Santa Clara University School of Law

Santa Clara High Technology Law Journal

Volume 31 | Issue 2 Article 4

January 2015

All That is Solid Melts Into Air: The Subject-Matter Eligibility Inquiry in the Age of Cloud Computing

Scott T. Luan

Recommended Citation

Scott T. Luan, *All That is Solid Melts Into Air: The Subject-Matter Eligibility Inquiry in the Age of Cloud Computing*, 31 Santa Clara High Tech. L.J. 313 (2015).
Available at: http://digitalcommons.law.scu.edu/chtlj/vol31/iss2/4

ALL THAT IS SOLID MELTS INTO AIR: THE SUBJECT-MATTER ELIGIBILITY INQUIRY IN THE AGE OF CLOUD COMPUTING

Scott T. Luan†

This article critiques and refines the subject-matter eligibility inquiry in patent law by examining the process of creativity in the context of software-implemented inventions. As a substantive critique of § 101 jurisprudence, this article, informed by copyright law, proposes a working hypothesis for a general structure of the subject-matter eligibility inquiry in which a critical determination is the appropriate level of abstraction for claim construction. As a discursive critique of the limits and limitations of judicial language, this article argues that courts have incorrectly presumed that contemporary legal thought is equipped, conceptually and linguistically, to understand the full significance of modern technologies and the grammars of their creation. Closely reading the human agency and design choices that inhere in technology, this article seeks to resolve the open questions posed by Mayo Collaborative Servs. v. Prometheus Labs., Inc.*, to harmonize* Diamond v. Diehr *and* Parker v. Flook*, and to refine* Alice Corp. v. CLS Bank Int'l.

† J.D., December 2014, The George Washington University Law School. BS, MS, Stanford University. The author is a patent examiner at the United States Patent & Trademark Office. The opinions expressed herein are those of the author and do not represent the official position, policy, or views of the Patent Office, the United States Department of Commerce, or the United States Government. The author will continue to perform his duties as a patent examiner in accordance with all applicable guidelines, rules, and regulations. I would like to thank Professors Dennis Karjala, Kenneth J. Rodriguez, Gerald J. Mossinghoff, and Ralph Oman for their helpful comments on this article. All errors are mine.

TABLE OF CONTENTS

INTRODUCTION 315
I. OPEN QUESTIONS IN *MAYO COLLABORATIVE SERVS. V. PROMETHEUS LABS., INC.* 322
II. ANALYSIS OF COMPUTER PROGRAMS IN COPYRIGHT LAW 330
A. Abstraction–Filtration in Computer Associates Int'l, Inc. v. Altai, Inc. 331
B. Critique of Altai 334
1. Architecture of Buildings: The Structure of Physical Space 334
2. Architecture of Software: The Structure of Conceptual Space 340
III. SUBJECT-MATTER ELIGIBILITY IN PATENT LAW 353
A. Mayo Revisited 353
B. The General Structure of the Patentable Subject-Matter Eligibility Inquiry 356
IV. CRITIQUE OF *ALICE CORP. V. CLS BANK INTL.* 359
A. Mayo's Framework, Step 1: Level of Abstraction 359
B. Technical Aspect of Mayo's Framework, Step 2: The Search for the Inventive Concept 368
1. Method Claims 368
2. System and Media Claims 371
C. Normative Aspect of Mayo's Framework, Step 2: The Changing Nature of the Machine 373
CONCLUSION 381

TABLE OF FIGURES

Figure 1. Adapter Pattern (Based on GoF, 139) 345
Figure 2. Command Pattern (Based on GoF, 233) 346
Figure 3. Strategy Pattern (Based on GoF, 315) 348

INTRODUCTION

For over two centuries, statutory language describing patentable subject matter has remained largely unchanged. The Patent Act of 1790 provided that "any useful art, manufacture, engine, machine, or device" is eligible subject matter.[1] The Patent Act of 1793, adopting categories that are mostly operative today, provided that "any new and useful art, machine, manufacture or composition of matter" is patentable.[2] More than 150 years later, the Patent Act of 1952 merely substituted the term "process" for "art."[3] The statutory language remains unchanged in the Smith–Leahy Act of 2011.[4]

The extraordinary constancy of the statutory language delimiting patentable subject matter is even more striking in the context of technological developments that span two centuries.[5] The statutory

1. Patent Act of 1790, ch. 7, § 1, 1 Stat. 109, 109–12.

2. Patent Act of 1793, ch. 11, § 1, 1 Stat. 318, 318–23.

3. Patent Act of 1952, ch. 950, 66 Stat. 797 (1952) (codified at 35 U.S.C. § 101) ("Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.").

4. However, the America Invents Act (AIA) provides that "no patent may issue on a claim directed to or encompassing a human organism." *See* Leahy–Smith America Invents Act, Public Law 112-29, § 33(a), 125 Stat. 284 (to be codified in scattered sections of 35 U.S.C.).

5. Karl Marx's prophetic vision of the revolutionary nature of modern technologies is worth quoting at length:

> The bourgeoisie cannot exist without constantly revolutionizing the instruments of production, and thereby the relations of production, and with them the whole relations of society. Conservation of the old modes of production in unaltered form, was, on the contrary, the first condition of existence for all earlier industrial classes. Constant revolutionising of production, uninterrupted disturbance of all social conditions, everlasting uncertainty and agitation distinguish the bourgeois epoch from all earlier ones. All fixed, fast-frozen relations, with their train of ancient and venerable prejudices and opinions, are swept away, all new-formed ones become antiquated before they can ossify. All that is solid melts into air . . .

KARL MARX & FRIEDRICH ENGELS, THE COMMUNIST MANIFESTO 12 (Wildside Press LLC, 2008). Note also that the term "cloud computing" in the title of this article is merely a catchy marketing slogan for a decades-old technology utilizing the server-client networked architecture to provide Internet-based services. "Cloud technology" is catchy precisely because it expresses the widely-felt sense that our experience with computer technology has lost its solidity and concreteness. Articulation of this sense is more difficult than sloganeering. Notwithstanding the title, I will attempt to articulate this sense in my treatment of the developing jurisprudence on subject-matter eligibility. *See*, *e.g.*, Katherine Hayles, *The Condition of Virtuality*, *in* LANGUAGE MACHINES: TECHNOLOGIES OF LITERARY AND CULTURAL PRODUCTION 183, 183–184 (Jeffrey Masten, Peter Stallybrass, & Nancy J. Vickers ed., 1997) ("Virtuality is the condition millions of people now inhabit . . . . Let me offer a strategic definition. *Virtuality is the cultural perception that material objects are interpenetrated by information patterns* . . . . When I say virtuality is a cultural perception, I do not mean it is merely a psychological phenomenon. It is also a mindset that finds instantiation in an array of powerful technologies. The perception facilities the development of the technologies, and the technologies reinforce the perception.").

language has survived the First and Second Industrial Revolutions[6] and the Digital Revolution. As development of technology is heralding a Third Industrial Revolution,[7] the statutory language remains seemingly transcendent.

The apparent stability and clarity of the statutory language is made possible by the evolving jurisprudence on subject-matter eligibility. Courts have attempted to define extra-statutory exclusions to eligible subject matter by introducing ambiguous and equivocal terms.[8] Until the twentieth century, these judicially-created exclusions were commonly articulated as laws, agencies, powers and properties of nature, principles, scientific facts, abstractions, naked ideas, mental processes, intellectual processes or operations, and conceptions of the mind.[9] In a landmark decision that defines the jurisprudence today on eligible subject matter, the Supreme Court in *Diamond v. Chakrabarty* held as unpatentable "laws of nature, physical phenomena, and abstract ideas."[10]

A specter is haunting legal discourse. Scholars have lamented the difficult philosophical and metaphysical nature of the term *abstract* and its questionable provenance in patent law.[11] Courts have also

5. *See* Gottschalk v. Benson, 409 U.S. 63 (1972); Parker v. Flook, 437 U.S. 584 (1978); Diamond v. Diehr, 450 U.S. 175 (1981).

6. *The Third Industrial Revolution*, ECONOMIST (Apr. 21, 2012), http://www.economist.com/node/21553017 ("The *first industrial revolution* began in Britain in the late 18th century, with the mechanisation of the textile industry . . . . The *second industrial revolution* came in the early 20th century, when Henry Ford mastered the moving assembly line and ushered in the age of mass production.") (emphases added).

7. *Id.* ("A number of remarkable technologies are converging: clever software, novel materials, more dexterous robots, new processes (notably three-dimensional printing) and a whole range of web-based services . . . . The factory of the future will focus on mass customization . . . . Now a product can be designed on a computer and 'printed' on a 3D printer. . . . The applications of 3D printing are especially mind-boggling.").

8. *See, e.g.*, Funk Bros. Seed Co. v. Kalo Inoculant Co., 333 U.S. 127, 134–35 (1948) ("For these are vague and malleable terms infected with too much ambiguity and equivocation.").

9. *See* Dana Remus Irwin, *Paradise Lost in the Patent Law? Changing Visions of Technology in the Subject Matter Inquiry*, 60 FLA. L. REV. 775, 788 (2008) (citing case law for each exclusion).

10. Diamond v. Chakrabarty, 447 U.S. 303, 309 (1980).

11. *See, e.g.*, William Michael Schuster, *Predictability and Patentable Processes: The Federal Circuit's In re Bilski Decision and its Effect on the Incentive to Invent*, 11 COLUM. SA. & TECH. L. REV. 1, 4 (2009) ("Specifically, inherently difficult metaphysical questions such as 'What is an abstract idea?' or 'What is the claimed invention?' are not the expertise of judges or patent examiners but rather philosophers."); Donald S. Chisum, *Patenting Intangible Methods: Revisiting Benson (1972) After Bilski (2010)*, 27 SANTA CLARA COMPUTER & HIGH TECH. L. J. 445, 452 (2011) (observing that "most distressingly, the Court's exclusion of 'abstract intellectual concepts' stems from an out-of-context combination of two statements from old cases that did not even involve intangible processes. The first statement—'an idea of itself is not patentable'—is

recognized difficulties associated with the term. For example, Justice Frankfurter characterized terms such as "abstract principles" to be "vague and malleable" and "infected with too much ambiguity and equivocation."[12] Justice Stevens argued that the Supreme Court has never adequately defined the term *abstract idea*.[13] The Federal Circuit, as Judge Rader observed, "will not presume to define 'abstract' beyond the recognition that this disqualifying characteristic should exhibit itself so manifestly."[14] Similarly, District Court Judge Wu for the Ninth Circuit recently suggested that "the two-step test [identifying an abstract idea] may be more like a one-step test evocative of Justice Stewart's most famous phrase ['I know it when I see it . . . .']."[15]

What is at stake cannot be underestimated. In *Alice Corp. v. CLS Bank Int'l*, the problematic term *abstract* figured as a critical notion in the Supreme Court's analysis of the computer-implemented invention at issue.[16] Yet, the term *abstract* remains undefined. According to the Amicus Brief submitted by IEEE-USA, "[t]here are nearly 1 million software-related U.S. patents in force today on which the public relies."[17] However, according to IEEE-USA, there is "widespread misunderstanding among lawyers, judges, and commentators about the

---

from the 1874 *Rubber-Tip Pencil* decision. The second statement—'a principle, in the abstract' is not patentable—is from the 1852 *Le Roy* opinion. The tenor of the two cases is, contrary to the implication of *Benson*, a positive one: that an 'idea' or a 'principle' is patentable when applied to create a novel and useful process or product even though the idea or principle itself is not patentable because it is either well known or too abstract.") (citations omitted).

12. *Funk Bros. Seed Co.*, 333 U.S. at 135.

13. Bilski v. Kappos, 561 U.S. 593, 621 (2010) (Stevens, J., concurring) ("The Court, in sum, never provides a satisfying account of what constitutes an unpatentable abstract idea.").

14. Research Corp. Technologies, Inc. v. Microsoft Corp., 627 F.3d 859, 868 (Fed. Cir. 2010).

15. Eclipse IP LLC v. McKinley Equip. Corp., No. 2:14-cv-00154-GW-AJW, slip op. at 4–5 (C.D. Cal., Sept. 4, 2014) ("Thus, so far, the two-part test for identifying an abstract idea appears to be of limited utility, while comparisons to previously adjudicated patents—or more precisely, to past cases' characterizations of those patent—have done the heavy lifting.") (citation omitted).

16. Alice Corp. Pty. Ltd. v. CLS Bank Int'l, 134 S. Ct. 2347 (2014).

17. Brief of Amicus Curiae IEEE-USA in Support of Neither Party, Alice Corp. Pty. Ltd v. CLS Bank Int'l 2 (Jan. 28, 2014), *available at* http://www.americanbar.org/content/dam/aba/publications/supreme_court_preview/briefs-v3/13-298_np_amcu_ieee.authcheckdam.pdf; *see also Big US Tech Companies Face Major Patent Losses in the Post-Alice World*, *IAM Research Reveals*, INTELL. ASSET MGMT. (Sept. 27, 2014), http://www.iam-magazine.com/Blog/Detail.aspx?g=2028b324-2d4a-4523-9f0d-f0773b8b3fa1 ("[A]ccording to ktMINE 49% of all IBM's US patent holdings could be affected by the *Alice* decision, as could 58% of Google's, 55% of Microsoft's and a whopping 76% of Oracle's . . . it is important to remember that ktMINE's findings are indicative only and that no individual patents were looked at during the research process; while being potentially affected by *Alice* does not automatically imply either invalidity or unenforceability. That said, the figures are stark . . . .").

nature of the innovations produced by the economically dynamic and vibrant software industry."[18] This article attempts to clarify the widespread misunderstanding by examining the nature of software innovations from a technological perspective.

Relevant to the attempt to understand emerging technologies is Gertrude Stein's observation that there is always a lag between our world and our conceptions.[19] For example, consider the battle plan conceived by World War I generals who envisioned "a nineteenth-century war . . . to be fought with twentieth-century weapons."[20] According to one scholar, in her explication of Stein, "we are, each and every one of us, nimble citizens of an always newly technologized, mediated world that hasn't yet entered, much less altered, our categories of thought."[21] For Stein, "every generation composes and explains its life in terms developed by people who did not see what they now see or do what they now do."[22] Stated differently, technological reality defies common sense.[23]

Our metaphors strain to capture the distinctive qualities of computer technology. Computers seem to be of a different order than

18. IEEE-USA Amicus Brief, *supra* note 17, at 3.

19. Similarly, Thorstein Veblen observed that "[a]s always, the language employed [to describe economic reality] and the principles acted on lag behind the facts." THORSTEIN VEBLEN, ABSENTEE OWNERSHIP AND BUSINESS ENTERPRISE IN RECENT TIMES: THE CASE OF AMERICA 5 (Augustus M. Kelley 1964) (1923) ("[Economic] matters are still spoken of in terms handed down from the past, and law and custom still run in terms that are fit to describe a past situation and conform to the logic of a bygone alignment of forces.").

20. Gertrude Stein, *Composition as Explanation*, *in* GERTRUDE STEIN: WRITINGS, 1903 TO 1932, at 520, 529 (Catherine R. Stimpson & Harriet Chessman ed., 1998). Similarly, John Reid, the former British secretary of state for defense, once remarked that "[w]e risk continuing to fight a twenty-first-century conflict with twentieth-century rules."

21. Adalaide Morris, *New Media Poetics: As We May Think/How to Write*, in NEW MEDIA POETICS: CONTEXT, TECHNOTEXTS, AND THEORIES 1, 2 (2006); *See also* ISAIAH BERLIN, KARL MARX: HIS LIFE AND ENVIRONMENT 118 (1963) ("The history of society is the history of the inventive labours that alter man, alter his desires, habits, outlook, relationships both to other men and to physical nature, with which man is in perpetual physical and technological metabolism.").

22. Morris, *supra* note 21, at 1–2 (observing that "[t]he most interesting thinkers about contemporary media are those who, like Stein, insist on a knowledge that exceeds current conceptual categories: the embodied knowledge, in our era, of a world in which children grow up playing with toys that have as much computing power as the giant IBM computers that sold for millions of dollars a generation ago.").

23. Noting the various ways in which technology defies common sense, this article provides support for Professor Feenberg's observation that "[t]hough we may be competent at using many technologies, most of what we think we know about technology in general is false." Andrew Feenberg, *Ten Paradoxes of Technology*, 14 TECHNÉ 3 (2010) (summarizing "many of the conclusions philosophy of technology has reached reflecting on the reality of our technological world. These conclusions appear as paradoxes judged from our everyday perspective. . . . It turns out that most of our common sense ideas about technology are wrong."). *See infra* notes 113, 181, 192, 241, 269, 290, 318.

player pianos.[24] The "basic tool" conception of computer programs seems, upon closer scrutiny, wrong as a factual matter or inconsistent with the principles of patent law.[25] Processors, inherently programmable, seem to elude the bright-line categories of general-purpose or specialized machines.[26] The practice of coding seems

24. *See, e.g.*, Marshall Dann, Commissioner of Patents and Trademarks, Petitioner, v. Thomas R. Johnston, Respondent., 1975 WL 173465 (U.S.), 46–47 ("Petitioner's 'new use' case culminates in a self-serving and deceptive metaphor which compares the stored computer program with recording devices such as the piano roll in the player piano."). *Contra* Robert Gottschalk, Comm'r of Patents v. Benson, 1972 WL 136228 (U.S.), 5 (U.S., 2004) ("The computer does not acquire a new function, in any sense recognizable by the patent law, every time it is programmed to perform a different set of arithmetical calculations, any more than a player piano acquires a new function each time it plays a new song."). The metaphor of the player-piano roll to describe software is deficient because the difference between the player piano and the computer is not a difference of degree but rather a difference in kind. Nobody would deny that computers have transformed the global economy. Nobody would maintain that player pianos have been similarly transformative. Yet, the "piano roll blues" argument remains in currency despite all that we seem to know about computers. But "*[w]hat is it that we know but do not yet know we know?*" Morris, *supra* note 21, at 2–3 (observing a difference between knowledge that is embodied "in our fingertips and attention spans, habits, suspicions, and predilections" and more mediated knowledge that is "book knowledge, cultural doxa, canonical convictions, and common sense.") (emphasis added). A more problematic epistemology, as this article suggests, can perhaps be approached by a critical inquiry into our discourse and language. *See, e.g.*, WENDY HUI KYONG CHUN, PROGRAMMED VISIONS, SOFTWARE ANDMEMORY 4 (2011) ("[S]oftware has become a metaphor for the mind, for culture, for ideology, for biology, and for the economy. . . . Cognitive science, as Paul Edwards has shown, initially comprehended the brain/mind in terms of hardware/software. Molecular biology conceives of DNA as a series of genetic 'programs.' More broadly, culture itself has been posited as 'software,' in opposition to nature, which is 'hardware.' Although technologies, such as clocks and steam engines, have historically been used metaphorically to conceptualize our bodies and culture, *software is unique in its status as metaphor for metaphor itself. As a universal imitator/machine, it encapsulates a logic of general substitutability* . . . .") (emphasis added).

25. Donald S. Chisum, *The Patentability of Algorithms*, 47 U. PITT. L. REV. 959, 983 (1985–1986) (observing that "[p]atents are regularly allowed for processes and structures that have primary and even sole utility in research, including chemical processes, electrical apparatus, and optical instruments. Literally, a microscope is a 'basic tool' for scientific work, but surely no one would assert that a new type of microscope lay beyond the scope of the patent system."). *But see, e.g.*, Gottschalk v. Benson, 409 U.S. 63, 67 (1972) ("Phenomena of nature, though just discovered, mental processes, and abstract intellectual concepts are not patentable, as they are the basic tools of scientific and technological work.").

26. *But see, e.g.*, *In re* Alappat, 33 F.3d 1526, 1545 (Fed. Cir. 1994), *abrogated by In re* Bilski, 545 F.3d 943 (Fed. Cir. 2008) ("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."). The distinction between general and special-purpose machines is a curious compounding of errors. As a technical error, the distinction overlooks technologies (e.g., field-programmable gate arrays) that exemplify the principle of hardware–software equivalence. As an analytical error, the distinction attempts to anchor a seemingly objective criterion on two relative terms—namely, general and special. As a category error, the distinction confuses a thing with the condition for its possibility: A computer can effect its special purpose (i.e., execute its program) only because of its *a priori* condition of being general-purposed (i.e., programmable). *See infra*

categorically different from mathematics and science and code seems much more than merely instructions to be executed on a computer.[27] Yet, we lack the proper language.

This article critiques *Alice* by relying on concepts that have informed computer science since the closing years of the twentieth century. The writings of Christopher Alexander, a noted architect, have profoundly influenced the field of computer science and the creative process of writing software. Alexander sought a new language, a pattern language, to understand the architecture of buildings. The architectural abstractions of software yield to a similar understanding—an understanding that, as this article argues, can be articulated in terms of design choice. Suggesting a new governing metaphor that is analytically useful and descriptively accurate, this article analogizes software abstractions to the architecture of buildings in order to imbue the intangible with a materiality that seems necessary to our contemporary, lagging imaginations.[28]

Part I critiques *Mayo Collaborative Servs. v. Prometheus Labs., Inc.*[29] To the extent that *Alice* relies heavily on the *Mayo* framework, *Alice* suffers from the same problems as *Mayo*. Taking as a point of departure the analytical fiction that an abstract idea should be treated as if it were known in the prior art, this part will examine *Mayo's* problematic harmonization of *Flook* and *Diehr*. *Mayo's* harmonization poses open questions that remain unanswered in *Alice*: Is the eligibility

---

note 306 and accompanying text.

27. *See, e.g.*, HAROLD ABELSON, GERALD JAY SUSSMAN & JULIE SUSSMAN, STRUCTURE AND INTERPRETATION OF COMPUTER PROGRAMS xviii (1996) ("Underlying our approach to this subject is our conviction that 'computer science' is not a science and that its significance has little to do with computers. The computer revolution is a revolution in the way we think and in the way we express what we think. The essence of this change is the emergence of what might best be called *procedural epistemology*—the study of the structure of knowledge from an imperative point of view, as opposed to the more declarative point of view taken by classical mathematical subjects. Mathematics provides a framework for dealing precisely with notions of 'what is.' Computation provides a framework for dealing precisely with notions of 'how to.'"); Peter Kroes & Anthonie Meijers, *Reply to Critics*, 6 TECHNÉ 112 (2002) ("In contrast to science, engineering (technology) is not geared to representing the world, but to changing the world and to making technical artifacts . . . . It seems that traditional epistemology has not taken this basic difference between science and technology seriously; a systematic analysis of engineering knowledge is virtually non-existent (the reason for this may be that often it is tacitly assumed that engineering knowledge is some form of applied scientific knowledge, which leads to the conclusion that the model of knowledge as justified true belief is also applicable to engineering knowledge).").

28. This analogy can be understood as part of a more difficult aspect of the article—that is, a critical inquiry into our language and discourse. As a discursive critique about the limits and limitations of our language, this aspect of the article is necessarily a more difficult project than the substantive critique which proposes to refine *Mayo's* framework.

