visited June 22, 2003).

38 *See* Julie E. Cohen & Mark A. Lemley, *Patent Scope and Innovation in the Software Industry*, 89 CAL. L. REV. 1, 3 (2001).

39 *See* 1994 Hearings, *supra* note 33.

40 *See* Pamela Samuelson, et al., *Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308 (1994).

invention because it is based upon competition and emulation.[41] Less drastic suggestions are for piecemeal refinements of patent rules for software. Some argue for borrowing the doctrine of fair use from copyright law and allowing reverse engineering.[42] Another approach would be to require inventors to disclose more of the software architecture and code to obtain patents, allowing a greater level of scrutiny of the "how and what" of software.[43] Others suggest employing a "level of abstraction" analysis.[44] Some of these approaches include requiring greater judicial emphasis on some particular aspect of Sections 102, 103, and 112 of the Patent Code.[45] Most of these recommendations would require changes in patent prosecution practice.

It is possible that one or more of these approaches might be helpful, but the focus of this article is on less drastic means which could be employed internally by the PTO. Conventional software has been difficult enough to classify, but neural software, based upon mental processes, is more abstract than conventional software, and neural software patents will be more difficult to manage than conventional software patents. Better classification of neural software inventions will improve the quality of prior art searches and examination.

---

41 Leo J. Raskind, *The* State Street Bank *Decision: The Bad Business of Unlimited Patent Protection for Methods of Doing Business*, 10 FORDHAM INTELL. PROP. MEDIA & ENT. L.J. 61, 66 (1999).

42 Maureen A. O'Rourke, *Toward a Doctrine of Fair Use in Patent Law*, 100 COLUM. L. REV. 1177, 1212, 1220-35 (2000); Cohen & Lemley, *supra* note 38, at 3 (reverse engineering).

43 *See* Alan L. Durham, *"Useful Arts" in the Information Age*, 1999 BYU L. REV. 1419, 1528 (1999).

44 Cohen & Lemley, *supra* note 38, at 47-50.

45 35 U.S.C. §§ 101, 102, 112 (2003).

The functions of the CNS provide a powerful model for assessing the novelty and obviousness of neural software patents for a very simple reason: the very intangibility of CNS functions is the stuff of neural software functionality. Such software-implemented methods are the analogs of human cognition, and thus cognitive neuroscience could shed light on questions of patent eligibility. The cognitive neuroscience model would provide a robust means of analysis at almost any level of abstraction because neural computing can be described in terms from the highest to lowest levels of abstraction.

The parallel and distributed functions of the CNS, discussed above, provide a ready set of categories with which to classify "process" software patents, as that term is used in the sense of cognitive neuroscience. These CNS functions, when applied to software, would be a series of CNS analogs describing an aspect of inventions in addition to the current, mostly product-oriented classifications. Such an approach is supported by the Federal Circuit rule that one should "look first to the nature of the problem confronting the inventor . . . If the reference is not within the field of the inventor's endeavor, one looks at whether the field of the reference is reasonably pertinent to the problem the inventor is trying to solve."[46] It is fitting that CNS analogs provide a roadmap to process software examination because software has, for the most part, been patented with highly abstract descriptions. Software patent applicants do not, as a general rule, submit detailed source code. Thus, a problem-solving approach would fit the

[46] Shatterproof Glass Corp. v. Libbey-Owens Ford Co., 758 F.2d 613, 620, 225 U.S.P.Q. (BNA) 634, 638 (Fed. Cir. 1985). See generally DONALD CHISHUM, 2 CHISUM ON PATENTS § 5.03(1) (1994) for presentation of the product-oriented approach and the problem-solving approach. The issue of "dumb art" discussed by Chisum in the referenced section is not the point of the author's recommendation for new patent classifications. Rather, the complexity and intangibility of neural software is what supports a CNS analog, *i.e.*, problem-solving approach.

current practice of filing software patent applications by describing functions of the software without disclosing any of the source code.[47]