29. 132 S. Ct. 1289, 1291 (2012).

inquiry, as a relative functional concern, based on predictable guiding principles?[30] How should courts resolve the methodological tension in eligibility inquiry between the "as-a-whole" analysis and analytical dissection?[31] What is the "something more" that is required to transform a claimed invention into eligible subject matter?[32] What is an *abstract idea*?[33]

Part II seeks answers to the questions posed by *Mayo* in copyright law. This part explicates and critiques *Computer Associates Int'l, Inc. v. Altai, Inc.*, the leading case on analyzing computer programs in copyright law.[34] Notably, this part does not attempt to clarify the relative roles of patent and copyright protection of computer programs.[35] Rather, this part argues for a more modest proposition—that the jurisprudence on the copyrightability of software can inform the jurisprudence on patent eligibility of software-implemented inventions.[36] Specifically, this part establishes that *Altai* articulates two

---

30. It is the burden of this article to show that understanding creativity as design choice (that *inter alia* encodes tradeoffs between benefits and costs) yields predictable guiding principles. *See infra* notes 236–251 and accompanying text.

31. This article argues that constitutive design choices can be fruitfully analyzed, at the appropriate level of abstraction, in terms of their degree of technical necessity in light of the whole. *See infra* notes 186–201 and accompanying text.

32. This article argues that the well-accepted principle that the whole must exceed the sum of the parts, as articulated by *Mayo*, calls not for arithmetic but for explication. The excess cannot exist apart from the inquiry. *See infra* notes 125–130 and accompanying text.

33. To be sure, some abstract ideas are undeniably fundamental building blocks of human ingenuity. This article argues that less fundamental abstract ideas at lower levels of abstraction (that may also be building blocks) can be fruitfully characterized as *abstractions*. Explicating the software architectures at issue in *Altai* and *Alice*, this article argues that, in the specific context of the eligibility inquiry, *abstraction* is semantically manifold—*abstraction* is at once a noun and a verb. In other words, abstractions are the products of the creative act of abstraction. *See infra* notes 148–156 and accompanying text.

34. 982 F.2d 693 (2d Cir. 1992).

35. The problem of coordination between copyright and patent law to find the optimal social policy balance of legal protection for computer software has already been examined in legal scholarship. For example, Professor Karlaja compelling argues that copyright law is not well-suited to protect the functional and structural aspects of computer programs. *See, e.g.*, Dennis S. Karjala, *The Relative Roles of Patent and Copyright in the Protection of Computer Programs*, 17 J. MARSHALL J. COMPUTER & INFO. L. 41, 73 (1998) (arguing that "[c]opyright protection for functional but nonliteral program elements, such as SSO [structure, sequence, and organization] and functional aspects of interfaces would represent an unnecessary and indeed unfortunate intrusion into the subject matter of patent law."); Dennis S. Karjala, *Distinguishing Patent and Copyright Subject Matter*, 35 CONN. L. REV. 439 (2002–2003); Dennis S. Karjala, *A Coherent Theory For The Copyright Protection Of Computer Software And Recent Judicial Interpretations*, 66 U. CIN. L. REV. 53 (1998).

36. Readers are invited to draw their own conclusions about the relative roles of the two bodies of law based on the truth of this proposition. Admittedly, this article suggests a degree of (potential) analytical redundancy or overlap between copyright and patent law, which may provide further support for Professor Karjala's thesis that copyright protection of computer

critical insights about the nature of software—namely, that software can embody multiple ideas at different levels of abstraction and that software is the product of design choices. Relying on the pattern language of software design and Alexander's architectural patterns, this part also critiques and refines *Altai* by arguing that *Altai* overlooks the hidden interior of software architecture, an architecture that is structured by design choices. This part serves as the basis for a critique of § 101 jurisprudence in the subsequent parts.

Part III revisits the questions posed by *Mayo*. This part argues that the harmonization of *Flook* and *Diehr* represents a special case of a more generalized structure of the eligibility inquiry that unfolds along three lines of analyses. Part IV argues that *Alice* should have been decided along one line of analysis rather than unfolding incorrectly along another line. This last part critiques *Alice* by relying (and elaborating) on the concepts developed in the article.

I. OPEN QUESTIONS IN *MAYO COLLABORATIVE SERVS. V. PROMETHEUS LABS., INC.*

Decided at the dawn of the Information Age, the *Benson*, *Flook*, and *Diehr* trilogy remains foundational in the developing jurisprudence on subject-matter eligibility, for better or for worse.[37] In the trilogy, the Supreme Court struggled with a question that is still largely unresolved today: how and when does a discovery of an abstract idea, scientific truth, or mathematical principle become a patentable invention?

In its attempt to harmonize *Flook* and *Diehr*, *Mayo* advanced § 101 jurisprudence. Unlike *Bilski*, which relied too heavily on the undefined term "abstract,"[38] *Mayo* sought to clarify analytical

---

programs should be reduced. Note also that the truth of the proposition does not mean that copyright jurisprudence should be taken at face value. This article takes *Altai* only as a starting point because *Altai* allows for a definition of *abstraction* that is analytically useful. Critiquing *Altai*, this article will argue that *abstraction* is really about (design) choice which, in turn, is fundamental to creativity. Thus, this article can be read as an argument *pace* Karjala that an inquiry into creativity should enter directly into the eligibility analysis because creativity itself clarifies (rather than muddles) the analysis. *But see* Dennis S. Karjala, *Copyright and Creativity*, 15 UCLA ENT. L. REV. 169, 173 (2008) ("[C]reativity itself is not analytically useful in making the most basic determination of whether the creativity under consideration falls within copyright subject matter, patent subject matter, or neither."). *See infra* notes 170–182 and accompanying text.

37. *See* Gottschalk v. Benson, 409 U.S. 63 (1972); Parker v. Flook, 437 U.S. 584 (1978); Diamond v. Diehr, 450 U.S. 175 (1981).

38. "[The claims] are attempts to patent *abstract* ideas. Indeed, all members of the Court agree that the patent application at issue here falls outside of § 101 because it claims an *abstract* idea. . . . The concept of hedging, described in claim 1 and reduced to a mathematical formula in claim 4, is an unpatentable *abstract* idea, just like the algorithms at issue in *Benson* and *Flook*.

difficulties in the trilogy.[39] However, *Mayo's* harmonization poses open questions. This part examines these questions which, as the following parts show, suggest a refinement of the eligibility inquiry. This part takes as a point of departure the analytical fiction that an abstract idea should be treated as if it were known in the prior art. The analytical fiction serves to reveal the conceptual difficulties in drawing terminological and methodological distinctions between the eligibility and patentability inquiries.

Analytical fictions obscure conceptual difficulties. *Mayo* questioned the awkward and well-established analytical fiction that an abstract idea should be treated as if it were known in the prior art.[40] The analytical fiction was first articulated in an English case and was subsequently adopted by American courts.[41] By subscribing to the analytical fiction, courts have introduced an additional and artificial requirement for doctrinal consistency.[42] Specifically, the analytical fiction has complicated courts' attempts to draw terminological and methodological distinctions between the eligibility and patentability inquires. In regards to the terminological question, by subscribing to the analytical fiction that abstract ideas always lack novelty, courts are

---

Allowing petitioners to patent risk hedging would pre-empt use of this approach in all fields, and would effectively grant a monopoly over an *abstract* idea. . . . Petitioners' remaining claims are broad examples of how hedging can be used in commodities and energy markets. *Flook* established that limiting an *abstract* idea to one field of use or adding token postsolution components did not make the concept patentable. That is exactly what the remaining claims in petitioners' application do. These claims attempt to patent the use of the *abstract* idea of hedging risk in the energy market and then instruct the use of well-known random analysis techniques to help establish some of the inputs into the equation. Indeed, these claims add even less to the underlying *abstract* principle than the invention in *Flook* did, for the *Flook* invention was at least directed to the narrower domain of signaling dangers in operating a catalytic converter. Today, the Court once again declines to impose limitations on the Patent Act that are inconsistent with the Act's text. The patent application here can be rejected under our precedents on the unpatentability of *abstract* ideas." Bilski v. Kappos, 561 U.S. 593, 609–12 (2010) (emphases added).

39. Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1298–1300 (2012) (characterizing *Flook* and *Diehr* as "two cases in which the Court reached opposite conclusions about the patent eligibility of processes that embodied the equivalent of natural laws.").

40. *Id.* at 1304 ("Intuitively, one would suppose that a newly discovered law of nature is novel. The Government, however, suggests in effect that the novelty of a component law of nature may be disregarded when evaluating the novelty of the whole. See Brief for United States as Amicus Curiae 27. But §§ 102 and 103 say nothing about treating laws of nature as if they were part of the prior art when applying those sections.").

41. *See, e.g.*, Neilson v. Harford, Web. Pat. Cases 295, 371 (1844); O'Reilly v. Morse, 56 U.S. 62, 115 (1853); *Flook*, 437 U.S. at 592.

42. Professor Chisum characterized this analytical fiction as "aberrational." *See* Donald S. Chisum, *The Patentability of Algorithms*, 47 U. PITT. L. REV. 959, 994–95 (1985–86) ("That aberration was so basically antithetical to patent law principles that it would have to be purged.").

tacitly advancing the position that there is some analytical commingling of eligibility and patentability. In regards to the methodological question, by treating as prior art the portion of the claim that recites an abstract idea, courts are implicitly advocating for analytical dissection. However, such tacit and implicit positions are far from settled in the developing jurisprudence on subject-matter eligibility.

The jurisprudence on subject-matter eligibility attempted to draw terminological distinctions between the eligibility and patentability inquires. In *Flook*, the Court first articulated the § 101 requirement of "new and useful" as an "inventive concept in its application."[43] The "inventive concept" formulation has gained wide currency.[44] However, Judge Rich argued that "confusion creeps in through such phrases."[45] In his critique of *Flook*, Judge Rich argued in *In re Bergy* that the Court had erred in "commingling of distinct statutory provisions which are conceptually unrelated."[46] Rich maintained that "[n]otwithstanding the words 'new and useful' in § 101, the invention is not examined under that statute for novelty because that is not the statutory scheme of things or the long-established administrative practice."[47] Judge Rich, relying on legislative history, argued that "in 1952 Congress voiced its intent to consider the novelty of an invention under § 102 where it is first made clear what the statute means by 'new', notwithstanding the fact that this requirement is first named in § 101."[48] In other words, "[o]f the three requirements stated in § 101, only two, utility and statutory subject matter, are applied under § 101."[49]

Diehr echoed Judge Rich's critique of *Flook*. *Diehr* cited *In re Bergy* to stand for the proposition that the question of "whether a particular invention is novel is wholly apart from whether the invention falls into a category of statutory subject matter."[50] Relying on the same legislative history as cited by Judge Rich, *Diehr* argued that the question of novelty is properly characterized as a § 102 analysis even

43. *See Flook*, 437 U.S. at 594.

44. *See, e.g.*, *Mayo*, 132 S. Ct. at 1294.

45. Application of Bergy, 596 F.2d 952, 959 (C.C.P.A. 1979), *vacated in part sub nom.*, *See also* Diamond v. Chakrabarty, 444 U.S. 1028 (1980), *aff'd sub nom.* Diamond v. Chakrabarty, 447 U.S. 303 (1980).

46. *Application of Bergy*, 596 F.2d at 959.

47. *Id.* at 960.

48. *Id.* at 961.

49. *Id.*

50. Diamond v. Diehr, 450 U.S. 175, 190 (1981) (citing *In re Bergy*, 596 F.2d at 961).

though the statutory language of § 101 includes the term "new."[51] *Diehr* reasoned that the language of § 101 is merely "a general statement of the type of subject matter that is eligible for patent protection 'subject to the conditions and requirements of this title [such as novelty].'"[52] The legislative history reveals that the language of § 102 is, in effect, an "amplification" and "definition" of the term "new" in § 101.[53] In short, consistent with Judge Rich's analysis of the legislative history, *Diehr* concluded that "[a] rejection on either of these [anticipation and obviousness] grounds does not affect the determination that respondents' claims recited subject matter which was eligible for patent protection under § 101."[54]

However, *Mayo* allowed for the possibility that the relationship between the eligibility and patentability inquiries cannot be clarified by a simple, terminological, bright-line rule. Contrary to Judge Rich's absolutism that "[t]o provide the option of making such a [anticipation] rejection under either § 101 or § 102 is confusing and therefore bad law,"[55] *Mayo* adopted a more nuanced approach. *Mayo* observed that "in evaluating the significance of additional steps, the § 101 patent-eligibility inquiry and, say, the § 102 novelty inquiry might sometimes overlap."[56] *Mayo* reasoned that allocating the determination of novelty *entirely* to § 102 "risks creating significantly greater legal uncertainty, while assuming that those sections can do work that they are not equipped to do."[57] Thus, *Mayo* stands for the proposition that the determination of novelty serves different purposes in the eligibility and patentability inquiries. According to *Mayo*, the underlying purpose of the eligibility inquiry is to determine "how much future innovation is foreclosed relative to the contribution of the inventor."[58] Accordingly, *Mayo* characterized the eligibility inquiry as a relative functional concern.[59] An open question remains whether the relative functional

---

51. "The corresponding section of [the] existing statute is split into two sections, section 101 relating to the subject matter for which patents may be obtained, and section 102 defining statutory novelty and stating other conditions for patentability." *Diehr*, 450 U.S. at 191 (citations omitted).

52. *Id.* at 189.

53. *Id.* at 190–91 (citations omitted).

54. *Id.* at 191.

55. Application of Bergy, 596 F.2d 952, 961 (C.C.P.A. 1979), *vacated in part sub nom.*, Diamond v. Chakrabarty, 444 U.S. 1028 (1980), *aff'd sub nom.*, Diamond v. Chakrabarty, 447 U.S. 303 (1980).

56. Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1304 (2012).

57. *Id.*

58. *Id.* at 1303.

59. *Id.* ("[T]he underlying functional concern here is a relative one . . . .").

concern is based on predictable guiding principles or merely a license for a free-ranging inquiry.

The methodological distinction between the eligibility and patentability inquiries has also been a source of conceptual difficulty. For example, *Flook* opined that the eligibility inquiry should consider the claim as a whole.[60] However, in practice, *Flook* analytically dissected the claim at issue by considering whether each of the recited steps was well known.[61] By contrast, *Diehr* cautioned that "[i]t is inappropriate to dissect the claims into old and new elements … new combination of steps in a process may be patentable even though all the constituents of the combination were well known . . . ."[62] *Mayo* acknowledged the methodological tension between "as-a-whole" analysis and analytical dissection by submitting that the eligibility inquiry is as much about the claim as a whole as about the claim's constituent parts. Relying on the well-established formulation that the whole must exceed the sum of the parts,[63] *Mayo* reasoned that the claim at issue is ineligible subject matter because the "[claimed] steps, when viewed as a whole, add nothing *significant* beyond the sum of their parts taken separately."[64] However, open questions remain about how the "sum [of the parts]" and "[the claim as a] whole" are to be determined. The mereology of claimed inventions remains undefined.

Mayo poses the question of how a critical difference (*i.e.,* between the "sum" and the "whole") should be conceived. *Mayo* suggests that the critical difference can be characterized as an "integration."[65] But this does not get us very far. If "integration" is a measure of (necessary)

60. Parker v. Flook, 437 U.S. 584, 594 (1978) ("Our approach to respondent's application is, however, not at all inconsistent with the view that a patent claim must be considered as a whole. Respondent's process is unpatentable under § 101, not because it contains a mathematical algorithm as one component, but because once that algorithm is assumed to be within the prior art, the application, considered as a whole, contains no patentable invention.").

61. *Id.* ("The chemical processes involved in catalytic conversion of hydrocarbons are well known, as are the practice of monitoring the chemical process variables, the use of alarm limits to trigger alarms, the notion that alarm limit values must be recomputed and readjusted, and the use of computers for 'automatic monitoring-alarming.'").

62. Diamond v. Diehr, 450 U.S. 175, 188 (1981).

63. *See, e.g.*, Great Atl. & Pac. Tea Co. v. Supermarket Equip. Corp., 340 U.S. 147, 152 (1950) ("The conjunction or concert of known elements must contribute something; only when the whole in some way exceeds the sum of its parts is the accumulation of old devices patentable.").

64. Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1298 (2012) (emphasis added).

65. *Id.* ("The Court pointed out that the basic mathematical equation, like a law of nature, was not patentable. But it found the overall process patent eligible because of *the way the additional steps of the process integrated the equation into the process as a whole*.") (emphasis added).

involvement, then the post-solution steps in both *Flook* and *Diehr* "integrate" their respective formulas into a process as a whole because the steps are dictated by the terms of the formulas, as discussed below. If "integration" is a measure of synergistic or emergent advantage, then the post-solution steps in both *Flook* and *Diehr* "integrate" their respective formulas into a process as a whole because the steps enable the formulas to acquire a new potency in the world (to control the processes of rubber-curing and catalytic conversion). (Or, does *Mayo* suggest an "integration" that is a different calculus of sums?) Therefore, *Mayo* does not adequately resolve the methodological tension between the "as-a-whole" analysis and analytical dissection.

This article attempts to answer the questions posed by *Mayo*. This article argues that the eligibility inquiry, as a relative functional concern, is not a free-ranging inquiry but rather one that is (or can be) guided by principles that reflect an understanding of the creative process. This article also argues that the relationship between the whole and the parts of a claim can only be properly appreciated at the appropriate level of abstraction. These answers are found in copyright law, the topic of the following part. As a segue to the following part, the discussion will now focus on why *Mayo's* harmonization of *Flook* and *Diehr* is problematic in a way that is corrigible by insights from copyright law.

Mayo's harmonization of *Flook* and *Diehr* is problematic. For example, *Mayo's* articulation of the requisite qualities for patent eligibility remains vague and tautological. *Mayo* never explicitly clarified what is required to "transform[] the [claimed] process into an inventive application of the formula."[66] *Mayo* simply concluded that "[t]hese other steps [in *Diehr*] *apparently* added to the formula *something* that in terms of patent law's objectives had significance."[67] However, this significant, inventive "something" is far from apparent because *Mayo's* reasoning is tautological:[68] A patent-eligible application of a formula that is inventive and significant has "something" that is necessary for patent eligibility. In other words, an

66. *Id.* at 1299.

67. *Id.* (emphases added).

68. The tautological reasoning in *Mayo* recalls the circular reasoning in *Benson*. *See* Donald S. Chisum, *Patenting Intangible Methods: Revisiting* Benson *(1972) After* Bilski *(2010)*, 27 SANTA CLARA COMPUTER & HIGH TECH. L.J. 445, 454 (2011) ("In the 'nutshell,' the Court's reasoning is circular. The applicants Benson and Tabbot asked for a claim to the algorithm itself as a new and useful process that did not reflect any natural law or scientific truth. The Court responded to this question, in effect saying: the claim to an algorithm cannot be allowed because if it is, the claim will be to the algorithm [which is ineligible subject matter].").

application of a formula is eligible subject matter because it has the requisite qualities for patent eligibility.

Further, *Mayo's* characterization of the claimed invention in *Flook* could be applied with equal validity to that in *Diehr*.[69] *Mayo* correctly observed that the post-solution steps in *Flook* (e.g., monitoring variables, recomputing and readjusting alarm limit value) were well-known, conventional, and obvious.[70] In fact, the post-solution steps in *Flook* are *dictated* by the terms of the equation.[71] The equation at issue in *Flook* is known in the signal-processing art as an "exponential smoothing" filter that, over time, assigns exponentially decreasing weights to past observations of a given signal.[72] Signal-processing filters, by definition, monitor input variable(s) and update output(s). Similarly, in *Diehr*, the post-solution steps are also *dictated* by the terms of the equation at issue.[73] The steps include "installing rubber [characterized by activation energy constant, *C*] in a press, closing the mold [characterized by a geometric constant, *x*], constantly determining the temperature of the mold [temperature variable, *Z*], constantly recalculating the appropriate cure time [cure time variable,

---

69. In his dissent in *Diehr*, Justice Stevens arrived at a similar conclusion. Stevens argued that "[t]here is no suggestion that there is anything novel in the instrumentation of the mold, in actuating a timer when the press is closed, or in automatically opening the press when the computed time expires. Nor does the application suggest that Diehr and Lutton have discovered anything about the temperatures in the mold or the amount of curing time that will produce the best cure. What they claim to have discovered, in essence, is a method of updating the original estimated curing time by repetitively recalculating that time pursuant to a well-known mathematical formula in response to variations in temperature within the mold. Their method of updating the curing time calculation is strikingly reminiscent of the method of updating alarm limits that Dale Flook sought to patent." Diamond v. Diehr, 450 U.S. 175, 208–09 (1981).

70. Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1299 (2012).

71. "Claim 1 of the patent describes the method as follows: 1. A method for updating the value of at least one alarm limit on at least one process variable involved in a process comprising the catalytic chemical conversion of hydrocarbons wherein said alarm limit has a current value of $Bo + K$ wherein *Bo* is the current alarm base and *K* is a predetermined alarm offset which comprises: (1) Determining the present value of said process variable, said present value being defined as *PVL*; (2) Determining a new alarm base *B1*, using the following equation: $B1 = Bo\ (1.0 - F) + PVL\ (F)$ "where *F* is a predetermined number greater than zero and less than 1.0; (3) Determining an updated alarm limit which is defined as $B1 + K$; and thereafter (4) Adjusting said alarm limit to said updated alarm limit value." Parker v. Flook, 437 U.S. 584, 596–97 (1978).

72. The formula is characterized as exponential because the terms can be expanded (i.e., for a range of time) into a geometric progression which is a discrete version of an exponential function.

73. *Diehr*, 450 U.S. at 181. Note that the claim at issue in *Diehr* does not specify the frequency of the calculations. While *Alice* did not engage in "mental steps" analysis, post-*Alice* opinions invalidating patent claims based on "mental steps" analysis seem to overlook the fact that the calculations based on the Arrhenius equation can easily be performed by humans. *See, e.g.*, Planet Bingo, LLC v. VKGS LLC, No. 2013-1663 (Fed. Cir. Aug. 26, 2014).

v] through the use of the formula and a digital computer, and automatically opening the press at the proper time [as dictated by the equation ln $v = CZ + x$]."[74] Therefore, *Mayo* incorrectly held that, in *Diehr*, "it [is] nowhere suggested that all these [post-solution] steps, or at least the combination of those steps, were in context obvious, already in use, or purely conventional."[75]

Its problems notwithstanding, *Mayo's* harmonization of *Flook* and *Diehr* is corrigible. According to *Mayo*, the claimed invention at issue in *Flook*, unlike that in *Diehr*, required something extra-algorithmic—that is, additional steps that are not dictated by the terms of the algorithm—in order to be eligible subject matter. *Mayo* suggests that the necessary extra-algorithmic step(s) may "explain how the variables [e.g., weighing factor, *F*] used in the formula were selected" or describe "the means of setting off an alarm."[76] In contrast, the patent-eligible claim at issue in *Diehr* does not need to recite *directly comparable* extra-algorithmic step(s) that explain how the variables (e.g., geometric constant, *x*) used in the formula were selected or the means of opening the press. That *Mayo* required something extra-algorithmic in *Flook* but not in *Diehr* suggests that there is a critical difference between the two formulas. *Mayo* suggests that the difference lies in the level of generality or abstractness.[77]

As a description of an underlying principle, the term "abstractness" is related to but distinct from "narrowness." While both concepts relate to the applicability of a given principle, the concept of "abstractness" can be understood *formally* while "narrowness" can be understood *substantively*. A principle is abstract because its generalized *formulation* allows for wide applicability independent of particular contexts.[78] For example, unlike the formula at issue in *Diehr*, the "smoothing" formula in *Flook* defines relationships among more abstract variables that are less directly connected to physical phenomena, processes, or quantities. For instance, the process variable (*PVL*) can be any variable within any process. In *Diehr*, the formula at

---

74. *Diehr*, 450 U.S. at 187.

75. *Mayo*, 132 S. Ct. at 1299.

76. *Id.*

77. Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1300 (2012) ("Other cases offer further support for the view that simply appending conventional steps, specified at a high level of generality, to laws of nature, natural phenomena, and abstract ideas cannot make those laws, phenomena, and ideas patentable.").

78. *See, e.g.*, Tun-Jen Chiang, *The Levels of Abstraction Problem in Patent Law*, 105 NW. U. L. REV. 1097, 1127 (2011) (observing that courts often confuse vagueness with abstraction and submitting that "an idea is abstract if it lacks specific context").

issue describes functional relationships among specific variables in a rubber-curing process.[79] For example, the temperature variable (*Z*) can only represent the temperature of the mold. Accordingly, the "smoothing" formula in *Flook* can be characterized as more abstract than the equation in *Diehr*.

By contrast, a formula is narrow because it applies, in *substance*, only to limited phenomenon. *Mayo* correctly characterized the principle at issue as narrower compared to Einstein's law of relativity.[80] The principle at issue in *Mayo* narrowly describes drug-induced physiological phenomenon (i.e., the level of 6–thioguanine in a subject).[81] Einstein's law of relativity is less narrow because it describes a wider range of phenomenon. Einstein's general theory of relativity applies to any physical phenomenon in which effects of gravitation are significant. However, narrower principles or formulas may nevertheless be more abstract. For example, Einstein's general theory of relativity is arguably less abstract than the narrower "smoothing" formula in *Flook* because the terms in Einstein's theory correspond to physical quantities (e.g., speed of light) and its solutions correspond to actual physical phenomenon (e.g., curvature of space-time).

Mayo observed that "our cases have not distinguished among different laws of nature according to whether or not the principles they embody are sufficiently narrow. And this is understandable. Courts and judges are not institutionally well suited to making the kinds of judgments needed to distinguish among different laws of nature."[82] By contrast, as discussed in the following part, copyright law routinely distinguishes ideas in terms of abstractness. Informed by copyright law, Parts III and IV will examine how the notion of abstractness (i.e., level of abstraction) can harmonize *Flook* and *Diehr*, serve as the basis for a critique of *Alice*, and refine the subject-matter eligibility inquiry in patent law.

II. ANALYSIS OF COMPUTER PROGRAMS IN COPYRIGHT LAW

Part II.A establishes that *Altai* articulates two critical insights about the nature of software. First, *Altai* observes that software can

79. The formula at issue in *Diehr* is a special empirical case of a more generalized Arrhenius equation. The Arrhenius equation, describing the temperature dependence of reaction rates, can be applied to other thermal processes.

80. *Mayo*, 132 S. Ct. at 1303.

81. *Id.* at 1295.

82. *Id.* at 1303.

embody multiple ideas at different levels of abstraction. Second, *Altai* observes that software is the product of design choices. *Altai's* observations about the nature of software will serve, in Part IV, as the basis for the critique of *Alice*.[83] Using the pattern language of software design, Part II.B critiques and refines *Altai* by arguing that *Altai* overlooks the hidden interior of software architecture that is structured by design choices. The critique serves the larger thesis that the eligibility inquiry in patent law (and *Alice* in particular) should focus on what has been obscured in *Altai*.

A. Abstraction–Filtration in Computer Associates Int'l, Inc. v. Altai, Inc.

In the seminal case *Nichols v. Universal Pictures Corp.*, Judge Learned Hand articulated the now-famous "abstraction test" regarding the scope of protectable subject matter in copyright law:

> Upon any work, and especially upon a play, a great number of *patterns of increasing generality* will fit equally well, as more and more of the incident is left out. The last may perhaps be no more than the most general statement of what the play is about, and at times might consist only of its title; but there is a point in this *series of abstractions* where they are no longer protected, since otherwise the playwright could prevent the use of his "ideas," to which, apart from their expression, his property is never extended.[84]

At the dawn of the Internet, *Computer Associates Int'l, Inc. v. Altai, Inc.* adapted Judge Hand's "abstraction test" to computer software.[85] *Altai* articulated a three-part abstraction-filtration-comparison test, the first part of which was derived from Hand's "abstraction test." The abstraction step "resembles reverse engineering on a theoretical plane."[86] Stated differently, the abstraction step mirrors the actual process of software creation, involving an analysis that dissects and isolates each level of abstraction within a program's structure. Worth reproducing in full is the court's description:

> As an anatomical guide to this procedure, the following description is helpful: At the lowest level of abstraction, a computer program

83. *See supra* notes 35–36 and accompanying text.

84. Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930) (emphases added).

85. Computer Associates Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 706 (2d Cir. 1992). *Altai* also relied on Professor Nimmer's proposed test for substantial similarity of non-literal elements in a computer program. *See* David Nimmer et al., *A Structured Approach to Analyzing the Substantial Similarity of Computer Software in Copyright Infringement Cases*, 20 ARIZ. ST. L.J. 625 (1988).

86. *Altai*, 982 F.2d at 707.

> may be thought of in its entirety as a set of individual instructions organized into a hierarchy of modules. At a higher level of abstraction, the instructions in the lowest-level modules may be replaced conceptually by the functions of those modules. At progressively higher levels of abstraction, the functions of higher-level modules conceptually replace the implementations of those modules in terms of lower-level modules and instructions, until finally, one is left with nothing but the ultimate function of the program. . . . A program has structure at every level of abstraction at which it is viewed. At low levels of abstraction, a program's structure may be quite complex; at the highest level it is trivial.[87]

Thus, "to retrace and map each of the designer's steps—in the opposite order in which they were taken during the program's creation" reveals that software creation is a process of abstraction.[88]

Relying on Hand's "abstraction test," *Altai* derived the abstraction step in its three-part test as a response to and a critique of *Whelan Associates, Inc. v. Jaslow Dental Lab., Inc*.[89] *Whelan* first introduced the idea/expression distinction to the software context. Legal scholarship has, with good reasons, overwhelmingly focused on the deficiencies of *Whelan*. However, a critical insight was obscured in the chorus of criticisms. *Whelan* observed that "the coding process is a comparatively small part of programming. By far the larger portion of the expense and difficulty in creating computer programs is attributable to the development of the structure and logic of the program … rather than to the coding."[90] *Whelan* stands for the premise that analyses of computer programs must be based on an accurate understanding of the creative process of writing software, a premise that has underwritten, implicitly or explicitly, the jurisprudence on the copyrightability of software.

Nevertheless, commentators have criticized *Whelan* for oversimplifying the complex nature of computer software. *Whelan* held that software embodies a single purpose or idea.[91] Therefore, according

---

87. *Id*. at 707 (citation omitted).

88. *Id*. at 706–07 ("[T]he theoretic framework for analyzing substantial similarity expounded by Learned Hand in the *Nichols* case is helpful in the present context. In *Nichols*, we enunciated what has now become known as the 'abstractions' test for separating idea from expression . . . . While the abstractions test was originally applied in relation to literary works such as novels and plays, it is adaptable to computer programs. In contrast to the *Whelan* approach, the abstractions test 'implicitly recognizes that any given work may consist of a mixture of numerous ideas and expressions.'") (citation omitted).

89. 797 F.2d 1222 (3d Cir. 1986).

90. *Id.* at 1231.

91. *Id.* at 1236.

to *Whelan*, the particular means chosen for achieving the desired purpose—the structure, sequence and organization of the computer program—is protectable expression.[92] Legal scholars have argued that *Whelan* overprotects computer software, renders ineffective "cleanroom" software development techniques, overburdens new entrants, and encourages inefficient divergence in the software technology.[93]

Joining in the chorus of criticisms, *Altai* maintained that "*Whelan's* general formulation that a program's overall purpose equates with the program's [single] idea is descriptively inadequate."[94] *Altai* advanced the notion that ideas and expressions can be found at different levels of abstraction within the structure of a computer program. "In contrast to the *Whelan* approach," *Altai* reasoned, "the abstractions test implicitly recognizes that any given work may consist of a mixture of numerous ideas and expressions."[95] *Altai's* conception of software as embodying multiple ideas at different levels of abstraction is directly relevant to the software at issue in *Alice*, as Part IV argues.[96]

The second, filtration step of the *Altai* test focuses on design choices in the creative process of writing software. *Altai* observed that "*Whelan's* approach to separating idea from expression in computer programs relies too heavily on metaphysical distinctions and does not place enough emphasis on practical considerations . . . . [A] satisfactory answer to this problem cannot be reached by resorting, a priori, to philosophical first principals [sic]."[97] The second step of the *Altai* test separates or filters protectable expression from non-protectable material by examining software components at each level of abstraction to determine the extent of a programmer's freedom of design choice. Software components dictated by considerations of efficiency or required by external factors are nonprotectable expressions.[98]

---

92. *Id.*

93. *See, e.g.*, Walter A. Effross, *Assaying Computer Associates v. Altai: How Will The "Golden Nugget" Test Pan Out?*, 19 RUTGERS COMPUTER & TECH. L.J. 1, 27–32 (1993).

94. Computer Associates Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 705 (2d Cir. 1992).

95. *Id.* at 707 (citation omitted).

96. *Id.* at 705 ("[A] computer program's ultimate function or purpose is the composite result of interacting subroutines. Since each subroutine is itself a program, and thus, may be said to have its own 'idea'. . . .").

97. *Id.* at 706.

98. *Id.* at 707 ("This [filtration] process entails examining the structural components at each level of abstraction to determine whether their particular inclusion at that level was 'idea' or was dictated by considerations of efficiency, so as to be necessarily incidental to that idea; required by factors external to the program itself; or taken from the public domain and hence is

The filtration step is based on two well-established doctrines in copyright law. First, the filtration step is based on the merger doctrine articulated in *Baker v. Selden*.[99] The merger doctrine holds that "[w]hen there is essentially only one way to express an idea, the idea and its expression are inseparable and copyright is no bar to copying that expression."[100] Second, the filtration step is based on the scènes-à-faire doctrine. The scènes-à-faire doctrine excludes from copyright protection expressions that are "as a practical matter, indispensable or at least standard in the treatment of a given [idea]."[101] In the software context, scènes à faire consist of standard techniques and widely-accepted programming practices that circumscribe a programmer's freedom of design choice.[102] While the scènes-à-faire doctrine relate to contexts that less severely circumscribe a programmer's freedom of design choice, the merger and the scènes-à-faire doctrines are both fundamentally concerned about design choice. This fundamental concern is an undercurrent in Christopher Alexander's writings on architecture which have profoundly influenced the practice of software engineering.

B. *Critique of* Altai

1. Architecture of Buildings: The Structure of Physical Space

Christopher Alexander, a noted architect, believed that the ancient problems of structuring physical space are often susceptible to invariant patterns of solutions that are as timeless as they are unconscious. In *The Timeless Way of Building*, the first volume of his three-volume treatise, Alexander sought to articulate this sense of timelessness.[103] He observed:

---

nonprotectable expression.").

99. Baker v. Selden, 101 U.S. 99, 103 (1879) ("And where the art it teaches cannot be used without employing the methods and diagrams used to illustrate the book, or such as are similar to them, such methods and diagrams are to be considered as *necessary incidents* to the art, and given therewith to the public . . . .") (emphasis added).

100. Computer Associates Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 707–08 (2d Cir. 1992) (quoting Concrete Machinery Co. v. Classic Lawn Ornaments, Inc., 843 F.2d 600, 606 (1st Cir. 1988)).

101. Data E. USA, Inc. v. Epyx, Inc., 862 F.2d 204, 208 (9th Cir. 1988) (citing Atari, Inc. v. N. Am. Philips Consumer Electronics Corp., 672 F.2d 607, 616 (7th Cir. 1982)).

102. *Altai*, 982 F.2d at 709–10 ("[A] programmer's freedom of design choice is often circumscribed by extrinsic considerations such as . . . widely accepted programming practices within the computer industry.").

103. CHRISTOPHER ALEXANDER, THE TIMELESS WAY OF BUILDING (1979).

> There is one timeless way of building. It is a thousand years old, and the same today as it has ever been. The great traditional buildings of the past, the villages and tents and temples in which man feels at home, have always been made by people who were very close to the center of this way. . . . This does not mean that all ways of making buildings are identical. It means that at the core of all successful acts of building and at the core of all successful processes of growth, even though there are a million different versions of these acts and processes, there is one fundamental invariant feature, which is responsible.[104]

In the second volume, *A Pattern Language* (1977), Alexander formulated a pattern language to articulate these unconscious patterns that are "archetypal—so deep, so deeply rooted in the nature of things, that it seems likely that they will be a part of human nature, and human action, as much in five hundred years, as they are today."[105]

Alexander's work profoundly influenced the field of computer science. Alexander studied mathematics and physics at Cambridge University before pursuing advanced studies in architecture at Harvard University. *A Pattern Language* and *The Timeless Way of Building* are almost always cited as the genesis for the concept of patterns in a paradigmatic approach to computer programming that had become the dominant methodology by the mid-1990s.[106] Just as Alexander et al. had proposed 253 patterns that organize physical space, the "The Gang of Four" (1995) proposed 23 design patterns or abstractions that structure conceptual space.[107] The design patterns formulated by "The Gang of Four" are recurring solutions to common problems in software engineering. Scholars have noted the similarities between Alexander's pattern language and the notion of generative grammar for natural languages.[108] Just as the rules of grammar can generate an infinite variety of sentences, so the rules of a pattern language can generate many possible structuring of physical or conceptual space.

104. *Id.* at 7–8.

105. CHRISTOPHER ALEXANDER ET AL., A PATTERN LANGUAGE: TOWNS, BUILDINGS, CONSTRUCTION xvii (1977).

106. LESLIE J. WAGUESPACK, THRIVING SYSTEMS THEORY AND METAPHOR-DRIVEN MODELING 9 (2010).

107. ERICH GAMMA, RICHARD HELM, RAPLH JOHNSON, & JOHN VLISSIDES, DESIGN PATTERNS: ELEMENTS OF REUSABLE OBJECT-ORIENTED SOFTWARE 8–9 (1995) (providing a brief overview of 23 design patterns discussed in the book) [hereinafter GoF].

108. *See, e.g.*, Juval Portugali, *Inter-representation Networks and Cognitive Mapping*, *in* THE CONSTRUCTION OF COGNITIVE MAPS 11, 32 (Juval Portugali ed., 1996). Generative grammar originates in the seminal work of linguist Noam Chomsky.

Alexander is critical of the import of his notion of pattern language in/into computer science as merely a vehicle of communication, allowing programmers "to write down good ideas about software design in a way that can be discussed, shared, modified, and so forth."[109] Alexander clarifies:

> However, that is not all that pattern languages are supposed to do. The pattern language that we began creating in the 1970s had other essential features . . . it allows people to create coherence, morally sound objects, and encourages and enables this process because of its emphasis on the coherence of the created whole.[110]

Of significance is Alexander's emphasis on wholeness—a concept that is at once elusive and fundamental.

According to Alexander, the structure of physical space involves a fundamental circularity.[111] While the profound implications that Alexander draws from this observation are beyond the scope of this article,[112] what directly concerns us is his conception of *wholeness* in architecture. According to Alexander, "wholeness is a structure of great subtlety . . . *it is useless of think of it as a relationship 'among the parts'* . . . . it is the wholeness which *creates* the parts."[113] For example,

---

109. Christopher Alexander, *The Origins of Pattern Theory: The Future of the Theory, and the Generation of a Living World*, 16 SOFTWARE, IEEE 71, 74 (Sep/Oct 1999).

110. *Id.*

111. "This circularity is not a mistake, or an indication of something logically vicious in the argument. *On the contrary, it is the essential feature of the situation.* Our understanding of both wholeness and life will come into focus at that moment when we thoroughly grasp this circularity and what it means." CHRISTOPHER ALEXANDER, THE NATURE OF ORDER: AN ESSAY ON THE ART OF BUILDING AND THE NATURE OF THE UNIVERSE, BOOK 1—THE PHENOMENON OF LIFE 118 (2002).

112. For example, contrary to the modernist vision of urban planning, as articulated by influential architect Le Corbusier, Alexander favors piecemeal, organic growth, a process which allows the whole to emerge gradually from local acts. *See also* RICHARD P. GABRIEL, PATTERNS OF SOFTWARE: TALES FROM THE SOFTWARE COMMUNITY 14 (1996) ("[H]ow do you enable a programmer to feel responsible for software developed earlier? Here is where habitability comes in. Just as with a house, you don't have to have built or designed something to feel at home in it. Most people buy houses that have been built and designed by someone else. These homes are habitable because they are designed for habitation by people, and peoples' needs are relatively similar. As I said earlier, a New England farmhouse is habitable, and the new owner feels just as comfortable changing or adapting that farmhouse as the first farmer was. But a home designed by Frank Lloyd Wright—though more habitable than most 'overdesigned' homes—cannot be altered because all its parts are too rigidly designed and built. The needs of the whole have overshadowed the needs of the parts and the needs of the inhabitants. Finally, if Alexander's lesson applies to software, it implies that a development project ought to have less of a plan in place than current thinking allows. This provides a mechanism for motivation and a sense of responsibility to those developers who later must work with the code.").

113. ALEXANDER, THE NATURE OF ORDER, *supra* note 111, at 86. *See also* Feenberg, *supra* note 23, at 3–4 ("The apparent origin of complex wholes lies in their parts but, paradoxical though

Alexander writes, "[t]he flower is not made *from* petals. The petals are made from their role and position in the flower."[114] In other words, the primacy or presence of the whole can be found in each of its parts. Similarly, in writing/reading and in speaking/listening, the significance of words in a sentence arises from their role and position in a sentence.[115] Alexander characterizes this phenomenon as a recursion or interpenetration.[116]

Wholeness has an elusive quality that reveals the limits of language. Alexander writes:

> Words fail to capture it because it is much more precise than any word. The quality itself is sharp, exact, with no looseness in it whatsoever. But each word you choose to capture it has fuzzy edges and extensions which blur the central meaning of the quality . . . . Imagine the quality without a name as a point, and each of the words which we have tried as an ellipse. Each ellipse includes this point. But each ellipse also covers many other meanings, which are distant from this point.[117]

According to Alexander, "[t]he word which we most often use to talk about the quality without a name is the word 'alive.'"[118]

Similarly, courts have employed the rhetoric of *life* and *vitality* in characterizing patent claims and technological expressions.[119] While

it seems, in reality the parts find their origin in the whole to which they belong . . . . The part can be separated from the whole but it then loses its function. A tire that has been removed from a car continues to be a tire but it cannot do the things tires are meant to do . . . it is easy to see that the form and even the existence of tires such as we know them depends on the whole car they are destined to serve. . . . The car is not just assembled from pre-existing parts since the nature of the parts is derived from the design of the car and vice versa.").

114. ALEXANDER, THE NATURE OF ORDER, *supra* note 111, at 87–88.

115. This semantic phenomenon is illustrated by a story of a British statesman, known as Lord Palmerston. While addressing an audience, Palmerston was interrupted with a question whether he will vote for a certain reform. Palmerston answered, "I will" while some audience cheered. He then added "not" while others hailed. Then he concluded with "tell you!" after which the whole audience applauded. *See* 33 PRAIRIE FARMER 827 (1866).

116. ALEXANDER, THE NATURE OF ORDER, *supra* note 111, at 116–118.

117. ALEXANDER, THE TIMELESS WAY OF BUILDINGS, *supra* note 103, at 39.

118. *Id.* at 29. Interestingly, in thinking about the difference between biology and (simple) engineered machines, John von Neumann struggled with a concept that has "no adequate name," a concept that was later named "complexity." JOHN VON NEUMANN, THEORY OF SELF-REPRODUCING AUTOMATA 78 (Arthur W. Burks ed., 1966). Von Neumann posited a complexity threshold beyond which a system changes qualitatively rather than merely quantitatively. *Id.* at 80.

119. For example, consider the jurisprudence on the interpretation of preambular language. *See, e.g.*, Schram Glass Mfg. Co. v. Homer Brooke Glass Co., 249 F. 228, 233 (7th Cir. 1918) ("[the preamble] may so affect the enumerated elements [of a claim] as to give life and meaning and vitality to them, as they appear in the combination."); Kropa v. Robie, 187 F.2d 150, 152 (C.C.P.A. 1951) ("[T]he preamble was considered necessary to give life, meaning and vitality to

such terms are arguably inexact, the terms describe an undeniable intuition about the nature of human expressions.[120] Fundamentally, human expressions—whether architectural or technological—are vital precisely because they arise from a fertile ground of alternatives (which are themselves at once intellectual fruits and soil).[121] Understandably, then, computer science professor Waguespack submits that the term *choice* serves well for the translation of Alexander's theories for physical space to the conceptual, cognitive space of information and computer systems.[122] Professor Waguespack observed:

> Choices are the centers that lie at the root of life in information systems. Choices address different aspects of system abstraction at different points or stages in system development. A choice by nature admits to alternatives and the prospect of reconsideration when an unfolding context of experience and understanding merits it. Alexander uses this term, unfolding, repeatedly to explain the evolution of an architectural conception toward a useful intensification of life. In this sense, a living information system model unfolds revealing a continuity of structure and function and consonance with the context within which it is intended to serve.[123]

"In Alexander's terms," Waguespack writes, "the degree to which these *choices* contribute to the whole (system) determines to what degree the system has *life*."[124]

---

the claims or counts."); *see also* Computer Associates Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 714 (2d Cir. 1992) (". . . qualitatively vital aspect of the plaintiff's protectable expression.").

120. Alexander's empirical research over 30 years yields the extraordinary fact that the degree of perceived life in things is an objective, measurable quality that transcends culture. CHRISTOPHER ALEXANDER, EMPIRICAL FINDINGS FROM THE NATURE OF ORDER 3 (2006), *available at* http://www.livingneighborhoods.org/library/empirical-findings.pdf ("Degree of life is an objective quality that may be measured by reliable empirical methods. The empirical test that most trenchantly predicts 'life' in things, in comparing two things, is a test that asks which of the two induces the greater wholeness in the observer.").