A. Problem-Solving Classifications Using CNS Analogs for Computer Vision

An example demonstrates why CNS analogs as classifications could bring clarity. Computer vision, as it now stands, involves extracting and classifying information from images for recognition of faces and signatures, matching fingerprints, inspecting parts on assembly lines, and guiding robots. Image processing and graphics are part of the overall combination which constitutes computer vision.[48] In visual neuroscience, it is known that detection of the edges of an object occurs in Area V1 of the neocortex by means of neurons with a center-surround receptive field. These edge detector neurons are distributed and connected locally to one another in a manner that produces a mapping of the visual field. "Center-surround" means that the center of the cell's receptive field responds to light differently than the outer, surrounding portion of its receptive field.[49] Greatly simplified, a luminance border, or object edge in a natural scene causes center-surrounds (with "On" center and "Off" surround) along a luminance boundary to be relatively inhibited if more of the receptive field is in the less

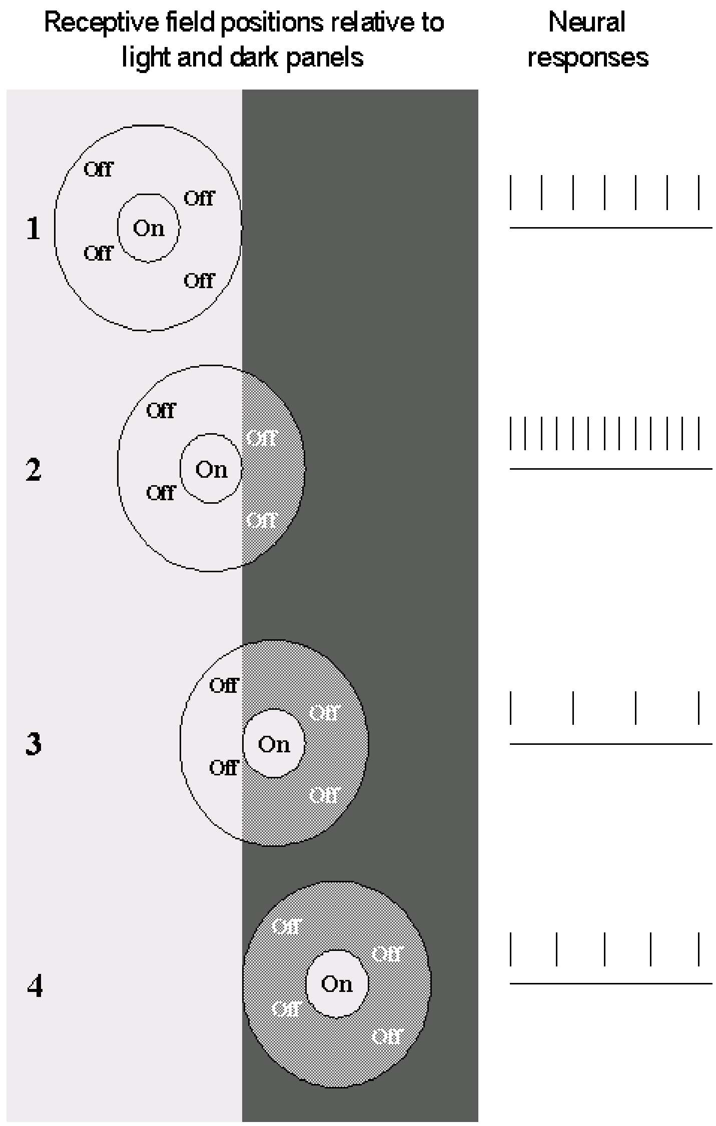
---

[47] See DAVID MARR, VISION: A COMPUTATIONAL INVESTIGATION INTO THE HUMAN REPRESENTATION AND PROCESSING OF VISUAL INFORMATION 24-27 (W.H. Freeman 1982) for a discussion of the levels of analysis of computational issues in computer vision.

[48] JIM R. PARKER, PRACTICAL COMPUTER VISION USING C at xv, xvi, 1 (Wiley 1993).