121. Based on empirical findings, Alexander concluded that the degree of life in things is correlated to 15 properties (e.g., "local symmetries," "deep interlock," "graded variation," etc.) that are generative. *Id.*

122. Underscoring this critical point, Professor Allen observes that "there is never one single solution to an engineering problem . . . [t]hat, I think, touches the heart of the matter, which is the availability and necessity of choice. Engineering is pervaded by choice. There are alternative ways to design any component of any system, with systematic implications for the rest of the design. Every technological problem has alternative solutions, and the more engineering there is, the more alternatives to choose from. One might even say that modern engineering does not exist until there is a relative density of technical alternatives and an art of technological choice." BARRY ALLEN, ARTIFICE AND DESIGN, ART AND TECHNOLOGY IN HUMAN EXPERIENCE 130 (2008).

123. WAGUESPACK, *supra* note 106, at 11–12.

124. *Id.* at 12.

Alexander's conception of *wholeness* suggests an approach to thinking about the well-accepted principle, as articulated by *Mayo*, that the whole must exceed the sum of the parts.[125] If the principle is formulated as a question (i.e., does the whole exceed the sum of the parts taken separately?), the question can be understood as a *constitutive* question. The question presupposes a naïve ontology (of technological parts and wholes) in order to make possible an inquiry. The constitutive question posits a given (i.e., the whole may exceed the sum of the parts), the impossibility of which the interrogator is assigned to investigate.[126]

More fruitfully, if the principle can be understood as calling for an operative action, the principle calls for a kind of explicative unfolding. What is unfolded is the interpenetration of the whole and its constituent parts. What is unfolded is the internal coherence of an architecture and the significance of the choices that structure the architecture as a whole. The recursion, it follows, is a kind of hermeneutic circle in which a better understanding of the parts leads to a more penetrating perception of the whole, which, in turn, deepens understanding of the parts.[127] The excess, it follows, has a diachronic rather than a synchronic character. The excess is inseparable from the deepening process of understanding and cannot exist apart from the inquiry. Stated differently, the principle does not call for a comparison of ready-made and isolatable quantities (i.e., the parts themselves, taken separately), or any measures derived therefrom (e.g., emergent features and functionality).[128] The principle calls not for arithmetic but for explication.[129]

125. *See supra* notes 63–64 and accompanying text.

126. Similarly, "[t]he social scientist asks, 'How is social order possible?' The form of the question…is naïve, not skeptical . . . it suppresses the moment of skepticism in order to constitute an entity, called social order, capable of being investigated." *See* WILLIAM RASCH, NIKLAS LUHMANN'S MODERNITY, THE PARADOXES OF DIFFERENTIATION 48–49 (2000) (observing that "the disciplinary question does more than just register the complexity it studies; it implicates itself in that complexity in order to allow for its own continued existence . . . . A discipline, a system, that could adequately and finally answer the question it poses for itself would cease to exist.") (citation omitted).

127. *See, e.g.*, *supra* note 115.

128. The notion of emergent functionality or property does not provide a discriminating criterion for the principle that the whole must exceed the sum of the parts. Emergent properties, especially for software, are found in every purposive combination of elements. Consider the following example. A byte (i.e., 8 bits) can encode 2^8 or 256 different values. A combination of 2 bytes (i.e., 16 bits) can encode 2^{16} or 65536 values. The whole (2^{16}) is more than the sum of the parts taken separately ($2^8 + 2^8$). The emergent property of the whole arises not because the basic functions of the bytes have changed. Rather, the primacy of the whole (i.e., two-byte encoding) induces a significance in the parts (i.e., higher-order byte vs. lower-order byte).

129. *But see* Great Atl. & Pac. Tea Co. v. Supermarket Equip. Corp., 340 U.S. 147, 152 (1950) ("The conjunction or concert of known elements must contribute something; only when

An explicative unfolding of the whole reveals the significance—the operative term[130]—of particular choices. Indeed, fundamentally at issue in *Alice* and in *Altai* are design choices (and their significance) in the creative process of writing software that is revealed, as the following parts show, by an unfolding of the software architectures at issue.

2. Architecture of Software: The Structure of Conceptual Space

Part II.B.2 explicates the software architecture at issue in *Altai*, illustrating the creative process of writing software, to reveal two difficulties that eluded the court—*what* is an abstraction and *why* code is abstracted in particular ways.

Computer Associates' software, CA-SCHEDULER, is a job scheduling program designed to create a schedule specifying when the computer should run various tasks, and then to control the computer as it executes the schedule.[131] CA-SCHEDULER is designed for the IBM System 370 family of computers, which may support one of three operating systems: DOS/VSE, MVS, or CMS.[132] CA-SCHEDULER contains a first component that contains only the task-specific portions of the program, independent of the operating system, and a second component, ADAPTOR, which contains all the interconnections between the first component and the operating system.[133] Because software written for one operating system will not, without modification, run under another operating system, the ADAPTOR component allows reuse of the first component by translating "system

---

the whole in some way exceeds the sum of its parts is the accumulation of old devices patentable . . . . Neither court below has made any finding that old elements which made up this device perform any additional or different function in the combination than they perform out of it. . . . *Two and two have been added together, and still they make only four*.") (emphasis added). However, the Court never inquired into precisely why "the resultant device words [sic] as claimed, speeds the customer on his way, reduces checking costs for the merchant, has been widely adopted and successfully used . . . ." *Id.* at 149. The Court never inquired into the significance of the constituent parts in light of the whole. *Id.* at 152 ("This counter does what a store counter always has done—it supports merchandise at a convenient height while the customer makes his purchases and the merchant his sales. The three-sided rack will draw or push goods put within it from one place to another—just what any such a rack would do on any smooth surface—and the guide rails keep it from falling or sliding off from the counter, as guide rails have ever done.").

130. Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1298 (2012) (". . . those steps, when viewed as a whole, add nothing significant beyond the sum of their parts taken separately.").

131. Computer Associates Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 698 (2d Cir. 1992).

132. *Id.*

133. *Id.* at 699.

calls" into the language of a particular operating system.[134] Altai hired Arney, a former employee of Computer Associates, to help rewrite its own job scheduling software, ZEKE, so that it could be run on the MVS operating system.[135] Arney introduced the "common system interface" architecture of CA-SCHEDULER into ZEKE. Arney also used approximately 30% of ADAPTOR's source code to create a similar component, OSCAR, which allows ZEKE to interface with the MVS operating system.[136]

The court in *Altai* largely agreed with and reproduced the District Court's analytical approach.[137] In applying the abstraction step, the court found that the program's overall structure comprises of the following levels of increasing abstraction: object code, source code, parameter lists, macros, lists of services, and organizational charts.[138] In applying the filtration step, the court determined that most of the parameter lists and macros were either in the public domain or dictated by the functional demands of the program.[139] In applying the comparison step, the court held that the few remaining parameter lists and macros are not protectable by copyright given their relative contribution to the overall program.[140] The court also held that the organizational charts are non-protectable because the charts "follow naturally from the work's theme rather than from the author's creativity."[141]

Altai has been overwhelmingly well-received by scholars and jurists. For determining substantial similarity in the non-literal aspects of computer programs, courts have adopted the analysis of the abstraction-filtration-comparison test.[142] *Altai* has also been endorsed by courts in Canada, the United Kingdom, and France.[143]

134. *Id.*

135. *Id.*

136. *Id.* at 700.

137. Computer Associates Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 714 (2d Cir. 1992) ("We note that [District Court] Judge Pratt's method of analysis effectively served as a road map for our own, with one exception. . . . We think that our approach—i.e., filtering out the unprotected aspects of an allegedly infringed program and then comparing the end product to the structure of the suspect program—is preferable . . . .").

138. *Id.* at 714.

139. *Id.*

140. *Id.* at 714–715.

141. *Id.* at 715 (quoting 3 Nimmer § 13.03 [F][3], at 13–65).

142. Mark A. Lemley, *Convergence in the Law of Software Copyright?*, 10 HIGH TECH. L.J. 1, 15 (1995) ("In the two and one-half years since *Altai* was decided, the Second Circuit's filtration approach has been endorsed by the Federal Circuit, the Fifth Circuit, the Ninth Circuit, the Tenth Circuit, and district courts in the Eleventh Circuit.") (citations omitted).

143. *Id.*

However, remaining largely obscured are two difficulties, which will be discussed in turn below. The first difficulty involves the question of *what* can properly be characterized as abstractions. For example, well-known in the art of computer software, parameter lists and macros are *not* abstractions of code. Rather, they are input that affect the run-time behavior of code. One can reasonably surmise that the parameter lists for both ADAPTOR and OSCAR include parameters that relate to the requirements of particular operating systems or particular tasks. One can also surmise that the macros define the run-time behavior of the interface or the executed tasks. Therefore, properly characterized as input, parameter lists and macros are *not* abstractions of code.[144] In the Reply Brief, Plaintiff-Appellant correctly argued that "parameter lists and macros are not in any sense abstractions of code, but are entirely different aspects of a computer program."[145] However, holding otherwise, the Plaintiff-Appellant argued, the court "provides no rationale for its actions and provides no guidance for reasoned decision-making in the future."[146] Consequently, despite the widespread adoption of the *Altai* test, courts have applied it inconsistently.[147]

Properly applied, the abstraction step of the *Altai* test should mirror the creative process of writing code. The abstraction step should "*retrace and map each of the designer's steps—in the opposite order in which they were taken during the program's creation.*"[148] Therefore, the abstraction step suggests a definition of *abstraction* that is semantically manifold—*abstraction* is at once a noun and a verb. In other words, abstractions are the products of the act of abstraction.

Abstractions are not interpretations that are derived a posteriori (even if those abstractions are themselves conditions for the possibility of further abstractions and creativity). A computer program, as a technical artifact, can surely be abstracted, interpreted, parsed in many, if not countless, possible ways. For example, a parameter list may be interpreted as abstracted code which is utilized for instantiating

144. Computer Associates Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 697–98 (2d Cir. 1992) ("A parameter list, according to the expert appointed and fully credited by the district court, Dr. Randall Davis, is 'the information sent to and received from a subroutine.' See Report of Dr. Randall Davis, at 12. The term 'parameter list' refers to the form in which information is passed between modules (e.g. for accounts receivable, the designated time frame and particular customer identifying number) and the information's actual content . . . .").

145. Reply Brief of Plaintiff-Appellant at 4, Computer Associates Int'l, Plaintiff-Appellant, v. Altai, Inc., Defendant-Appellee., No. 91-7893, 1991 WL 11010234 (C.A.2).

146. *Id.* at 5.

147. *See* Lemley, *infra* note 151–152 and accompanying text.

148. *Altai*, 982 F.2d at 707 (emphasis added).

variables during the execution of the computer program.[149] Similarly, "a computer program can often be parsed into at least six levels of generally declining abstraction: (i) the main purpose, (ii) the program structure or architecture, (iii) modules, (iv) algorithms and data structures, (v) source code, and (vi) object code."[150] However, ad hoc interpretations of a computer program result in inconsistent applications of the *Altai* test, despite a superficial appearance of convergence, as argued by Professor Lemley.[151]

Ad hoc interpretations of computer programs also diminish the potency of the term *abstraction*. While Professor Lemley is correct in arguing that "[c]ourts adjust the [*Altai*] test to fit their particular view of programming . . . in the computer cases everyone seems to have forgotten that the abstraction test is only a judicial construct."[152] However, contrary to Professor Lemley's argument, in the software context, unlike the literary context, *abstraction* is not an artificial construct.[153] As the following pages show, the creative process of writing software can be understood as a formal process of abstraction. Thus, a narrower and more fruitful definition of *abstraction* (i.e., as the product of the act of abstraction in the process of creation) allows for an inquiry into creativity to enter directly into and clarify the eligibility

149. Thanks to Professor Karjala for this interesting perspective. E-mail from Professor Dennis S. Karjala, Jack E. Brown Professor of Law, ASU College of Law, to Scott T. Luan (Sept. 4, 2014, 17:53 EST) (on file with author).

150. Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823, 835 (10th Cir. 1993) (citations omitted). Such an interpretation of computer programs is problematic because *inter alia* it confuses categories of concept (e.g., algorithms, architecture) with methods of implementation (e.g., modules). For example, an algorithm (or data structure) may utilize functions (or data types) from a plurality of modules or libraries that are linked into an executable program. Admittedly, *modules*, as a term of art, is ambiguous. However, the point remains valid regardless of whether *modules* is construed to be packages, libraries, source files, subroutines, or code segments.

151. Mark A. Lemley, *Convergence in the Law of Software Copyright?*, 10 HIGH TECH. L.J. 1, 3 (1995) ("Having finally resolved the debate that has been plaguing software copyright law since its inception, courts are discovering to their chagrin, that deciding *what* test to apply actually tells you very little about *how* to apply that test. Despite the convergence of courts on *Altai*'s filtration approach, courts remain fundamentally conflicted in deciding how broadly to protect software copyright.").

152. *Id.* at 20–21 (arguing that "[a]ll of this debate [about the proper level of abstraction] misconstrues and unnecessarily complicates the abstraction step of the *Altai* test. The problem is that lawyers and courts take the abstraction step too literally, forgetting that it is intended to be, in fact, an *abstraction*.").

153. *Id.* at 20 (submitting that "in the literary context we all know that this 'higher level of abstraction' is merely an artificial construct to help us decide the case. There may or may not be a 'chapter outline' and a 'paragraph outline' which were used in writing the copyrighted work. But it doesn't really matter. Certainly, the inquiry is not whether the defendant had access to and copied from the outline, but whether he copied *from the finished work at that particular level of generality*.").

determination.[154] To understand why, we would have to consider a second difficulty.

The second difficulty is not as easily clarified. The first difficulty relates to the question of *what* is an abstraction. By contrast, the second difficulty involves the related questions of *how* and *why* code is abstracted in particular ways during the process of creation. The second difficulty is best approached by way of examples from Alexander's *A Pattern Language* and the seminal book on software design patterns by "The Gang of Four." The nature of the material requires a treatment that is unavoidably technical. However, I believe that the material allows for simplification and suggestion without risking technical inaccuracies.

Note that my approach is explicatory rather than based on analysis of actual source code. The intent of this part is to reveal the technical possibilities and software design choices even though Arney, the engineer of the software at issue in *Altai*, may not have explicitly utilized the pattern language formulated by the "Gang of Four." Similarly, Alexander's pattern language of architectural design describes *timeless* solutions to recurring problems even though architects from past centuries had not relied on Alexander's specific formulations. I believe this approach most clearly illustrates the process of creativity, a process that has eluded legal scholars and jurists. The basic argument is that software design choices are *at once* subjective and objective, satisfying *both* cognitive and machine requirements. The argument is supported by additional references to a classic paper in computer science by Stanford University professor Donald Knuth[155] and to the well-established engineering practice of "code refactoring."[156]

154. *See infra* note 181.

155. *See infra* notes 169–177 and accompanying text.

156. *See infra* note 178.

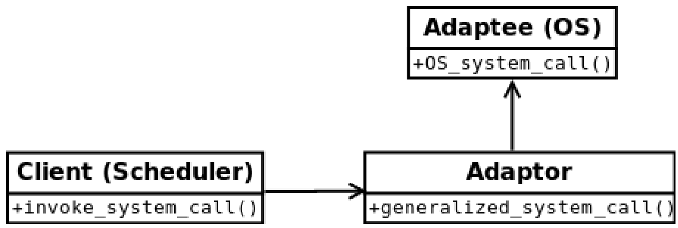

Figure 1: Adapter Pattern (based on GoF, 139)

The Adapter pattern, as depicted in Fig. 1, describes the high-level structure of the ADAPTOR component. The pattern interfaces the *Client* with the *Adaptee*. For Computer Associates' software, the SCHEDULER corresponds to the *Client* and the host operating system (OS) corresponds to the *Adaptee*. Utilizing a generalized system-call interface, the *Adaptor* translates the request by the *Client* into an OS-specific system call. Thus, the Adapter pattern provides a system-agnostic interface such that SCHEDULER can invoke an unchanging set of system calls regardless of the particular host operating system. Accordingly, SCHEDULER may be reused in different operating systems without changes to the source code.

The Adapter pattern structures conceptual space just as Alexander's Arcade pattern structures physical space. Reflecting on the archetypal, deeply-rooted nature of certain timeless patterns, Alexander noted that "[w]e doubt very much whether anyone could construct a valid pattern language, in his own mind, which did not include the pattern ARCADES . . . for example, or the pattern ALCOVES . . . ."[157] According to Alexander, one recurring problem in architecture is the possibility that "there are no strong connections between the territorial world within the building and the purely public world outside. There are no realms between the two kinds of spaces which are ambiguously a part of each—places that are both characteristic of the territory inside, and simultaneously, part of the public world."[158] Consequently, buildings are often much more unfriendly than they need to be.[159] The arcade, according to Alexander, is the most simple and beautiful way of structuring physical space such that the space contains the character of a building's interior while simultaneously being open to the public world. Similarly, the Adapter

157. ALEXANDER ET AL., A PATTERN OF LANGUAGE, *supra* note 105, at xvii.

158. *Id.* at 581.

159. *Id.*

pattern structures cognitive, conceptual space such that two digital realms[160] are connected by an interface that is a part of each.

The system calls invoked by the *Client* may comprise complex commands with various parameters. A simple, hypothetical example is illustrative. A command that saves data to file requires the actual data, the file name, and the directory path to the file. Other parameters may relate to formatting, encryption, and scheduling requirements. These parameters may be included as input in parameter lists, as in the SCHEDULER software. During execution of the program when various commands are invoked, these parameters are passed to the appropriate components.

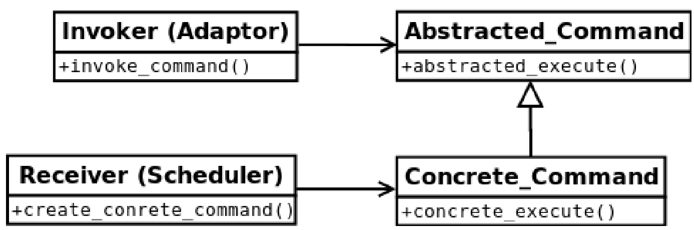

Figure 2. Command Pattern (based on GoF, 233)

The Command pattern, as depicted in Fig. 2, allows the programmer to encapsulate all the information needed to execute a particular command. For Computer Associates' software, the *Invoker* corresponds to the ADAPTOR and the *Receiver* corresponds to the SCHEDULER. The *Receiver* creates a *Concrete_Command* or "command object" as an instantiation of the *Abstracted_Command*. The *Concrete_Command* encapsulates all the particularities of the specific command and provides the *Invoker* with a simple mechanism, *abstracted_execute()*, that is common to all "command objects." The *Invoker* (i.e., ADAPTOR) simply executes the *abstracted_execute()* function from a high-level perspective (*i.e.,* without knowledge of the underlying implementation of *Concrete_Command*) after deciding which commands to execute based on their scheduling requirements as determined by *Receiver* (i.e., SCHEDULER).

The Command pattern structures conceptual space just as Alexander's Alcove pattern structures physical space. Alexander observed that the intimate physical space shared by a family is critical.

160. Computer Associates' ADAPTOR can be understood as an interface between "kernel space" and "user space."

The emotional qualities of a healthy family life "will only come into existence if the members of the house are physically able to be together as a family."[161] Shared spaces that do not allow for members of the family to be together, even when they are doing different things, inevitably encourages the habit of doing things apart. According to Alexander, the alcove is a fundamental pattern that reconciles the "opposing needs for some seclusion and some community at the same time in the same space."[162] Alcoves are small spaces at the periphery of larger family room such that "any clutter that develops in them does not encroach on the communal uses of the main room."[163] Thus, alcoves structure physical space in a way that reconciles private and shared aspects of family life. Similarly, the Command pattern structures conceptual space in a way that reconciles different aspects of software design. Just as alcoves seclude "clutter," so the Command pattern encapsulates and "hides" the disparate information required to execute a particular command.

In computer science, scheduling disciplines or strategies are algorithms that allocate shared resources to multiple processes.[164] The shared resource may be memory access or cycles of the microprocessor. If multiple processes or commands need to be executed, the scheduling strategy algorithm allocates cycles of the microprocessor so that each command is executed in a timely manner.

---

161. ALEXANDER ET AL., A PATTERN LANGUAGE, *supra* note 105, at 829.

162. *Id.* at 831.

163. *Id.* at 830.

164. Scheduling may be implemented in a variety of ways. For example, the First-Come, First-Served (FCFS) strategy simply adds commands to a list or queue. The commands are executed in the order that they are added to the queue. The Shortest-Job-First (SJF) strategy estimates the remaining execution time for each command and rearranges the command queue such that the command closest to being completed is scheduled for immediate execution. The fixed priority pre-emptive scheduling strategy assigns a fixed priority to each command and rearranges the command queue in order of increasing priority. The command with the highest priority is immediately executed. By contrast, the round-robin scheduling strategy assigns a fixed time unit for each command and cycles through them.

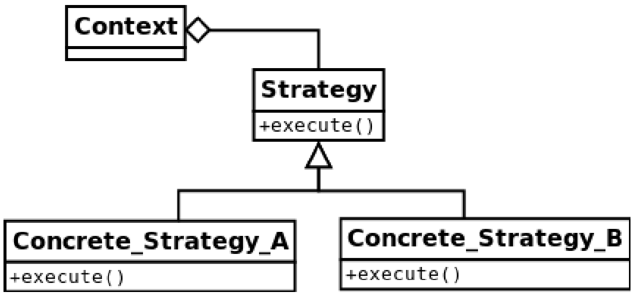

Figure 3. Strategy Pattern (based on GoF, 315)

The Strategy pattern, depicted in Fig. 3, is a design pattern which allows an algorithm's behavior to be modified during runtime. For example, the scheduling algorithm can be modified during runtime in order to select a particular *Concrete_Strategy* based on *Context*. Such a selection of strategy may be understood as a "macro" or "macro-instruction" which operates on or invokes large blocks of code corresponding to a particular strategy. Similarly, the "macros" at issue in *Altai* relate to runtime configuration of software behavior.

For Computer Associates' software, the Strategy pattern may be an abstraction that is implemented in the SCHEDULER component. Such an architecture is conceptually compelling because the responsibility of scheduling can be completely allocated to the SCHEDULER. The ADAPTOR component simply translates the scheduled commands to the particular system calls of the host operating system. Therefore, in this architecture, the Strategy pattern occupies a higher level of abstraction than the Command pattern in the sense that the *Context* or choice of strategy is independent of the particularities of various commands. Thus, such a structuring of conceptual space is consistent with the court's finding in *Altai* that macros are a "higher level of abstraction" than parameter lists.

Alternatively, the Strategy pattern may be an abstraction that is implemented in the ADAPTOR component. Such an architecture has the advantage that scheduling strategies are more attuned to the characteristics of particular host operating systems.[165] In Linux,

165. This statement is not unconditionally true. For example, consider a controversy that has far-reaching implications with respect to the "UNIX philosophy" of implementing "mechanism, not policy," as originated by Ken Thompson. *See The Unix Philosophy: A Brief Introduction*, LINUX INFORMATION PROJECT (Aug. 6, 2006), http://www.linfo.org/unix_philosophy.html. Thompson, in collaboration with Dennis Ritchie, designed and implemented the original Unix operating system at Bell Labs in 1969. The controversy was centered on a "patch" (i.e., an

Microsoft Windows, and other operating systems, the code that schedules processes is found in "kernel space" within the operating system itself (rather than "user space," in which applications such as SCHEDULER are executed).[166] If the scheduling and execution of commands involve time-critical requirements, then the Strategy pattern may be implemented in "kernel space" of ADAPTOR. In this architecture, command parameters may include the particularities of the scheduling strategy. Therefore, given this dependency, the Command pattern and its corresponding parameter lists may be understood as occupying a higher level of abstraction than the Strategy pattern and its corresponding notion of "macros." This design choice yields a result that is contrary to the findings of the court in *Altai*.

The foregoing hypothetical design choices are divergent, resulting in contrary determinations about the levels of abstraction and revealing a deficiency in the court's analysis in *Altai*. The court did not inquire more carefully into the architecture underlying the external aspects of the software. The "macros" and "parameter lists" are not abstractions *per se*. Rather, they are externalizations of a hidden interior, a conceptual space that allows for different possible software architectures that would, in turn, yield contrary determinations by a court. Without a careful inquiry of the creative process, without a careful scrutiny of possible design choices and the resulting abstractions, the court in *Altai* arrived at the conclusory finding that no

upgrade) for the Linux kernel that implemented a scheduling algorithm known as the "Completely Fair Scheduler" (CFS). Based on ideas suggested by Linus Torvalds, the chief architect of the Linux kernel, the CFS was expected to significantly boost multi-tasking performance. However, not everyone was convinced about the value of the kernel patch. In response to a challenge by Torvalds—"Numbers talk, bullshit walks"—Lennart Poettering of Red Hat, a company providing open-source software products, developed an equivalent user-space implementation of CFS. *See Re: [RFC/RFT PATCH v3] Sched: Automated Per Tty Task Groups*, MARC (Nov. 16, 2010), http://marc.info/?l=linux-kernel&m=128993691017646&w=2. Poettering responded: "Here's my super-complex patch btw, to achieve exactly the same thing from userspace without involving any kernel or systemd patching and kernel-side logic." *Id.* The discussion thread reveals that design choices in the development of software are far from being determinable with "mathematical precision." *But see* Marci A. Hamilton & Ted Sabety, *Computer Science Concepts in Copyright Cases: The Path of a Coherent Law*, 10 HARV. J. L. & TECH. 239, 279–80 (1996–1997) (arguing that the answer to the question of implementation can be determined with "mathematical precision."). According to Poettering, "there are a lot of userspace folsk [sic] being afraid of and too lazy to hacking the kernel and hence rather workaround kernel fuckups in userspace then fixing it properly. But you [Torvalds] are doing it the other way round, since userspace gives you the creeps . . . ." *See Re: [RFC/RFT PATCH v3] Sched: Automated Per Tty Task Groups*, MARC (Nov. 16, 2010), http://marc.info/?l=linux-kernel&m=128993691017646&w=2.

166. Thus, ADPATOR may be understood as a translation or interface between "user space" and "kernel space."

weight should be accorded to the program's organization because it is "so simple and obvious."[167]

The foregoing hypothetical design choices reveal an elusive aspect of software. The material and abstractions that are used to build software are of a character such that their functions serve pivotal roles in the process of creation. Just as the architecture of buildings organizes physical space, the architecture of software structures conceptual space with similar notions of encapsulation, modularity, translation, and interface. The design choices in the architecture of a building balance the goal of achieving economy of materials and space with the goal of achieving certain affects among the inhabitants of the building. A poetics of architecture, according to Alexander, appreciates fine balances that achieve both goals within a confined space. Similarly, the design choices in software architecture balance the external requirements of machine performance with the requirements of—for lack of a better term—cognitive efficiency.[168]

This critical point can also be made by reference to a classic paper in computer science by Stanford University professor Donald Knuth.[169] In his attempt to resolve the controversy surrounding "go-to" statements at the dawn of "structured programming,"[170] Knuth clarifies the role of abstractions in the creative process of writing software code. Knuth poses the question: "when do they [go-to statements] correspond to a good abstraction?"[171] Knuth argues that "the presence or absence of *go to* statements is not really the issue. The underlying structure of the program is what counts, and we want only to avoid usages which somehow clutter up the program."[172] Thus, according to Knuth, the real

---

167. Computer Associates Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 715 (2d Cir. 1992).

168. Professor Allen makes the same point in the context of architectural design of the Holland Tunnel. Allen observes that "[o]ne reason the Holland Tunnel ventilation system works so well nearly a century after it was built is that it was designed with maintenance in mind, designed to be accessible, to be moved through and manipulated by a human body in a way that is coherent with our most efficient cognitive preferences." ALLEN, *supra* note 122, at 147.

169. *See* Donald E. Knuth, *Structured Programming with go to Statements*, 6 COMPUTING SURVEYS 261 (1974).

170. "Structured programming" is an approach to programming that aims to improve the readability and quality of computer programs by utilizing various control structures such as if-then conditional statements, iteration loops, and subroutines. *See generally* O.J. DAHL, E. W. DIJKSTRA & C. A. R. HOARE, STRUCTURED PROGRAMMING (1972). Knuth observes that, unlike go-to statements, "sequential composition, iteration, and conditional statements present syntactic structures that the eye can readily assimilate." Knuth, *Structured Programing*, *supra* note 169, at 292.

171. Knuth, *Structured Programing*, *supra* note 169, at 294.

172. *Id.* at 275.

issue is the "*subjective* question of program structure."[173] Knuth warns that "premature emphasis on efficiency [during software development] is a big mistake which may well be the source of most programming complexity and grief."[174] Rather, Knuth argues, "we should strive most of all for a program that is easy to understand."[175] Quoting Sir Charles Antony Richard Hoare, a renowned British computer scientist, Knuth defined structured programming as "the systematic use of abstraction to control a mass of detail, and also a means of documentation which aids program design."[176] According to Knuth, "for each particular problem there seems to be an appropriate level [of abstraction]."[177]

In the comparisons between architectural patterns for software and for buildings, for conceptual and for physical space, we have seen that abstractions are the results of design choices that address both machine and cognitive requirements.[178] The functional purpose of software is not just to achieve an optimality under the constraints of an operative environment, but also to achieve an affect/effect that can be characterized as cognitive—a kind of cognitive efficiency.[179] The

173. *Id.* at 291 (emphasis added); *see also* Donald E. Knuth, *Literate Programming*, 27 IEEE COMPUTER JOURNAL 97, 97 (1984) ("Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a *computer* what to do, let us concentrate rather on explaining to *human beings* what we want a computer to do. The practitioner of literate programming can be regarded as an essayist, whose main concern is with exposition and excellence of style."). This point is underscored by another classic text in computer science which had originally served as an introductory textbook at MIT. *See* ABELSON ET AL., *supra* note 27, at xxii ("Our design of this introductory computer-science subject reflects two major concerns. First, we want to establish the idea that a computer language is not just a way of getting a computer to perform operations but rather that it is a novel formal medium for expressing ideas about methodology. Thus, *programs must be written for people to read, and only incidentally for machines to execute*.") (emphasis added).

174. Knuth, *Structured Programing*, *supra* note 169, at 294.

175. *Id.*

176. *Id.* at 292.

177. *Id.* at 294.

178. The point can also be made by reference to the well-established software development practice of "code refactoring." Introduced in the early 1990s, the practice of code refactoring aims to improve the nonfunctional attributes of software such as code readability and maintainability and to create a more expressive internal architecture that easily allows for future extensions or refinements (i.e., extensibility). *See generally* JOSHUA KERIEVSKY, REFACTORING TO PATTERNS (2004). According to Fowler's canonical text, refactoring is the "process of changing a software system in such a way that it does not alter the external behaviour of the code yet improves its internal structure." *See* MARTIN FOWLER, REFACTORING: IMPROVING THE DESIGN OF EXISTING PROGRAMS xxi (14th prtg. 2004). "Some refactorings remove code redundancy, some raise the level of abstraction, some enhance the reusability, and so on." *See* Tom Mens & Tom Tourwé, *A Survey of Software Refactoring*, 30 IEEE TRANSACTIONS ON SOFTWARE ENGINEERING 126, 130 (Feb. 2004).

179. Utilitarian definitions of technology based on the seemingly objective criteria of efficiency are rendered problematic by the more elusive notion of cognitive efficiency. *See infra*

functional purpose of a computer program is not just to perform in an external environment, but also to structure, during the formation of the program, the conceptual space within the imagination of the programmer. Within the completed form of a computer program are artifacts of its creation. This is a critical point. Stated differently, as more than a set of instructions, code encodes a conceptual space.

The foregoing hypothetical design choices illustrate a possible difficulty encountered by courts during fact-finding. Ideally, code should be, what the art understands as, "self-documenting"—that is, a reader of code should be able to understand the underlying reasoning and choices by reading the code itself without any annotation or commentary. The development and evolution of high-level programming languages have afforded code a "readability" that approximates that of natural languages. With some patience and imagination, such code may likely yield its original intent, as it were. By contrast, older code that remains operative, despite the long absence of its authors, may likely be inscrutable.[180] Even though we know the external effects of such code, as externalized in parameter lists and macros, for example, we cannot access the reasoning and abstractions that structure those effects, a structuring that is determinative of how the various levels of abstraction are ordered and represented. Thus, contrary to Professor Karjala, an inquiry into creativity would clarify (rather than muddle) the determination of subject-matter eligibility.[181] In the final analysis, without understanding the creative process of writing software, courts cannot properly and objectively apply the abstraction step of the *Altai* test. Similarly, without a careful inquiry into the creative process, as the following parts argue, courts cannot

---

note 291.

180. *See, e.g.*, JASON SMITH, ELEMENT DESIGN PATTERNS 5 (2012) ("[S]oftware has a peculiar trait of living long past its expected lifetime. COBOL is still a force to be reckoned with in business systems around the globe. Fortran still performs much of the computation in the world's scientific modeling software. Currently shipping major high-performance computer systems have code embedded deep in their firmware that was first created three decades or more ago, in assembler or C. You can be almost certain that somewhere in the millions of lines of implementation that came with your latest personal computer acquisition lies a piece of source code that no person currently understands.").

181. *See* Feenberg, *supra* note 23, at 7 ("No device emerged full blown from the logic of its functioning. Every process of development is fraught with contingencies, choices, alternative possibilities. The perfecting of the technical object obliterates the traces of the labor of its construction. . . . It is this process that adjusts the object to its niche and so the occlusion of its history contributes to the forgetfulness of the whole to which it belongs. I call *this the paradox of the origin: behind everything rational there lies a forgotten history*."). *But see* Dennis S. Karjala, *Copyright and Creativity*, 15 UCLA ENT. L. REV. 169, 173 (2008) ("[C]reativity itself is not analytically useful in making the most basic determination of whether the creativity under consideration falls within copyright subject matter, patent subject matter, or neither.").

properly determine the scope of claims directed to software-implemented inventions.

III. SUBJECT-MATTER ELIGIBILITY IN PATENT LAW

First, this part will revisit the questions posed by *Mayo*, as discussed in Part I. This part argues that *Altai* serves to better harmonize *Flook* and *Diehr*. Second, this part argues that the article's harmonization of *Flook* and *Diehr* represents a special case of a more generalized structure of the eligibility inquiry in patent law. Informed by a generalized conception of the eligibility inquiry, the Part IV critiques *Alice*.

A. Mayo *Revisited*

As discussed in Part I, *Mayo's* harmonization of *Flook* and *Diehr* is part of a developing jurisprudence that has sought to clarify the difference between the patentability and eligibility inquiries. *Mayo's* harmonization poses open questions. For example, open questions remain about whether the larger objective of the eligibility inquiry as a relative functional concern can be based on predictable guiding principles, how the methodological tension between the "as-a-whole" analysis and analytical dissection can be resolved, and how the notion of the level of generality or abstractness should figure in the eligibility inquiry.

Alice does not clarify *Mayo*. *Alice* described the second step in *Mayo's* framework as "a search for an 'inventive concept.'"[182] *Alice* defines "an inventive concept" as "an element or combination of elements that is sufficient to ensure that the patent in practice amounts to *significantly more* than a patent upon the [ineligible concept] itself."[183] According to *Alice*, the "inventive concept" integrates an *abstract idea*—the "building block of human ingenuity"—into "something more."[184] Yet, the critical notions of "abstract idea" and "something more" remain undefined. Further, "the search for the inventive concept" lacks guiding principles. *Alice* merely opined that the elements of a claim should be considered individually *and* as an ordered combination.[185] Methodologically, it is unclear how the two considerations should inform each other.

182. Alice Corp. Pty. v. CLS Bank Int'l, 134 S. Ct. 2347, 2355 (2014).
183. *Id.*
184. *Id*. at 6 (emphasis added).
185. *Id.* at 7.

Patent law can learn from copyright law. Jurisprudence on the analysis of software in copyright law is more receptive to the creative process of writing software, that is, to software as writing.[186] The critique of *Altai* in Part II shows that the creative process involves design choices at various levels of abstraction. By considering design choices, the individual elements are meaningfully integrated as an ordered whole at the appropriate levels of abstraction. Thus, the initial determination of the appropriate level of abstraction is pivotal. As discussed below, this initial determination is the starting point for the harmonization of *Flook* and *Diehr* and for the critique of *Alice*.

At the appropriate levels of abstraction, *Flook* and *Diehr* can be harmonized. The harmonization proceeds by two analytical steps. First, even though the claims at issue in *Flook* and *Diehr* recite steps that simply apply the underlying formulas, the claims should be construed at different levels of abstraction. In *Flook*, the recited steps need not operate on variables in a catalytic conversion process.[187] As a signal-processing algorithm, the "smoothing" formula in *Flook* can be applied to other phenomena (e.g., even to a rubber-molding process in *Diehr*) and to other contexts. By contrast, the underlying formula in *Diehr* is less abstract. In *Diehr*, the underlying formula is a special empirical case of the Arrhenius equation.[188] The recited steps, by definition, operate on specific variables in a rubber-molding process. As a control algorithm, the formula in *Diehr* can only be applied to the thermal-diffusion phenomenon of rubber-curing and to the function of determining cure time. Therefore, because the underlying formula in *Flook* is more abstract than the formula in *Diehr*, the claim at issue in *Flook* should be construed at a higher level of abstraction.

Second, at the appropriate level of abstraction, the recited steps are considered in light of the claim as a whole. In *Flook*, because the underlying formula at issue is applicable to other phenomena and contexts (i.e., other than catalytic conversion), the steps are tangential to the claim as a whole. Just as steps directed to surveying techniques are tangential to a claim applying the Pythagorean Theorem, so steps directed to updating alarm limits for a catalytic conversion process is tangential to a claim applying the "smoothing" formula in *Flook*. Thus,

---

186. This article suggests that understanding of software-as-writing is essential for the understanding of software-as-machine. *See, e.g.,* Pamela Samuelson, Randall Davis, Mitchell D. Kapor, & J.H. Reichman, *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2320 (1994); *see also infra* note 330 ("[S]oftware [is] a machine whose medium of construction happens to be text.").

187. *See supra* note 72 and accompanying text.

188. *See supra* note 79 and accompanying text.

Flook stands for the proposition that a claim reciting steps that are tangential to the claim as a whole is ineligible for patent protection.[189] Claims reciting tangential steps are likely mere efforts in draftsmanship. As such, by definition, these claims are attempts to monopolize the underlying principle and should be ineligible subject-matter.

By contrast, in *Diehr*, because the recited steps are dictated by the terms of the underlying formula which is only applicable to determining cure time in a rubber-molding process, the steps are necessarily incidental to the claim as a whole. Thus, *Diehr* stands for the proposition that a claim reciting steps that are necessarily incidental to the claim as a whole may be eligible for patent protection. Claims reciting incidental steps may be eligible for patent protection if the claims satisfy normative requirements. *Diehr* suggests that the risks of overcuring or undercuring are problems the solutions for which should be promoted by patent law.[190] That *Diehr* did not provide much elaboration does not take away from the normative character of its reasoning. The solution to the problem of overcuring or undercuring rubber can easily be justified on utilitarian grounds in terms of industrial efficiency.[191] Utilitarianism remains the hidden principle in

189. Parker v. Flook, 437 U.S. 584, 590 (1978) ("A competent draftsman could attach some form of post-solution activity to almost any mathematical formula; the Pythagorean theorem would not have been patentable, or partially patentable, because a patent application contained a final step indicating that the formula, when solved, could be usefully applied to existing surveying techniques."). As discussed below, the Court also analogized the claim at issue to "a claim that the formula 2[π]*r* can be usefully applied in determining the circumference of a wheel." *Id. at* 595. Thus, the Court's holding—"if a claim is directed essentially to a method of calculating, using a mathematical formula, even if the solution is for a specific purpose, the claimed method is nonstatutory"—should be limited to (more abstract) formulas the terms of which do not dictate particular applications. *Id.*

190. Diamond v. Diehr, 450 U.S. 175, 187 (1981) ("[I]f the computer use incorporated in the process patent significantly lessens the possibility of 'overcuring' or 'undercuring,' the process as a whole does not thereby become unpatentable subject matter."). However, that is not to say that innovative technologies that prevent failures in catalytic converters should not be promoted by patent law.

191. *Id.* at 188 ("[W]hen a process for curing rubber is devised which incorporates in it a more efficient solution of the equation, that process is at the very least not barred at the threshold by § 101."). Arguably, patent law can also be underwritten by normative principles that serve broader social goods. *See, e.g.*, Diamond v. Chakrabarty, 447 U.S. 303, 315 (1980) ("The subject-matter provisions of the patent law have been cast in broad terms to fulfill the constitutional and statutory goal of promoting 'the Progress of Science and the useful Arts' with all that means for the social and economic benefits envisioned by Jefferson."); Kewanee Oil Co. v. Bicron Corp., 416 U.S. 470, 480 (1974) ("The productive effort thereby fostered will have a positive effect on society through the introduction of new products and processes of manufacture into the economy, and the emanations by way of increased employment and better lives for our citizens."); Brett Frischmann & Mark P. McKenna, *Intergenerational Progress*, 2011 WIS. L. REV. 123, 126–128 (2011) (arguing that Progress should not be measured by a utilitarian calculus that is inherently

Diehr.[192]

B. The General Structure of the Patentable Subject-Matter Eligibility Inquiry

Flook and *Diehr* represent the outer boundaries of the eligibility inquiry. Between *Flook* and *Diehr* spans a spectrum of technical necessity. Between the unnecessary tangential limitations in *Flook* and the necessarily incidental limitations in *Diehr* are those limitations that are directed to meaningful design choices. Choices admit of alternatives. By definition, choices are not strictly dictated by necessity. Yet, design choices are meaningful insofar as their significance is necessarily compelling. Meaningful design choices can utilize conventional patterns of proven solutions. Thus, the spectrum of technical necessity is a continuum spanning limitations that are necessarily incidental (analogous to the merger of *idea* and *expression*), conventional (analogous to *scènes à faire*), unconventional, and tangential. These characterizations are not discrete points in the spectrum of technical necessity but rather are matters of degree.

The foregoing harmonization of *Flook* and *Diehr* suggests a working hypothesis of a general structure of the eligibility inquiry. The structure largely coincides with *Mayo's* framework and is sensitive to the caveats in *Benson* and *Bilski* against approaches that are too rigid or too fine.[193] Three aspects of the general structure will be discussed in turn below.

First, the eligibility inquiry proceeds along two analytical steps. The first step in the eligibility inquiry determines the appropriate level of abstraction. This corresponds to the abstraction step in the *Altai* test. The second step considers, at the appropriate level of abstraction, the constituent elements of a claim in light of the claim as a whole. This corresponds to the filtration step of the *Altai* test. The "search for the

---

short-sighted, but rather Progress should be informed by the notion of inter-generational justice to address the long-term well-being of the species).

192. *See* Feenberg, *supra* note 23, at 12 (submitting an "eighth paradox, which I will *call the paradox of value and fact: values are the facts of the future*. Values are not the opposite of facts, subjective desires with no basis in reality. Values express aspects of reality that have not yet been incorporated into the taken for granted technical environment. That environment was shaped by the values that presided over its creation. Technologies are the crystallized expression of those values.").

193. Eligibility rules risk "freez[ing] process patents to old technologies, leaving no room for the revelations of the new, onrushing technology." Gottschalk v. Benson, 409 U.S. 63, 71 (1972). "[C]ourts may pose questions of such intricacy and refinement that they risk obscuring the larger object of securing patents for valuable inventions without transgressing the public domain." Bilski v. Kappos, 561 U.S. 593, 606.

inventive concept" can be understood as a filtration.[194] Just as the "golden nugget"[195] describes an aspect of software that is eligible for copyright protection, the "inventive concept" describes an aspect of an invention that transforms the claim into subject-matter eligible for patent protection.

Second, the eligibility inquiry has technical and normative aspects.[196] The two aspects inhere in the second step of the eligibility inquiry. As a technical matter, the second step determines the degree of necessity of the recited limitations that apply the underlying principle. As a normative matter, the second step determines whether the claimed invention is the kind of function or technology that should be promoted by patent law. According to *Diehr*, the eligibility inquiry involves a determination whether a claim "when considered as a whole, is performing a *function* which the patent laws were designed to

---

194. Alice Corp. Pty. v. CLS Bank Int'l, 134 S. Ct. 2347, 2355 (2014).

195. Computer Associates Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 710 (2d Cir. 1992) ("Once a court has sifted out all elements of the allegedly infringed program which are 'ideas' or are dictated by efficiency or external factors, or taken from the public domain, there may remain a core of protectable expression. In terms of a work's copyright value, this is the golden nugget."). While the comparison between a "golden nugget" and an "inventive concept" highlights the analytical similarities between copyright and patent law, the comparison is not flawless since, with only a few exceptions, copyright has always eschewed protection of functional works. *See, e.g.*, Dennis S. Karjala, *A Coherent Theory For The Copyright Protection Of Computer Software And Recent Judicial Interpretations*, 66 U. CIN. L. REV. 53, 80–81 (1998) (arguing that "few nonliteral elements of value can survive the filtering process follows directly from an honest reading of *Computer Associates* . after nonliteral elements related to efficiency, compatibility, and functionality have been filtered out, it is difficult to see how anything remaining could even be important . . . .").

196. These two aspects are more clearly revealed or distinguishable in patent law than in copyright law perhaps because patent law confronts more directly or problematically the dual nature of software as writing and as machine. *See infra* note 329; *See also* Pamela Samuelson, Benson *Revisited: The Case Against Patent Protection for Algorithms and Other Computer-Related Inventions*, 39 EMORY L. J. 1025, 1128–30 (1990) (observing that "[w]ith von Neumann's discovery, machines could become writings, and writings could become machines. It is no wonder, then, that the existing intellectual property law systems have found it so difficult to respond to the phenomenon of computer programs. Programs are, in truth, too much of a mechanical process to fit comfortably in the copyright system and too much of a writing to fit comfortably in the patent system. They are a hybrid—both writing and machine at the same time—in a legal system that has generally assumed that an intellectual product is either a writing (and hence copyrightable) or a machine (and hence patentable), but not both at the same time. . . . Patent law, which has traditionally regarded writings and the intellection they embody as outside its domain, has slowly, and with mixed results, moved toward protecting the valuable abstractions embodied in programs. However, just as judges in copyright cases have found it difficult to perceive and respond to the functionality of programs, judges in patent cases have ignored the written and symbolic character of software. Neither copyright nor patent law has been able to come to terms with the essential nature of computer programs, and that is why the law in this area is so confusing.").

protect."[197] Stated differently, determining whether an invention is the kind of function or technology that should be promoted by patent law requires ascertaining first what the purported invention is. (And ascertaining what the purported invention is requires a determination of the appropriate level of abstraction.)