[49] *See, e.g.,* David Hubel & Torsten Wiesel, *Receptive Fields, Binocular Interaction, and Functional Architecture in the Cat's Visual Cortex*, 160 J. PHYSIOLOGY 106, 115-17 (1962).

luminous area, or relatively excited if less of the receptive field is in the more luminous area, as in the following diagram:[50]

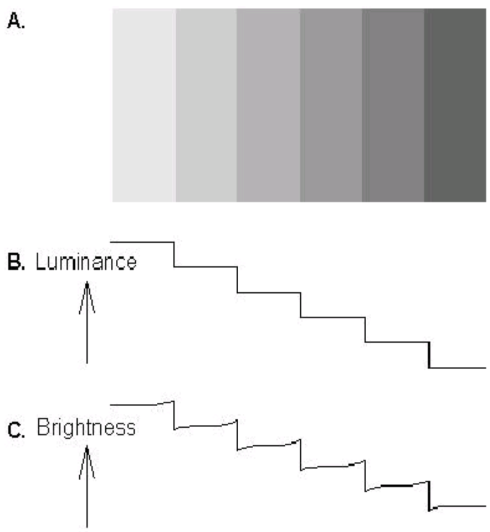


---

[50] THOMAS T. NORTON, ET AL., THE PSYCHOPHYSICAL MEASUREMENT OF VISUAL FUNCTION fig.6-5 (2002).

Neuron 2 is most activated, and neuron 3 is least activated, so the location of the edge is fixed by the relation between the two. A collection of neurons thus relatively inhibited and excited simultaneously produces a pattern of spikes the brain "sees" as an edge.

In a phenomenon called Mach bands, the center-surround organization of the receptive field enables the brain to interpret luminance boundaries as seen in the following three figures:[51]



[51] *Id*. fig.6-4

Each panel appears darker on its right and brighter on its left as a result of accentuation of brightness and darkness at the borders. The different bands of luminance are depicted in figure A, a graph of the absolute luminance level is figure B, and the brightness perceived by the visual cortex is approximated by figure C.

Edge detection, then, is one of the basic CNS visual functions, used by the brain to solve the problem of identification of components of natural scenes such as a face, obstacle, or symbol. Edge detection is also a basic method used by artificial vision systems intended to recognize or manipulate objects using neural computation.[52] There are a number of hypotheses (*i.e.* algorithms) regarding how the brain computes these edges. Correspondingly, there are numerous efforts to design software which can recognize objects or people.[53] Although edge detection algorithms differ, the goal of the inventors is the same: faithful representation of the features of real world objects.

The PTO web database shows, for instance, that edge detection is employed by five patented software inventions in five patent classifications which appear to be unrelated to one another because of the current classification system: bar code scanning, missile telemetry, computer graphics, x-ray equipment, and facial recognition.[54] Each

---

[52] PARKER, *supra* note 48, at 160. A television is not intelligent in the sense of recognizing any object it displays. Object recognition is left to the human viewer.

[53] COMPUTATIONAL MODELS OF VISUAL PROCESSING (Michael S. Landy & J. Anthony Movshon, eds., MIT Press 1991).

[54] U.S. Patent No. 6,328,211 (issued Dec. 11, 2001) (bar code scanner in classification 235); U.S. Patent No. 5,610,598 (issued Mar. 11, 1997) (missile telemetry in classification 340); U.S. Patent No. 5,467,138 (issued Nov. 14, 1995) (computer graphics in classification 348); U.S. Patent No. 5,909,478 (issued June 1, 1999)(x-ray equipment in classification 378); U.S. Patent No. 6,181,806 (issued Jan. 30, 2001)(facial

patent referenced is representative of other patents in its own classification, and the patents cited are not cross-referenced to the other referenced classifications. There are a number of different sub-classifications which contain patents which are not cross-referenced, such as those in 382/118 (facial recognition) which are not cross-classified to 382/119 (signature verification). Some of these inventions might or might not use the same edge detection algorithm, but the actual algorithm is not disclosed by all of these patents. The current application-specific classifications by themselves do not provide a full picture of the relationship among a key feature of these patented inventions. This is an example of how the intangible nature of neural software functionality makes it difficult to classify using PTO application-specific categories.