Third and lastly, in patent law, the technical and normative aspects of the eligibility inquiry are related according to the following lines of analysis. First, as discussed above, claimed limitations that are technically tangential to the underlying principle compel, by definition, a normative determination of ineligibility. *Flook* exemplifies this first line of analysis. Second, claims reciting limitations that are necessarily and technically incidental to the underlying principle may or may not be eligible, as a normative matter. The patent-eligible claim in *Diehr* and the patent-ineligible claim in *Mayo* exemplify contrary normative determinations in this second line of analysis.[198] The normative determination, as a relative functional concern, weighs the contribution and creative value of the claimed invention against the risks of pre-emption.[199] The former outweighs the latter in the *Diehr*[200] while the latter outweighs the former in *Mayo*.[201] However, in this second line of analysis, for both *Diehr* and *Mayo*, the normative (rather than the technical) question is dispositive. Third, claims reciting limitations that are technically meaningful design choices strongly support the case for eligibility. In this third line of analysis, the technical (rather than the normative) question is likely dispositive.

197. Diamond v. Diehr, 450 U.S. 175, 191-93 (1981) (emphasis added).

198. As a technical matter, *Mayo* held that the additional steps are necessarily incidental to implementing the underlying principle. Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1298 (2012).

199. *Id.* at 1303.

200. Although not explicit, this normative calculus is suggested by the reasoning in *Diehr*. *See, e.g.*, *Diehr*, 450 U.S. at 178 ("[T]he industry has not been able to obtain uniformly accurate cures because the temperature of the molding press could not be precisely measured, thus making it difficult to do the necessary computations to determine cure time. Because the temperature inside the press has heretofore been viewed as an uncontrollable variable, the conventional industry practice has been to calculate the cure time as the shortest time in which all parts of the product will definitely be cured, assuming a reasonable amount of mold-opening time during loading and unloading. But the shortcoming of this practice is that operating with an uncontrollable variable inevitably led in some instances to overestimating the mold-opening time and overcuring the rubber, and in other instances to underestimating that time and undercuring the product.").

201. *Mayo* reasoned that the claims "threaten to inhibit the development of more refined treatment recommendations (like that embodied in *Mayo's* test), that combine Prometheus' correlations with later discovered features of metabolites, human physiology or individual patient characteristics…including later discovered processes that measure metabolite levels in new ways . . . these patents tie up too much future use of laws of nature . . . ." *Mayo*, 132 S. Ct. at 1302.

This article argues that *Alice* should have been decided according to the third line of analysis, as discussed above. This article advances technical arguments that the recited limitations should be understood, at the appropriate level of abstraction, as a meaningful technological solution. Given the normative requirements for eligibility (e.g., improvement in technology or technical field), as articulated by *Alice*, the claims at issue should be eligible. This article critiques *Alice* as incorrectly unfolding along the first line of analysis—the normative determination is a forgone conclusion based on mischaracterizations of the recited limitations as a technical matter. These two lines of analysis diverge at the critical first step of *Mayo's* framework, a step which is clarified below as the determination of the appropriate level of abstraction for claim construction.

IV. CRITIQUE OF *ALICE CORP. V. CLS BANK INTL.*

A. Mayo's *Framework, Step 1: Level of Abstraction*

Alice observed that "[a]t some level [of abstraction], 'all inventions . . . embody, use, reflect, rest upon, or apply laws of nature, natural phenomena, or abstract ideas.'"[202] Therefore, *Alice* cautioned that "we [must] tread carefully in construing this exclusionary principle [excluding abstract ideas] lest it swallow all of patent law."[203] However, *Alice's* syllogism was neither careful nor correct. *Alice's* logical fallacy can be summarized as follows:

(1) Construed at a high level of abstraction, the claims at issue in *Bilski* are directed to the basic concept of hedging.[204]

(2) The concept of hedging is a fundamental economic practice.[205]

(3) The underlying concept of intermediated settlement in the claims at issue in *Alice* is also a fundamental economic practice.[206]

202. Alice Corp. Pty. v. CLS Bank Int'l, 134 S. Ct. 2347, 2354 (2014) (citation omitted).

203. *Id.*

204. "[T]he patent at issue in Bilski claimed an 'abstract idea'…the claims described 'the basic concept of hedging, or protecting against risk.'" *Id.* at 8 (citations omitted).

205. "The Court explained that '[h]edging is a fundamental economic practice long prevalent in our system of commerce and taught in any introductory finance class.'" *Id.* (citations omitted).

206. "Like the risk hedging in *Bilski*, the concept of intermediated settlement is 'a fundamental economic practice long prevalent in our system of commerce.'" *Id.* at 9 (citations omitted).

(4) Therefore, the claims at issue in *Alice* should be construed at a high level of abstraction.[207]

Alice erred in assuming that because, at some level, the claims at issue in *Bilski* and *Alice* reflect fundamental economic practices, the claims should be construed at the *same level* of abstraction.[208]

Contrast the Supreme Court's analysis with that of the Federal Circuit. The majority in the Federal Circuit conceded the undeniable conclusion that the system claims in *Alice* should be construed at a lower (albeit "incrementally reduced") level of abstraction than the method claims.[209] Similarly, the concurrence-in-part and dissent-in-part (Rader, Linn, Moore, O'Malley) (hereinafter "Dissent") in the Federal Circuit construed the system claims at a lower level of abstraction.[210] The Dissent reasoned that "[t]he key to this inquiry is whether the claims tie the otherwise abstract idea to a specific way of doing something with a computer, or a specific computer for doing something; if so, they likely will be patent eligible . . ."[211] The invention, the dissent observed, is a specific way of doing something with a computer: the computers are specifically (and architecturally) configured for distributed multiprocessing.[212] Therefore, because the

207. "Thus, intermediated settlement, like hedging, is an 'abstract idea' beyond the scope of § 101 . . . we need not labor to delimit the precise contours of the 'abstract ideas' category in this case. It is enough to recognize that there is no meaningful distinction between the concept of risk hedging in *Bilski* and the concept of intermediated settlement at issue here." *Id.* at 9–10.

208. The flawed reasoning in *Alice* is similar to the reasoning in *Bilski*. In his concurrence in *Bilski*, Justice Stevens was precisely on point in his criticism that the Court "discounts the application's discussion of what sorts of data to use, and how to analyze those data, as mere 'token postsolution components'. . . [and] *artificially limits petitioners' claims to hedging*, and then concludes that hedging is an abstract idea rather than a term that describes a category of processes including petitioners' claims. Why the Court does this is never made clear." Bilski v. Kappos, 561 U.S. 593, 620 (emphasis added). *Mayo* cautioned against a similar problem. *See* Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1304 (2012) (". . . studiously ignoring all laws of nature when evaluating a patent application under §§ 102 and 103 would make all inventions unpatentable because all inventions can be reduced to underlying principles of nature which, once known, make their implementation obvious.") (citation omitted).

209. CLS Bank Int'l v. Alice Corp. Pty. Ltd., 717 F.3d 1269, 1290 (Fed. Cir. 2013), *cert. granted*, 134 S. Ct. 734 (U.S. 2013).

210. *Id.* at 1310 (". . . these limitations are not stated at a high level of generality."). Note that Chief Judge Rader and Judge Moore affirmed the district court's conclusion that the method and media claims are patent ineligible. *Id.* at 1313.

211. *Id.* at 1302.

212. *Id.* at 1307 ("[the specification] states, for example, that the 'core of the system hardware is a collection of data processing units.' '375 Patent col. 7 ll. 22–23. Each processing unit 'is operably connected with . . . one or more mass data storage units . . . to store all data received from stakeholders, and other data relating to all other software operations generating or retrieving stored information.' *Id.* col. 7 ll. 39–43. The specification also explains that the communications controllers 'effect communications between the processing units . . . and the

recited limitations cover "the use of a computer and other hardware specifically programmed to solve a complex problem"[213] and the steps are not inherent to the abstract idea of an escrow,[214] the system claims should be construed at a lower level of abstraction. Otherwise, the dissent observed, "[l]abeling this system claim an 'abstract concept' wrenches all meaning from those words."[215]

In *Alice*, the determination of the appropriate level of abstraction for claim construction is pivotal. To appreciate why this is so, consider first the system claims at issue in *Alice*.

As a technical matter, the Court's characterization of the system claims is erroneous. It is simply not true that "[n]early every computer" will have the specific hardware of the claimed invention.[216] The system claims recite hardware that is far from generic. The system claims are directed to distributed multiprocessing architectures comprising of multiple interconnected computers. For example, the system claims in US 5,970,479 and US 7,725,375 recite a decentralized architecture.[217] The system claims in US 7,725,375 recite a centralized architecture for which communication is coordinated by a specialized controller.[218] Solving the particular *technological problem* of providing sufficient computational capability, the distributed multiprocessing architecture

---

various external hardware devices used by the stakeholders to communicate data or instructions to or from the processing units.' *Id.* col. 7 ll. 46–52. The computer can connect to the communications controller by means of another machine, a modem. *Id.* col. 7 ll. 57–60.").

213. *Id.* at 1307.

214. *Id.* at 1309 ("[S]omeone can use an escrow arrangement in many other applications, without computer systems, and even with computers but in other ways without infringing the claims.").

215. *Id.* at 1290.

216. Alice Corp. Pty. v. CLS Bank Int'l, 134 S. Ct. 2347, 2360 (2014).

217. For example, independent claim 16 of US 5,970,479 recites "a plurality of main data processing devices interconnected by at least one data communications link, each said data processing device running an operating system and applications software," "one or more data storage devices to which each data processing device has access," "a plurality of data input/output channels providing connection to a plurality of stakeholder locations," "each said location having data processing means," and "stakeholders inputting to a said data storage device." US 5,970,479 at 62:20–34. Independent claim 1 of US 7,725,375 recites a computer configured to "receive a transaction from said first party device." Dependent claim 12 of US 7,725,375 recites the limitation that the "computer is further configured to receive a transaction from said second party device." Dependent claim 13 of US 7,725,375 recites "said first party device and said second party device include a computer." US 7,725,375 at 65:15, 61–62, 65–67.

218. Independent claims 14 and 26 of US 7,725,375 recite a computer that is configured to "receive a transaction from said first party via said communications controller." US 7,725,375 at 66:13–14, 67:10–11. Dependent claims 25 and 37 of US 7,725,375 recite the limitation that the computer is further configured "to receive a transaction from said second party . . . via said communications controller." *Id.* at 66:61–62, 68:1–2. Claim 38 of US 7,725,375 recites "said first party device and second party device include a computer." *Id.* at 68:4.

is "a large system designed to handle the transactions of thousands of stakeholders, the input and output data generated by those stakeholders, and risk management contact pricing, matching and subsequent processing functions."[219] As a technological solution, the distributed multiprocessing architecture has specific technical advantages such as scalability,[220] configurability,[221] and adaptability.[222] Such technical advantages of distributed multiprocessing architectures have also enabled other innovations from climate-change modeling[223] to distributed robotics.[224]

---

219. US 7,725,375 at 7:34–38. US 5,970,479 at 7:45–49. Admittedly, the role of the specification in claim construction is a matter of controversy. While Professor Chiang has insightfully characterized the claim construction debate as policy disagreements about the appropriate level of abstraction, Chiang may likely be overstating his argument by diminishing the problem of linguistic vagueness. *See* Chiang, *supra* note 78, at 1130 ("Rather than linguistic vagueness, the disagreement between the two claim construction camps [textualism and contextualism] is over the permitted level of abstraction. Interpreting claims [contextually] in light of the specification—that is, importing limitations from the specification—has the effect of reducing the abstractness of a claim and its corresponding scope."). *But see* Andrew Chin, *The Ontological Function of the Patent Document*, 74 U. PITT. L. REV. 263, 312–13 (2012–2013) (arguing that "the written description requirement serves to ensure that the claims are kinds that pick out well-defined classes, as is necessary to satisfy the patent system's criteria of ontological commitment . . . [and] to ensure that one who reads the applicant's claims in light of the specification thereby incurs *de dicto* ontological commitments to those claims according to the patent system's criteria for such commitments.").

220. "The 'virtual' level of the system 10 is termed INVENTCO. INVENTCO is a collection of one or more potentially interrelated systems, as shown in FIG. 3. The hardware configuration shown in FIG. 2, is to be understood both as a realisation for a single INVENTCO system, and equally can represent a number of INVENTCO SYSTEMS, where the processing unit 20 is common to all and supports a number of communications co-ordination and security units 25, others of which are not shown, together with associated external communications devices 70, also not shown . . . . Where there are a number of INVENTCO systems, those systems may be inter-dependent or stand-alone in nature. If inter-dependent, INVENTCO (10) is responsible for transactions between those systems." US 7,725,375 at 8:34–56. US 5,970,479 at 8:44:65

221. "The generic system 10 shown in FIG. 1 encompasses many varied configurations, relating not only to the number and types of stakeholders, but also the 'architectures' realisable by the system hardware and software in combination. In that sense the arrangement shown in FIG. 2 is to be considered only as broadly indicative of one type of hardware configuration that may be required to put the invention into effect." US 7,725,375 at 8:23–29. US 5,970,479 at 8:35–43.

222. "INVENTCO and all of its component parts can be legally or geographically domiciled in separate countries or states. The supra-national nature of INVENTCO enables the stakeholders to avail themselves of the risk management mechanisms independently of legal domicile or other such restrictions that are often a feature of some conventional risk management mechanisms, subject to meeting certain criteria regarding credit worthiness and such. Indeed, the legal domicile, location, ownership and participating stakeholders of INVENTCO, or any of the sub-systems, can be continually changing." US 7,725,375 at 8:57–67. US 5,970,479 at 8:66–9:9.

223. For example, Climateprediction.net (CPDN) is a distributed computing project developed by Oxford University. *See* CLIMATEPREDICTION.NET (Jan. 13, 2015), http://www.climateprediction.net.

224. For example, the Distributed Robotics Laboratory at MIT is developing systems of self-organizing robots based on distributed algorithms. *See* DRL WIKI, http://groups.csail.mit.edu/drl

Alice misunderstood each claim as a whole because *Alice* misconstrued the recited limitations. For the system claims, *Alice* mischaracterized the recited hardware as generic.[225] Consequently, *Alice* construed the system claims at a high level of abstraction as simply applying the abstract idea of intermediated settlement.[226] Similarly, *Alice* also mischaracterized the recited method steps as generic.[227] *Alice* construed the method claims at a high level of abstraction as simply applying the same abstract idea.[228]

In *Alice*, construing the claims at different levels of abstraction yields different results. Construed at a sufficiently high level of abstraction, the claims are directed to ineligible subject matter because the distributed multiprocessing architecture is tangential to the underlying concept of intermediated settlement. Intermediated settlement is a fundamental economic practice that does not require computers. Evaluated at a lower level of abstraction, the distributed multiprocessing architecture is more easily appreciated as a technological solution that solves a technological problem. Therefore, evaluating the claimed invention at the appropriate level of abstraction is pivotal.

Accordingly, the first step of *Mayo's* framework should be clarified as the determination of the appropriate level of abstraction for claim construction.[229] Contrary to *Alice*, the first step is *not* a determination of whether the claims at issue are directed to an abstract

/wiki/index.php?title=Main_Page (last modified May 22, 2014, 3:29 PM).

225. "But what petitioner characterizes as specific hardware . . . is purely functional and generic." Alice Corp. Pty. v. CLS Bank Int'l, 134 S. Ct. 2347, 2360 (2014).

226. "[T]he system claims recite a handful of generic computer components configured to implement the same idea." *Id.*

227. "Taking the claim elements separately, the function performed by the computer at each step of the process is "[p]urely conventional. . . . In short, each step does no more than require a generic computer to perform generic computer functions." *Id.* at 15.

228. "Instead, the claims at issue amount to 'nothing significantly more' than an instruction to apply the abstract idea of intermediated settlement using some unspecified, generic computer." *Id.*

229. Professor Chiang compellingly argues that the "task of defining patent scope—translating an embodiment into a protected idea—is a classic levels of abstraction problem. The choice between levels of abstraction as a means of expressing some idea or principle presents problems of arbitrariness, a difficulty well known in many areas of law. The problem is worse in patent law, however, because courts have not even acknowledged the nature of the problem. Instead, courts routinely treat the idea and embodiment in a patent as the same thing so that the correct level of abstraction to express the idea is made to seem self-evident. This only makes the task of ascertaining patent scope more difficult by obscuring the process by which one level of abstraction is chosen from the many available." *See* Chiang, *supra* note 78, at 1100.

idea.[230] At some level, all inventions embody abstract ideas.[231] Further, ideas are found at different levels of abstraction, as *Altai* observed.[232] Implicit in this reformulation of the first step of *Mayo's* framework is a workable definition of an *abstract idea*. To be sure, some *abstract ideas* are undeniably fundamental building blocks of human ingenuity, such as the concept of an escrow or hedging. This article argues that less fundamental *abstract ideas* at lower levels of abstraction (that may also be building blocks) can be fruitfully characterized as abstractions—that is, the products of the creative act of abstraction.[233]

Admittedly, copyright law has recognized the *ad hoc* nature of a balancing of interests in the level-of-abstraction determination. There is no bright-line rule on determining the appropriate level of abstraction and, concomitantly, the proper scope of copyright protection. Learned Hand maintained that, in the determination of the boundary between protectable and non-protectable subject matter, "[n]obody has ever been able to fix that boundary, and nobody ever can."[234] In formulating a nearly identical three-part test that prefigured the abstraction-filtration-comparison test advanced by *Altai*, the court in *Lotus Dev. Corp. v. Paperback Software Int'l* observed that "no principle can be stated as to when an imitator has gone beyond copying the 'idea,' and has borrowed its 'expression.' Decisions must therefore inevitably be ad hoc. In another context, Hand described such 'ad hoc' decision making as 'fiat.'"[235]

Even so, this article argues that the level-of-abstraction analysis of useful artifacts in patent law (as compared to purely textual, literary works in copyright law) may admit of a more predictable approach or principle.[236] This article submits that the principled approach should

230. "First, we determine whether the claims at issue are directed to one of those patent-ineligible concepts." Alice Corp. Pty. v. CLS Bank Int'l, 134 S. Ct. 2347, 2355 (2014).

231. "At *some* level [of abstraction], 'all inventions . . . embody, use, reflect, rest upon, or apply . . . abstract ideas.'" *Id.* at 2354.

232. *See supra* notes 94–96 and accompanying text.

233. *See supra* notes 148–154 and accompanying text.

234. Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930).

235. 740 F. Supp. 37, 60 (D. Mass. 1990) (internal quotation marks omitted).

236. Although ultimately incomplete, as discussed below, Professor Morris's notion of "action" and Professor Chin's notion of "causal power" suggest that the hermeneutics, as it were, of the meaning of useful artifacts (and claims thereof) is more constrained than compared to the explication of the meaning of literary works. *See infra* notes 238–50 and accompanying text; *see also* Knuth, *Structured Programing*, *supra* note 169, at 294. *But see* Chiang *supra* note 78, at 1123 ("Within the spectrum of abstractions, each description is as equally accurate as any other. As a matter of formalist principle, there is almost no limit on how far one can move up or down the abstractions ladder. Between claiming every last detail of the specification embodiment and omitting all details and claiming a fundamental principle, there is no legally principled limit. The

focus on or emphasize the role of human agency rather than an essential property of the artifact itself—designed artifacts are useful precisely because of the human agency and purpose that inheres in the artifact.[237] As an example of the former approach emphasizing the role of human agency, this article argues that, by carefully inquiring into the creative process of writing software, courts can uncover design choices which suggest an appropriate level of abstraction. Exemplifying the latter approach is Professor Chin's article, *The Ontological Function of the Patent Document*, which, on my reading, represents the one of the most well-considered theoretical approaches to the subject-matter inquiry.[238] However, Chin may be pursuing a theoretically more difficult direction by positing the existence of an "essential causal power" that serves as the basis for an "essential causation requirement." The requirement,

criteria of novelty, usefulness, and nonobviousness provide no definitive answer. Nor do enablement and written description provide an answer: these two related doctrines are both internally contradictory and thus provide no coherent principle at all. Nonetheless, courts must and do draw lines on the permissible level of abstraction. The lines, however, are drawn silently, without informing the public and litigants of any coherent principle.") (citations omitted).

237. Rather than implying a kind of vitalism, this is simply to say that artifacts cannot be understood without understanding its purpose and design.

238. Generally, in searching for textual "clues" in case law, commentators have missed the opportunity to think about deeper, cohering principles. Consequently, such "clues" yield a disparate collection of rules that remain conceptually problematic. *See, e.g.*, Richard H. Stern, *Mayo v. Prometheus: No Patents on Conventional Implementations of Natural Principles and Fundamental Truths*, 34 EUROPEAN INTELL. PROP. REV. 502, 514 (2012) (arguing that "case law gives us ample and powerful clues [for a positive test of patent-eligibility] . . . . The implementation of the principle must add to the principle more than just the conventional expedients that scientists or artisans in the relevant field know and use, or the implementation should at least purport to do so. . . . A novel apparatus helps. Novel co-operation or co-action is very helpful. . . . The claims should not be open-ended and exceed what the inventor has taught the public to do to exploit the principle . . . . That is how *Prometheus* teaches us to recognise when a claim 'amounts to significantly more than a patent upon the natural law itself.'"). *But see, e.g.*, Edmund W. Kitch, *The Nature and Function of the Patent System*, 20 J. L. & ECON 265, 268 (1977) (arguing that the "[o]ne reason the prospect function of the patent system may have been so long overlooked is that the 'hornbook' rule is very misleading—the inventor may not claim more than he has invented, and the claim marks the outer bounds of his rights . . . . But the rule is misleading, because the invention as claimed in the patent claims and the physical embodiment of the invention are two quite different things."); Chiang, *supra note* 78, at 1152 (observing that "some degree of control beyond the precise embodiment is necessary for patent incentives to work. If the patent's scope is confined to precise replication of the first working embodiment, then pirates would quickly learn to copy the principle or the heart of the patent without replicating the precise embodiment."); Chin, *supra* note 219, at 273–74 (observing that "[a] widespread misconception about patent claims is that they are merely sets of embodiments, so that certain doctrines about claim scope are reducible to set-theoretic propositions . . . it is an imprecise and inadequate ontological description because while the definition of a set necessarily determines a patent claim's elements, the language of a claim does not determine which, if any, of its embodiments exist. Conversely, the number of existing patent claim embodiments has no effect on the claim's scope. All empty sets are identical, yet there are many distinct patent claims with no existing embodiments.") (citations omitted).

according to Chin, provides "a precise criterion for distinguishing between a patent-ineligible abstract idea and a patent-eligible practical method or means of producing a beneficial result or effect."[239]

The problem with Chin's approach is that his narrow conception of the causal power of artifacts is devoid of human agency. Chin overlooks the role of human agency which, as this article has argued, is determinative of the appropriate level of abstraction. Chin submits that "[g]enerally accepted explanatory [causal] principles governing the involvement of [real-world] resources in the essential causal powers of the claim's embodiments may range from the conservation laws of physics to the scheduling disciplines implemented in operating systems."[240] However, by reifying the implementation of scheduling disciplines as if it were a causal law of nature, Chin overlooks the design choices in the creative process. Those design choices, as this article has shown, are not only contingent but are also determinative of how the scheduling disciplines are integrated into the software architecture as a whole.[241] Chin suggests that any software-implemented invention that achieves a practical and useful effect is amenable to explanation by a single causal account in terms of real-world resources such as "CPU cycles, network bandwidth, memory, disk space, and battery life."[242] However, no causal account is complete without an explanation of how such real-world resources are mediated by human agency that inheres in software.[243] In sum, Chin avoids the

239. Chin, *supra* note 219, at 324.

240. *Id.*

241. *See supra* notes 164–67 and accompanying text. *See also* Feenberg, *supra* note 23, at 12 (observing that "in years to come the technical experts will forget the politics behind their reformed designs and when new demands appear will defend them as a product of pure and objective knowledge of nature!").

242. Chin, *supra* note 219, at 324.

243. Rather than the philosophical position of *scientific realism*, which Chin argues is an epistemological commitment by patent law to a *mind-independent* world, perhaps the notion of *irrealism* (as distinct from *anti-realism*) may be more fruitful in the analysis of human agency that inheres in software. *See* Chin, *supra* note 219, at 295–99 (submitting that the patent system is committed to scientific realism). *But see* Dick Hamlet, *Science, Mathematics, Computer Science, Software Engineering*, 55 COMPUT. J. 99, 103 (2012) ("[T]here is reason to doubt there will ever be a science of software. Programs and programming have a made-up, abstract nature that is disconnected from external constraints. There can be no software natural laws because the whole thing is an arbitrary human invention. No statement about software could be falsified by an experiment, because there is no 'reality' for software that decides truth.") (footnote omitted). The notion of *irrealism* has been explored by scholars, albeit not in the context of software. For example, in distinguishing his conception of the *irrealism* from that of Nelson Goodman who first introduced the philosophical position, sociologist Nikolas Rose writes, "[m]y own irrealism is technical, not psyhologistic. It is technical in so far as it asserts that thought constructs its irreal worlds through very material procedures. *Thought, that is to say, becomes real by harnessing*

level-of-abstraction problem and the question of human agency by focusing solely on the external aspects of an artifact.[244] The determination of the appropriate level of abstraction, a critical step in the eligibility inquiry, should instead conceive of artifacts as animated by human agency.

In comparison, artifice and action are two persistent themes that Professor Morris perceives in disparate subject matter upheld as patent-eligible.[245] Artifice is "roughly defined as perceived degree of alteration through human intervention." and action is "roughly defined as new operation or activity through human intervention."[246] According to Morris, artifice and action can serve as criteria for distinguishing between patent-ineligible and patent-eligible subject matter. Morris argues that "[e]xplicitly adopting artifice and action as the standard for patentable technology offers a number of benefits. The artifice-plus-action standard allows patent law to move away from wrangling over what an 'abstract idea' or 'process' is or from deciding whether an invention is truly a machine or just cleverly claimed as such."[247] Morris is correct in submitting that artifice and action are matters of degree: "All patentable subject matter displays some threshold degree of artifice [and] . . . at least a threshold degree of action."[248] However, Morris overlooks the possibility that different artifice and different actions can co-exist (or not) at different levels of abstraction, as this article has shown.[249] Therefore, while Morris recognizes the important

---

itself to a practice of inscription, calculation, and action." NIKOLAS ROSE, POWERS OF FREEDOM, REFRAMING POLITICAL THOUGHT 32 (1999) (emphasis added); *see also* NEW MATERIALISMS, ONTOLOGY, AGENCY, AND POLITICS 6–7 (Diana H. Coole & Samantha Frost eds., 2010) (observing that "we are finding our environment materially and conceptually reconstituted in ways that pose profound and unprecedented normative questions" that call for a 'new materialist' critique of modernity's presumptions about agency and causation).

244. Kroes et al. submit a relevant distinction between "thin" and "thick" conceptions of function, wherein the latter conception recognizes that "at some level in some way functions are connected with intentions." Kroes et al., *supra* note 27, at 113 (Any physical system will exhibit a reliable association between inputs and outputs … and thus can be ascribed a thin function. But how does that specific function relate to how we want that system to behave. As long as any reliable association between inputs and outputs can pass for a (thin) function of the object, it is not clear how such a thin function can establish a connection between the actual behavior of a system and its intended behavior . . . . There are no relevance criteria and therefore an infinity of functions can be associated with an artifact. Again, the interpretation of outputs as ends does not solve this problem. In the first place, since any kind of output becomes an end . . . .").

245. Emily Michiko Morris, *What is "Technology,"* 20 B.U. J. SCI. & TECH. L. 24, 65 (2014).

246. *Id.* at 25.

247. *Id.*

248. *Id.*

249. *See, e.g.*, *supra* notes 155–177 and accompanying text.

role of human agency (by including "human intervention" in the definitions of artifice and action), like Chin, Morris overlooks the level-of-abstraction problem.[250] Ultimately, Morris's approach is inherently indeterminate.

Therefore, the first step of *Mayo's* framework should be clarified as the determination of the appropriate level of abstraction by focusing on human agency and by inquiring into the creative process. As such, the relative functional concern of the eligibility inquiry, as articulated by *Mayo*, is not a free-ranging inquiry. The dissent in the Federal Circuit warned that "[a] court cannot go hunting for abstractions by ignoring the concrete, palpable, tangible limitations of the invention the patentee actually claims."[251] Hunting for abstractions with its erroneous analogy to *Bilski*, the Court in *Alice* overlooked the technological significance of the limitations at issue. In *Alice*, if the Court had considered the claimed invention at a lower level of abstraction, perhaps the Court would have been more receptive to the design choice of a distributed multiprocessing architecture as a technological solution. At the appropriate level of abstraction, the significance of the claimed limitations and any improvement to a technology, technical field, or the functioning of the computer itself become more apparent.

B. Technical Aspect of Mayo's Framework, Step 2: The Search for the Inventive Concept

1. Method Claims

Alice observed that "[s]tating an abstract idea while adding the words 'apply it' is not enough for patent eligibility."[252] *Alice* held that "the claims at issue amount to 'nothing significantly more' than an instruction to apply the abstract idea of intermediated settlement."[253]

---

250. Admittedly, Morris does recognize another dimension in the determination of artifice and action which seems to suggest that claims should be construed at an appropriate level of abstraction. According to Morris, not only are the constituent elements of an invention characterized according to their level of artifice and action but also "their respective roles affects whether the invention as a whole displays adequate artifice and action." Morris, *supra* note 245, at 63. However, it remains entirely unclear how the qualities of artifice and action of the constituent elements translate to that of the invention as a whole. (According to Morris, artifice and action are scalar (i.e., one-dimensional) qualities. *Id.* at 35. As such, without another mediating concept, the invention as a whole can *never* be more (or less) than the sum of its parts.)

251. CLS Bank Int'l v. Alice Corp. Pty., 717 F.3d 1269, 1298 (Fed. Cir.), *cert. granted*, 134 S. Ct. 734 (2013), *aff'd*, 134 S. Ct. 2347 (2014).

252. Alice Corp. Pty. v. CLS Bank Int'l, 134 S. Ct. 2347, 2358 (2014) (citation omitted).

253. *Id.* at 15.

The "apply it" formulation is imprecise.[254] As discussed above, the formulation does not adequately capture the difference between the claims at issue in *Flook* and in *Diehr*.[255] Both the patent-eligible claim in *Diehr* and the patent-ineligible claim in *Flook* simply apply the underlying formulas.[256] This article argues that *Flook* and in *Diehr* can be harmonized by construing their respective claims at the appropriate level of abstraction.[257] Similarly, in *Alice*, the "apply it" formulation obscures the proper analysis of the claims at issue. The proper analysis involves construing the claims at the appropriate level of abstraction.

Consider the "shadow record" limitation at a lower level of abstraction vis-à-vis the fundamental economic idea of intermediated settlement.[258] Like the macros and parameter lists in *Altai*, the shadow record in *Alice* is an external aspect of a hidden architecture. As a factual matter, the Court's characterization of the shadow record as amounting to "'nothing significantly more' than an instruction to apply the abstract idea" is incorrect.[259]

The shadow record is a *proxy*—a term of art that describes a software design pattern. The "Gang of Four," in their seminal book, *Design Patterns: Elements of Reusable Object-Oriented Software*, described design patterns as "simple and elegant solutions to specific problems in . . . . software design."[260] As a *technological solution*, the proxy pattern introduces an intermediate abstraction (e.g., a proxy or shadow) to achieve specific technical advantages. The "Gang of Four" described the *proxy pattern* as an optimizable means to "[p]rovide a surrogate or placeholder for another object to control access to

254. *Id.* at 11 ("*Mayo* made clear that transformation into a patent-eligible application requires 'more than simply stat[ing] the [abstract idea] while adding the words 'apply it.'"); *Id.* at 13 ("Stating an abstract idea while adding the words 'apply it' is not enough for patent eligibility.") (citation omitted).

255. *See supra* notes 69–75 and accompanying text.

256. *See supra* notes 69–75 and accompanying text.

257. *See supra* notes 187–190 and accompanying text.

258. "The representative method claim in this case recites the following steps: (1) 'creating' shadow records for each counter party to a transaction; (2) "obtaining" start-of-day balances based on the parties' real-world accounts at exchange institutions; (3) 'adjusting' the shadow records as transactions are entered, allowing only those transactions for which the parties have sufficient resources; and (4) issuing irrevocable end-of-day instructions to the exchange institutions to carry out the permitted transactions." Alice Corp. Pty. v. CLS Bank Int'l, 134 S. Ct. 2347, 2356 (2014).

259. *Id.* at 15. Commentators have generally accepted the Court's erroneous characterization at face value.

260. GoF, *supra* note 107, at xi (advising readers that the implementation of design patterns "might take a little more work than *ad hoc* solutions. But the extra effort invariably pays dividends in increased flexibility and reusability. Once you understand the design patterns and have had an 'Aha!' (and not just a 'Huh?') experience with them, you won't ever thinking about … [software] design in the same way.").

it . . . . The *proxy pattern* introduces a level of indirection when accessing an object . . . allow[ing] additional housekeeping tasks [e.g., checking for overdrafts] when an object is accessed."[261] For example, a real-time computer system may utilize a proxy or shadow to solve timing or "concurrency" problems that arise when multiple tasks require access to a shared resource.[262] In contrast to the adapter, as discussed above, which provides a different interface to the object it adapts, a proxy provides the same interface as the object it mediates.[263]

For the real-time settlement system at issue in *Alice*, the proxy or shadow record encodes a specific settlement mechanism (e.g., checking for overdrafts) and account structure.[264] For example, the prior art of record describes an alternative settlement mechanism that utilizes a queuing mechanism whereby uncovered payment orders can be held pending.[265] The prior art also contemplates a third alternative settlement mechanism utilizing "daylight" overdraft credits that allow for short-lived overdrafts.[266] In contrast, the claims at issue in *Alice* recite the limitation:

> [F]or every transaction resulting in an exchange obligation, adjusting the respective shadow credit record or shadow debit record, *allowing only those transactions* that do not result in the value of the shadow debit record being less than the value of the shadow credit record at any time, each said adjustment taking place in *chronological order* . . . .[267]

According to the prior art of record, "[t]he debate over what constitutes an optimal settlement arrangement, however, has only recently been started. So far, it has not led to conclusive results . . . In reality, the

261. *Id.* at 207–10.

262. *See, e.g.*, Dimitrios I. Kosmopoulos, *A Design Framework for Sensor Integration in Robotic Applications*, *in* INDUSTRIAL ROBOTICS 1, 11 (Kin-Huat Low ed., 2007) ("A good *trade-off* between flexibility and usability is provided if we build a library of components. . . . The implementation of the *design decisions* at the control object layer . . . . The implementation of the design framework uses the services of a communication layer … which has to be deterministic to cover the needs of a real-time industrial system. . . . A method for communication of modules that reside in different processes is the '*proxy' pattern*.") (emphases added).

263. GoF, *supra* note 107, at 216 ("However, a proxy used for access protection might refuse to perform an operation that the subject will perform so its interface may be effectively a subset of the subject's.").

264. *See, e.g.*, U.S. Patent No. 7,725,375 col.29 l.2–29; U.S. Patent No. 5,970,479 col.24 l.66–25:24; *see also* Christian Vital, *The Architecture of Real Time Gross Settlement* Systems, *in* GLOBAL PAYMENT SYSTEMS 23, 28, 31–32 (1996). *See, e.g.*, Office Action Appendix, Application No. 11/166387 (Mar. 30, 2010).

265. Vital, *supra note* 264, at 28.

266. *Id.*

267. *See, e.g.*, U.S. Patent No. 5,970,479 claim 12 (emphases added).

issues are complex. The comparison of benefits and costs is a difficult task . . . the question is likely to remain controversial for some time to come."[268]

The utilization of the proxy or shadow record in the context of real-time settlement systems produces new complications.[269] For example, institutions that initiated rejected transactions will have to resubmit the transactions at a later time and, for large payment volumes, this solution presupposes the existence of automated queuing mechanisms within the participating institutions.[270] Thus, the shadow record (like the design choices in *Altai*, as discussed above) encodes a particular tradeoff of benefits and costs in the complex endeavor of designing real-time settlement systems.

Therefore, contrary to *Alice*, the method claims do not simply implement the abstract idea of intermediated settlement on "some unspecified, generic computer."[271] As a factual matter, the Court's characterization of the shadow record as amounting to "electronic recordkeeping—one of the most basic functions of a computer" is incorrect. Probed more deeply, at the appropriate level of abstraction, the shadow record is revealed to be the external aspect of a more complex architecture. The Court's reductive mischaracterization is akin to describing a towering architecture only in terms of the pressure it exerts on the ground.

2. System and Media Claims

The Court's two-paragraph analysis of the system and media claims is double-pronged.[272] The first and second paragraphs are analytically distinct. The second paragraph is not merely an alternative and equivalent re-articulation, despite the Court's language (i.e., "Put another way"). The first paragraph focuses on hardware limitations of the system/media claims. By contrast, the second paragraph focuses on a comparison between the method and system/media claims. The first paragraph considers the system/media claims on their own terms. By contrast, the second paragraph construes the system/media claims in

---

268. *Id.* at 23.

269. Professor Feenberg observes that "the process of decontextualizing and recontextualizing technical objects sometimes results in unexpected problems . . . in simplifying, technological projects . . . produce new complications. This is why context matters." *See* Feenberg, *supra* note 23, at 10–11 (submitting "*a seventh paradox of* complexity which can be succinctly stated as: *Simplification complicates*.").

270. Vital, *supra* note 264, at 28.

271. Alice Corp. Pty. v. CLS Bank Int'l, 134 S. Ct. 2347, 2360.

272. *Id.*

reference to the method claims, based on the same level of abstraction.[273]

The analysis articulated in the first paragraph involves the more difficult technical and normative questions of whether the recited hardware provides meaningful limitations. Factually, the Court's conclusions are incorrect. As discussed above, construed at a lower level of abstraction, the limitations are directed to distributed multiprocessing architectures. As such, the limitations are meaningful because they are directed to technological solutions that solve technological problems.

By contrast, the analysis in the second paragraph leads to a foregone conclusion because the system/media claims are construed in reference to the ineligible method claims. The Court held that the "computer components [are] configured to implement the same [abstract] idea [of intermediated settlement]."[274] The Court construed the hardware limitations in the system/media claims at the same high level of abstraction. At this height of abstraction, the architectural details are easily overlooked as generic and tangential. Consequently, the Court invoked the warning against making patent eligibility "depend simply on the draftsman's art."[275]

However, the problem with the "draftsman's art" formulation is that it is meaningless. All claims are drafting efforts attempting to monopolize (to varying degrees) the underlying ideas. Yet courts continue to rely on the formulation as if it provides a meaningful distinction.[276] Just as the "apply it" formulation is imprecise,[277] so the "draftsman's art" characterization is a problematic formulation that does not adequately capture the technical significance (or insignificance) of the claimed limitations at issue. Courts can more meaningfully characterize claimed limitations in terms of the degree of technical necessity (based on an appropriate level of abstraction).[278] Technically descriptive characterizations of claimed limitations are critical because, at bottom, normative prescriptions depend on accurate descriptions.

---

273. ". . . the system claims recited . . . to implement the same idea [as the method claims]." *Id.*

274. *Id.*

275. *Id.*

276. Judicial language does not provide guidance for distinguishing between claims that "depend simply" on the draftsman's art and claims that "depend not so simply" on the draftsman's art.

277. *See supra* notes 254–257 and accompanying text.

278. *See supra* notes 187–196 and accompanying text.

C. *Normative Aspect of* Mayo's *Framework, Step 2: The Changing Nature of the Machine*

The normative concern finds expression in various modes. One mode of expression is speculative.[279] Speculation is inevitable because the eligibility inquiry, as articulated by *Mayo*, is a relative functional concern about the creative value of a discovery and the future risks of pre-emption.[280] The normative concern also finds expression in a propositional mode. For example, *Alice* held that patent law should promote improvements in a technology, a technical field, or the functioning of the computer itself.[281] While the speculative mode (like all forecasting efforts) is inherently uncertain, the propositional mode admits of problems that can be remedied.

This part argues that *Alice*, exemplifying the latter mode of normative expression, is problematic because of (1) an erroneous understanding of a particular machine (e.g., computers and the software that programs them) and (2) misconceptions about technology and the nature of modern machines. Each will be discussed in turn below. Informed by a more descriptively accurate conception of technology, this part will propose a more fruitful approach to the normative inquiry.

Alice's normative criterion regarding "improvements in the functioning of the computer itself" is problematic because the formulation is based on a misconception of computer technology. As a descriptive phrase, "the functioning of the computer itself" is imprecise for at least two reasons. First, if the function of computers is inherently governed by software, then, contrary to patent doctrine, improvements in software *per se* necessarily improves the *functioning* of the computer itself.[282] Second, the pronoun *itself* seems to emphasize a distinction

---

279. *See, e.g.*, O'Reilly v. Morse, 56 U.S. 62, 113 ("For aught that we now know some future inventor, in the onward march of science, may discover a mode of writing or printing at a distance by means of the electric or galvanic current, without using any part of the process or combination set forth in the plaintiff's specification. His invention may be less complicated-less liable to get out of order-less expensive in construction, and in its operation."); Gottschalk v. Benson, 409 U.S. 63, 68 (1972) ("The end use may (1) vary from the operation of a train to verification of drivers' licenses to researching the law books for precedents and (2) be performed through any existing machinery or future-devised machinery or without any apparatus."); Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1302 (2012) ("[the claims] threaten to inhibit the development of more refined treatment recommendations (like that embodied in Mayo's test), that combine Prometheus' correlations with later discovered features of metabolites, human physiology or individual patient characteristics . . . including later discovered processes that measure metabolite levels in new ways . . . these patents tie up too much future use of laws of nature . . . .").

280. *Mayo*, 132 S. Ct. at 1303.

281. Alice Corp. Pty. v. CLS Bank Int'l, 134 S. Ct. 2347, 2359 (2014).

282. *See supra* note 26.

that is largely obsolete in the age of cloud computing. Computers are no longer self-contained machines. Processing is distributed and diffused through networks of interconnected machines. One example is the distributed multiprocessing architecture recited by the claims at issue in *Alice*.

Alice's erroneous and outdated understanding of computers is compounded with its misconceptions about *technology*. The juxtaposition, in *Alice*, between the terms *technology* and *technical field* recalls a history that is directly relevant to the focus of this article.[283] In its original, pre-modern usage, *technology* named a field of study.[284] Karl Marx was one of the first writers to use the term *technology* in its modern sense.[285] In its modern usage, *technology* means something much more abstract and elusive. Before turning to the significance of *Alice's* use of the term *technology*, this part will examine the conceptual and discursive changes revealed by the history of the term.

Technology is a critical term in patent law.[286] For instance, Professor Willoughby submits that "the whole edifice of patent law is arguably built on the conceptual foundation of something called 'technology.'"[287] Yet, the notion of technology remains largely intuitive and undefined rigorously.[288] Generally, commentators lament the elusiveness of the term *technology*[289] or unsuccessfully attempt a definition.[290] This article takes a different approach by examining the

283. *Alice Corp.*, 134 S. Ct. at 2359.

284. Leo Marx, *Technology, The Emergence of a Hazardous Concept*, 51 TECH. AND CULTURE 561, 575 (2010).

285. *Id.* at 570.

286. Arguably, patent law has used the word technology to mean something narrower and more closely related to the traditional concept of *machine*.

287. *See* Kelvin W. Willoughby, *How Much Does Technology Really Matter in Patent Law? A Comparative Analysis of Doctrines of Appropriate Patentable Subject Matter in American and European Patent Law*, 18 FED. CIR. B. J. 63, 137 (2009).

288. *Id.*

289. *See, e.g.*, *Interim Guidelines for Examination of Patent Applications for Patent Subject Matter Eligibility*, 1300 OFFICIAL GAZ. PAT. & TRADEMARK OFF. 1, 44 (Oct. 26, 2005), *available at* http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/guidelinesl01_20051026.pdf ("[A]ny attempts to define what is 'in the technological arts' raises more questions than it appears to answer.").

290. For example, Professor Willoughby proposed a definition of technology that is based on the problematic and multifarious notion of efficiency. *See* Willoughby, *supra* note 287, at 126 ("A technology is an artifact or system of artifacts, either tangible or intangible, which functions as a means towards the attainment of predetermined ends in a rational, *efficient* and causal manner.") (emphasis added). *But see* Feenberg, *supra* note 23, at 7 ("There is a corollary of the paradox of the origin. I call this fourth paradox, the *paradox of the frame* and formulate it as follows: *efficiency does not explain success, success explains efficiency* . . . there is no general

history of the term because, ultimately, it is the *changing nature of the machine* that has confused the patent law jurisprudence on subject-matter eligibility.[291] Thus, by revealing the changing nature of the machine, the history of *technology*—as machine and as word—can guide jurisprudence towards the right direction.[292]

Technology filled a semantic void. In nineteenth-century discourse about the increasingly-wide-ranging effects of machine power, *technology* filled a void left open by conventional terms such as *machine*, *mechanism*, and *mechanic arts*.[293] The nineteenth century witnessed a transformation in the character and complexity of machines into something recognizable today as technological systems. The railroad system, telegraph network, and power grids were complex configurations of material artifacts and human organizations.

rule under which paths of development can be explained. Explanation by efficiency is a little like explaining the presence of pictures in a museum by the fact that they all have frames. Of course all technologies must be more or less efficient, but that does not explain why they are present in our technical environment.").