Identifying the CNS analog to the neural software process in question would allow another method for examining patent applications and classifying issued patents in addition to the product-oriented categories of the current classifications. The current system of classifying neural software inventions by application in this way fails to account for how algorithms are re-used, and fails to disclose the novel aspect of an invention. This pattern is repeated over and over in the current classification system. Each of the CNS analogs for neural computation, if used to classify inventions, could provide a better method for disclosure to the public of the invention's novelty. Likewise, it would provide an additional method for determining obviousness.

For example, suppose a patented invention is classified with the functions of edge detection and image stabilization. Now imagine a new invention which incorporates these two functions plus effective lightness constancy. The presence of the claim of lightness constancy would be a quick indication of novelty. Closer analysis of novelty would be required in any case, of course, but functional classification would provide large clues about novelty in a highly abstract field of invention.

---

recognition in classification 382).

B. Shades of Gray

In any classification system there will be shades of gray which are more difficult to classify. Expert systems, for example, are knowledge-based programs for solving problems within a field, such as medical diagnosis, based upon human input of if-then rules.[55] Another software regime is that of fuzzy systems, which can represent explicit but ambiguous common-sense knowledge and relations.[56] Fuzzy logic is a multi-valued logic with intermediate values to be defined between conventional binary evaluations like zero/one or yes/no. States like "the temperature is high" can be formulated mathematically through the use of sets and processed by fuzzy logic. Although fuzzy logic attempts to apply a more human-like way of thinking in computer programming, it is based upon rules—fuzzy rules.[57]

Software will also contain some elements of machines and some of process. Some elements might be somewhere in between. Each function must be analyzed separately so that its combination in the invention can be understood more clearly.

Other software inventions might appear to reflect neural computation by means of interweaving intricate if-then rules which produce adaptation based only upon pre-set values and thresholds. For example, U.S. Patent No. 5,954,777 claims a system, including software, for determining gear ratio changes in an automatic transmission according to the behavior of the driver, the traffic situation, and the

---

[55] JACK COPELAND, ARTIFICIAL INTELLIGENCE: A PHILOSOPHICAL INTRODUCTION 31 (Blackwell 1993).

[56] NIKOLA K. KASABOV, FOUNDATIONS OF NEURAL NETWORKS, FUZZY SYSTEMS, AND KNOWLEDGE ENGINEERING at xiii (MIT Press 1996).

[57] *Id.* at 167, 192-96.

driving situation to which the vehicle is subjected.[58] According to the patent's "Background of the Invention" and "Summary of the Invention," the invention evaluates whether the driver is inclined toward power-oriented or consumption-optimized driving. A determination can be made whether the vehicle is in city traffic, before or in a curve, on a hill, or in overrun operation. The following variables are supplied to a characteristic field: the detected position of the accelerator pedal, the straight-line vehicle speed, and the transmission rpm or engine rpm. "The essence of the invention is that the determination of the adaptation variable by means of the evaluation mode is dependent upon a comparison of the time-dependent change of the first input variable to at least a threshold value."[59] The outcome of these operations appears to be neuro-mimetic, but needs careful classification in light of CNS functions such as LTP and LTD. Any gray areas in devising new problem-oriented classifications, however, should not delay initiation of the task of developing new classifications.

V. CONCLUSION

Neural software inventions should be examined in light of CNS analogs. Employing cognitive neuroscience as a model for analysis of neurally modeled software, including software-implemented business method patents, would empower the PTO and courts to use novelty, obviousness, and the section 112 requirements to differentiate better among software inventions. As cognitive neuroscience evolves, richer

---

[58] U. S. Patent No. 5,954,777 (issued Sept. 21, 1999). No attempt has been made to review the patent's file wrapper to assess the prosecution history. This patent is accepted at face value for the purpose of illustrating that software-implemented inventions make claims of plasticity in varying ways.

[59] *Id.*

and more meaningful distinctions among neurally-based programs would be possible. The PTO should begin developing a new set of CNS analog classifications for use in conjunction with the current classifications. Once employed, the new classifications would enable greater clarity for examiners, practitioners, and the courts in sorting through novelty, obviousness, best mode, written description, and enablement. The new classifications would be a tool to help render more tangible the difficult task of examination of neural software patent applications.