291. Similarly, recalling Raymond Williams's famous study, in *Culture and Society* (1983), of the striking interdependence in the relations between certain keywords (e.g., *class*, *industry*, *democracy*) and fundamental changes in society and culture, MIT Professor Leo Marx submits that the emergence of *technology* as a keyword can be employed to chart the changing character of contemporary society and culture. Marx, *Technology, The Emergence of a Hazardous Concept*, *supra* note 284, at 563.

292. In writing about technology, commentators have generally assumed that (1) *technology* as machine has not fundamentally changed, and/or that (2) *technology* as a word does not have a discursive history which reflects that fundamental change. These assumptions often detract from the worthwhile thesis that patent law should understand technology as a means for broader improvements in society. *See, e.g.*, Arthur J. Cockfield & Jason Pridmore, *A Synthetic Theory of Law and Technology*, 8 MINN. J. L. SCI. & TECH. (2007) (proposing a general theory of law and technology that includes "substantive" theories of technology, as articulated by Karl Marx, Herbert Marcuse, and Jacques Ellul); Dana Remus Irwin, *Paradise Lost in the Patent Law? Changing Visions of Technology in the Subject Matter Inquiry*, 60 FLA. L. REV. 775, 794, 802, 813 (2008) (insightfully observing a shift in subject-matter jurisprudence from a discourse based on the nature/artifice distinction to a rhetoric of science and technology, a shift that the author incorrectly argues should be reversed, as "some categories [of eligible subject matter, including software-implemented inventions] seem entirely disconnected from the natural world."); John R. Thomas, *The Question Concerning Patent Law and Pioneer Inventions*, 10 BERKELEY TECH. L.J. 35, 81 (1995) (arguing that "the work of Martin Heidegger provides the philosophical basis for rejecting patent law's view of technology as a dominating, neutral, inevitable force."). The foregoing representative theses, searching for alternative philosophic bases for patent law, are reactions to an over-reliance by judges and scholars on narrow economic and utilitarian justifications. Whether these reactions yield specific and workable proposals for reform is an open question. However, these reactions are, I believe, part of the growing realization—within and outside the legal profession—of the vulnerabilities (and commensurate resourcefulness) of the human species. The project to reconceive the foundations of patent law is a pressing one, especially in an era in which technological development and its impacts are so far-reaching that the body of law solely focused on the promotion of technological innovation cannot afford to remain neutral.

293. Marx, *Technology, The Emergence of a Hazardous Concept*, *supra* note 284, at 563.

Celebrated and iconic technological innovations such as the steam engine were only the most visible aspect of wider transformations. For example, the locomotive was made possible not just by the network of railroads and the ancillary infrastructures for the production of parts and materials. Technological innovations also required modern institutions such as large corporations capable of harnessing labor and capital.[294] As discrete machines were transformed into more diffused and complex technological systems, the term *technology* served to describe a reality in which the artifactual and socio-economic components became inextricable.[295] The deep interpenetration of the machine and the human dimensions of technology was something entirely new. For Marx's contemporaries who witnessed firsthand this profound transformation, their vision had got ahead of their language.[296] Supplanting the outmoded lexicon of the *mechanic arts*, the word *technology* filled a semantic void.[297]

The development of the concept of technology is inseparable from its discursive history. Just as technology as a word describes something diffused and atmospheric, technology as a concept is rarified and abstract. While *mechanic arts* conjures the soiled hands in the workshop, *technology* seems purified of physicality.[298] While *industrial arts* is mired in the mundane practicality of work, *technology* seems elevated to the realm of higher learning and fine arts.[299] Technology is the work of highly-educated specialists, blurring the boundaries between the technical and the cultural.

The modern concept of technology assumed normative dimensions. In contrast to the Enlightenment notion of machine power as the means to a more just, more peaceful society, technology became an end in itself.[300] The modern technocratic notion of progress is based

---

294. Professor Leo Marx observed that "[i]t is noteworthy that the concept of technology gained currency during the 'incorporation of America.'" *Id.* at 575 ("[M]achines became working parts of a dynamic system, and the motives for change, the source of industrial dynamism, lay not in the inanimate machine but in the economic necessities perceived by its owners." (quoting ALAN TRACHTENBERG, THE INCORPORATION OF AMERICA 55 (1982))).

295. *Id.* (observing "the blurring of the boundary between the material (physical, or artifactual) components of these large socio-technological systems and the other, bureaucratic and ideological components. Even more significant, perhaps, is the erosion of the 'outer' boundaries, as it were, those separating the whole technological system from the surrounding society and culture.").

296. LEO MARX, THE MACHINE IN THE GARDEN, TECHNOLOGY AND THE PASTORAL IDEAL IN AMERICA 166 (2000).

297. Marx, *Technology, The Emergence of a Hazardous Concept*, *supra* note 284, at 574.

298. *Id.*

299. *Id.*

300. Leo Marx, *The Idea of "Technology" and Postmodern Pessimism*, *in* DOES

on the faith that once technical mastery is achieved, the rest will take care of itself.[301] Technology has become at once a self-justifying object (i.e., an end and a thing) and an autonomous causal agent of history.[302] As an object without a specifying adjective, technology invites endless reification.[303] As a disembodied signifier lacking a referent, technology is invested with metaphysical properties and potencies.[304]

Judges are susceptible to these interpretative or philosophic errors. Consider *Alice*. Rather than understood as a product of human creativity, the claimed technology at issue in *Alice* has become an object largely inaccessible to analysis. The significance of the *shadow record* is reduced to the basic function of a computer, to a mere aspect of a thing. *Alice* not only reflects the contemporary tendency to reify technology, but *Alice* also reflects the contemporary technocratic notion of progress. The unqualified normative criterion of improvement in technology is a self-justifying end in itself.

How can patent law reflect the changing nature of the machine? One approach is suggested by the schematic notion of second-order machines.[305] The notion has gained much purchase among technologists.[306] As Professor Allen explains, "[f]irst-order machines

---

TECHNOLOGY DRIVE HISTORY? THE DILEMMA OF TECHNOLOGIAL DETEMINISM 237, 250–51 (Merritt Roe Smit & Leo Marx eds. 1994) ("The idea of history as a record of progress driven by the application of science-based knowledge was not simply another idea among many. Rather it was a figurative concept lodged at the center of what became, sometime after 1750, the dominant secular world-picture of Western culture. That it was no mere rationale for domination by a privileged bourgeoisie is suggested by the fact that it was as fondly embraced by the hostile critics as by the ardent exponents of industrial capitalism. . . . To later followers of Marx and Engels, the most apt name of that power leading to communism, the political goal of progress— of history— is 'technology.'").

301. Leo Marx, *Does Improved Technology Mean Progress?*, TECHNOLOGY REVIEW, Jan. 1987, at 37.

302. Marx, *The Idea of "Technology" and Postmodern Pessimism*, *supra* note 300, at 251.

303. Marx, *Technology, The Emergence of a Hazardous Concept*, *supra* note 284 at 576.

304. Marx, *The Idea of "Technology" and Postmodern Pessimism*, *supra* note 300, at 249.

305. The notion is attributed to Grant. A. Whatmough who introduced the concept in 1976 in an unpublished paper, *Money, Machines, Energy, and Wealth*. *See* Grant. A. Whatmough, *Money, Machines, Energy, and Wealth*, *in* LIVING WITH THE EARTH (Kent A. Peacock ed. 1996). The notion of second-order machines is a more fruitful characterization of programmable computers than the false distinction between general- and special-purpose computers. See *supra* note 26.

306. *See, e.g.*, Christopher G. Langton, *Artificial Life*, *in* SFI STUDIES IN THE SCIENCES OF COMPLEXITY 1, 11 (Christopher Langton ed., 1988) ("Various threads of technological development—programmable controllers, calculating engines, and the formal theory of machines—have come together in the general purpose, stored program computer. Programmable computers are extremely general behavior generators. They have no intrinsic behavior of their own. Without programs, they are like formless matter. They must be told how to behave. By submitting a program to a computer—that is: by giving it a formal specification for a machine—

are devices that extend human capabilities by exploiting a mechanical advantage. . . . A second-order machine is an *assembly* of first-order machines, coupled to produce a multiplying effect."[307] A second-order machine is a system or nested systems of systems of first-order machines. According to Allen, technology is a "threshold phenomenon" that arises once second-order machines begin to predominate wider economies.[308]

The foregoing schematization has several advantages.[309] First, the schematization reveals more clearly what is at stake in the eligibility inquiry for first-order and second-order machines—the natural world and/or the artifactual world of human ingenuity, respectively. Courts have been more explicit about what is at stake for second-order machines. The normative inquiry for second-order machines focuses on how machines interface to other machines, how machines are enmeshed into wider systems within a technologized, artifactual world, a world made possible by human ingenuity. Courts have expressed the normative concern about hampering human ingenuity by invoking the notion of "building blocks."[310] As second-order machines, modern technologies are (re)configurations and (re)combinations of components or "building blocks."[311] As discussed below, the "building block" concern lends itself most directly to the speculative mode.[312]

---

we are telling it to behave as if it were the machine specified by the program. The computer then 'emulates' that more specific machine in the performance of the desired task. Its great power lies in its plasticity of behavior. If we can provide a step-by-step specification for a specific kind of behavior, the chameleon-like computer will exhibit that behavior. *Computers should be viewed as second-order machines-given the formal specification of a first-order machine, they will 'become' that machine.* Thus, the space of possible machines is directly available for study, at the cost of a mere formal description: computers 'realize' abstract machines.") (emphasis added).

307. ALLEN, *supra* note 122, at 107 (emphasis added).

308. *Id.* at 106–08.

309. With their Procrustean tendency, schemas and classifications should invite suspicion. The burden of this part is to show that, whatever its deficiencies, the foregoing schematization can be fruitfully employed in the eligibility inquiry. The foregoing schematization may also serve as a counterexample to Professor Peter Yu's thesis that formalisms in patent law are inherently oversimplifications that reduce the cognitive burdens on generalist judges. Peter Lee, *Patent Law and the Two Cultures*, 120 YALE L.J. 2, 82 (2010) (submitting that "the inquiry-truncating nature of formalism limits the universe of technological facts that judges must consider in deciding patent issues.").

310. *See, e.g.*, Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1303 (2012); Alice Corp. Pty. v. CLS Bank Int'l, 134 S. Ct. 2347, 2356 (2014).

311. John Lienhard, professor of engineering and history, observed that "[t]oday's engineers have reached the point where they invest far more time in the problems of *combining* elements effectively than they spend inventing them in the first place." JOHN LIENHARD, THE ENGINES OF OUR INGENUITY 170 (2000).

312. *Mayo* recognized that the "building block" concern is not easily administrable. *See Mayo*, 132 S. Ct. at 1303 ("And so the cases have endorsed a bright-line prohibition against

In contrast, courts have been more hesitant to articulate normative criteria for first-order machines. For example, in *Diehr*, the normative criterion of industrial efficiency remains implicit.[313] *Diehr* does not explicitly reason how a technical effect (e.g., "transforming an article to a different state or thing"[314]) satisfies a normative criterion (e.g., industrial efficiency). Perhaps courts are not well-suited to regulate first-order machines and to make technocratic policy decisions about their environmental, economic, and social impacts.[315] Perhaps the constitutional term "progress" should be simply construed as simply relating to the dissemination or spread of knowledge. Even so, given the pervasiveness of second-order machines, courts cannot avoid making technocratic determinations about second-order machines as "building blocks" for future technologies.[316] It is an open questions whether the fact that courts are already engaging in such technocratic inquiries would encourage courts to be more explicitly normative (e.g., clearer articulation of values) and/or regulatory in the context of first-order machines.

Second, the schematization reveals more clearly when the eligibility inquiry may likely be speculative.[317] Speculation about second-order machines as "building blocks" for future technologies is likely unavoidable because, especially in an economy in which second-order machines are pervasive, a particular course of technological development is far from inevitable.[318] As Allen observed, "[a]n economy of second-order machines causes technical alternatives to proliferate, continuously enlarging the problems of synergy and the

patenting laws of nature, mathematical formulas and the like, which serves as a somewhat more easily administered proxy for the underlying 'building-block' concern.").

313. *See supra* notes 190–192 and accompanying text.

314. Diamond v. Diehr, 450 U.S. 175, 192 (1981).

315. I am using the term "technocratic [policy decision]" in the sense that policy should be decided by experts (e.g., scientists, engineers, and policy makers) who have a better grasp of the technological possibilities and their impacts. *See, e.g.*, *Flook*, 437 U.S. at 595 ("Difficult questions of policy concerning the kinds of programs that may be appropriate for patent protection and the form and duration of such protection can be answered by Congress on the basis of current empirical data not equally available to this tribunal.") (footnote omitted); *Benson*, 409 U.S. at 73 ("If these programs are to be patentable, considerable problems are raised which only committees of Congress can manage, for broad powers of investigation are needed, including hearings which canvass the wide variety of views which those operating in this field entertain.") (footnote omitted).

316. *See supra* note 308 and accompanying text.

317. *See supra* notes 279–280 and accompanying text.

318. *See, e.g.*, Feenberg, *supra* note 23, at 14 (observing that "the public is constituted by the technologies that bind it together but in turn it transforms the technologies that constitute it. Neither society nor technology can be understood in isolation from each other because neither has a stable identity or form.").

scope of technical choice."[319] If speculation is unavoidable, courts should proceed carefully. For example, the unexamined premise of *Benson's* futuristic vision that includes "operation of a train to verification of drivers' licenses to researching the law books for precedents . . . performed through any existing machinery or future-devised machinery" is that such machinery will not utilize alternative encodings or number systems (e.g., ternary logic of three possible values), both of which have been realized.[320] In retrospect, *Benson's* premise may be justifiable given the patent term and the pace of technological development. However, *Benson's* rhetoric of technological inevitability obscures an unavoidable speculative turn in the eligibility inquiry.

Flook also obscures the speculative turn in the eligibility inquiry. In contrast to *Benson*, which lists possible future technologies, *Flook* addresses the "building block" concern by analogizing the technology at issue to an application of the $2\pi r$ formula to calculate the circumference of a wheel.[321] However, the rhetoric of analogy differs from that of the list. The analogy strives for identity rather than plurality. The analogy, unlike the list, does not positively recite speculations about other possible applications of the formula at issue. The analogy leaves for the reader the unenviable task of completing the argument with speculations about other applications and future technologies. However, for the attentive reader, the analogy is flawed. A circumference cannot be calculated without the $2\pi r$ formula which is a fundamental equation in Euclidean geometry. By contrast, there are many formulas or filters to "smooth" a signal based on its history.[322] Such filters were already widely employed to process signals for a wide range of applications.[323] Rather than avoiding the difficulties inherent in the eligibility inquiry, the Court in *Flook* should have (more

---

319. ALLEN, *supra* note 122, at 111.

320. Gottschalk v. Benson, 409 U.S. 63, 68 (1972) ("Here the 'process' claim is so abstract and sweeping as to cover both known and unknown uses of the BCD to pure binary conversion. The end use may (1) vary from the operation of a train to verification of drivers' licenses to researching the law books for precedents and (2) be performed through any existing machinery or future-devised machinery or without any apparatus.").

321. "[R]espondent's claim is, in effect, comparable to a claim that the formula $2[\pi]r$ can be usefully applied in determining the circumference of a wheel." Parker v. Flook, 437 U.S. 584, 595 (1978).

322. As discussed above, the algorithm at issue in *Flook* is a form of exponential smoothing which assigns to past observations exponentially decreasing weights over time. *See supra* note 72.

323. *See generally* ROBERT GOODELL BROWN, SMOOTHING, FORECASTING AND PREDICTION OF DISCRETE TIME SERIES (1963).

explicitly) considered these existing technologies as "building blocks" in its speculative inquiry about future risks of pre-emption.

Lastly, the schematization provides for the normative inquiry a starting point and a direction for refinement. By initially characterizing a technology as primarily first-order or second-order, courts can rely on normative criteria developed in case law. Next, by considering how a technology exceeds or transcends the schema, courts can refine the normative inquiry. For example, the Court in *Alice* could initially characterize the technology at issue as a second-order machine. At the appropriate level of abstraction, the technology at issue in *Alice* can be characterized a "building block" because, for example, the proxy pattern (i.e., as exemplified by the "shadow record") may be useful in other fields or applications. The Court may inquire whether the proxy pattern "building block" is applicable to and, as claimed, risks pre-empting other applications.

Next, the Court can refine its normative inquiry by considering "first-order" effects of the invention as a second-order machine. For example, the Court may inquire into other kinds of efficiency that are more abstract (for which the relationship to physical quantities is more attenuated). Just as this article's critique of *Altai* suggests a cognitive efficiency, so the dissent in the Federal Circuit, with its analogy to *Diehr*, suggests a transactional efficiency in machine-mediated or machine-machine systems.[324] Thus, freed from the outmoded conception of technology as discrete machines, courts can more clearly see the object of its analysis. For software-implemented inventions, courts can be more receptive to a vision of technology that recognizes the changed nature of the machine.

CONCLUSION

The jurisprudence on subject-matter eligibility in patent law has borne out of Gertrude Stein's observation that there is lag between our world and our conceptions.[325] Just as Karl Marx and his contemporaries

324. "The claims here are analogous to those found patent eligible in *Diehr*. Indeed, the computer system [in *Diehr*] supplied the speed, accuracy, reliability, and automaticity that enhanced and applied the known rubber molding process and formulae. . . . Here, the claim recites a machine and other steps to enable transactions. The claim begins with the machine acquiring data and ends with the machine exchanging financial instructions with other machines. The 'abstract idea' present here is not disembodied at all, but is instead integrated into a system utilizing machines. In sum, the system claims are indistinguishable from those in *Diehr*." CLS Bank Int'l v. Alice Corp. Pty., 717 F.3d 1269, 1310–11 (Fed. Cir. 2013), *cert. granted*, 134 S. Ct. 734 (2013), *aff'd*, 134 S. Ct. 2347 (2014) (emphases added).

325. *See supra* notes 19–22 and accompanying text.

struggled to understand a new technologized world, so courts are faced today with a similar difficulty. Courts have presumed that contemporary legal thought is equipped, conceptually and linguistically, to understand the full significance of modern technologies. I submit that § 101 jurisprudence may bear out Stein's observation that we remain "at least several generations behind [our]selves."[326]

This article examined how juristic formulations reveal the limits and limitations of our concepts and our language.[327] The limitations of legal thought in understanding a complex technological reality should not be an unpalatable idea for the history of scientific thought also reveals the limited reach of our concepts in our attempts to understand a complex physical reality. For example, the classical concepts of "particle" and "wave" fail to fully explain the phenomena of light. Einstein observed that "[w]e are faced with a new kind of difficulty. We have two contradictory pictures of reality; separately neither of them [particle or wave theory] fully explains the phenomena of light, but together they do."[328] Similarly, with software, we are faced with an entirely new technological artifact that seems to be at once a writing and a machine.[329] Software reveals its dual nature in copyright law as software-as-writing and in patent law as software-as-machine.

This article argues that patent law can learn from copyright law. Copyright law teaches that the reception of software-as-writing in the eligibility inquiry should involve a reading that is faithful to the actual process of its creation. This article's critique of *Altai* shows that the creative process of writing software can be described in terms of design

326. Stein, *supra* note 20, at 521.

327. *See, e.g., supra* notes 24–27, 64–65, 119, 126–129, 254–257, 275–278, 286–292 and accompanying text. I submit that Knuth's observation about programming languages applies also to jurisprudential language: "[t]he language in which we express our ideas has a strong influence on our thought processes." Knuth, *Structured Programing*, *supra* note 169, at 275.

328. ALBERT EINSTEIN & LEOPOLD INFELD, EVOLUTION OF PHYSICS 263 (1967).

329. The scholarship on the nature of software remains rudimentary. In *The Ontology of Cyberspace*, David Koepsell, professor of philosophy and law, observed that "[a]s in languages, whose flexibility accommodate a possibly infinite number of forms of expression for any general idea, computers may be programmed a possibly infinite number of ways. This flexibility has demonstrated that atoms and bits, like language, are just other media of expression." *See* DAVID R. KOEPSELL, THE ONTOLOGY OF CYBERSPACE: LAW, PHILOSOPHY, AND THE FUTURE OF INTELLECTUAL PROPERTY 103 (2000). While Koepsell correctly suggests that emerging technologies such as software have revealed conceptual difficulties in intellectual property law, Koepsell fails to closely examine the dual and unique nature of software. One critic laments that Koepsell treats software as ontologically unexceptional, and is unwilling to take seriously the difficult questions surrounding the unique novelty of software. Rita F. Lin, Note, 14 HARV. J. L. & TECH. 315, 329 (2000–01) (reviewing DAVID R. KOEPSELL, THE ONTOLOGY OF CYBERSPACE (2000)).

choices (unfolding a grammar of creation, as it were). Like choice of words and turns of phrases, design choices in the creative process of writing software have consequences that reverberate through the whole edifice, through its hidden couplings and abstractions to its external aspects. Just as a literary scholar or critic cannot fully attend to the deeper movements of a text without a comfortable fluency in the language, so jurists cannot explicate the full significance of the object of its analysis without being technologically literate.[330] The analysis of the software at issue in *Altai* revealed that its external aspects (e.g., parameter lists and macros) intimate a hidden architecture that is structured by design choices, the reasons for which are determinative of how those external aspects are characterized. Similarly, the analysis of the software at issue in *Alice* revealed that, at a lower level of abstraction, its external aspects (e.g., the shadow record) are integrated into a hidden architecture, the whole of which is more than the sum of its parts. In the final analysis, without a careful inquiry into the creative process, courts cannot properly determine the appropriate level of abstraction for claim construction—a critical determination in the patentable subject-matter eligibility inquiry.[331]

330. Here, *technological literacy* means not only a fluency in the technical language of computer science but also an informed understanding of the nature of computer technology. Yet, the metaphor of literacy may still lead us away from a needful kind of software criticism that this article has elaborated, from an access to and a critical reception of the human act of creation. "Even the transposed, groping metaphor of computer literacy indicates a latent concern for something more in the notion of literacy than the sheerly instrumental value of an alphabet that can aid communication . . . on a deeper ontological level, the reading and writing of books has significance beyond the merely instrumental. The book defines a certain way in which access to things in the world is fostered, a certain modeling of the way things are held up to attention. Literacy is as much an expression of the way a person inhabits the world as it is an instrument within the world of human concerns." *See* MICHAEL HEIM, ELECTRIC LANGUAGE: A PHILOSOPHICAL STUDY OF WORD PROCESSING 167 (1987).

331. *But see* Chiang, *supra* note 78, at 1152 (arguing compellingly that the level-of-abstraction problem has been obscured in patent law but ultimately presenting a false choice that "[s]election [of the appropriate level of abstraction] can be done either by the transparent [economic] balancing of competing interests or by the invocation of absolutist yet contradictory rules that yield no coherent principle.").