JOSHUA WALKER

ON LEGAL AI

ON LEGAL AI

Joshua Walker

fastcase FULL COURT PRESS

A Full Court Press, Fastcase, Inc., Publication.

Printed and bound in the United States of America.

10 9 8 7 6 5 4 3 2 1

ISBN (print): 978-1-949884-07-4
ISBN (online): 978-1-949884-08-1

For my children

That the darkness we may see be vanquished … before realized.

Contents

Acknowledgments

There are far too many people to thank than can be appended here. Most of all, there are people to thank for the opportunities and works described herein. Such *actual* works will far outlive and outperform this one. The convictions in the Rwandan war crimes tribunal, the ICTR, are an extraordinarily obvious example. But two heroes, Michael Hourigan and Amadou Deme, should be called out for their unconquerable courage and dynamism in that difficult setting. Many others too. For example, the achievements of Lex Machina belong to a very large set of creators and influencers, including at Stanford and well beyond. It took the creativity and chutzpah of many to realize it. Judge Vaughn Walker (ret.) was and remains a hero of a different kind, in that and numerous other legal settings. He reminds us that justice transcends—indeed, is the antivenom of—all politics.

For the book, I particularly thank Morgan Morrissette Wright, Sharon D. Ray, Steve Errick, and Ed Walters. Morgan and Sharon and Steve heavy-lifted with brilliance of action. But Ed also had that primordial entrepreneurial chutzpah. All others, from all eons, thank you too.

The errors in this volume are my own. I release it to catch them, and evolve:

E.
D.
E.
N.

Introduction

This book is designed to make you a more valuable attorney. By introducing the concept of the "legal engineer," and other disruptive principles, it seeks to catalyze your transformation from a "check the box" cost center (the enterprise version of a circulatory hemorrhage), to a core part of product development and business growth. This book is designed to help you change—both in fact and with regard to how others perceive you—from a roadblock to a creator of provable enterprise value.

How? We explore how you can improve technical and business outcomes of artificial intelligence (AI) and related software efforts. We explore how to reframe your role and retime your interventions. We explore how to translate legal outcomes into other languages: from qualitative to quantitative, and back again. If this is a treatise, it is a *fast* treatise—one focused on your results, not academic pretensions.

Why read it? Why complete the exercise? If you are a veteran attorney, the answer is obvious. You have yourself heard the "financial hemorrhage" metaphor applied to legal procedures and professionals, or similarly unflattering comparisons. If you are a student, or just starting out, what you will eventually yourself—or here virtually—learn is that the creative life of a useful enterprise, and its reins, are rarely held by so-called "cost centers"/"unenlightened" attorneys (and when they are it usually signals dire straits). Attorneys are generally not perceived as people who add value, get things done, creatively solve problems, or positively contribute to the company's bottom line.

Thus memed, thus: legal "cost centers" are constantly at risk of being shaved down, cut radically, ignored in critical planning, sidelined, or worse. In most cases, the reins of corporations are held by those perceived to be "generators" or maintainers of value. While perception rarely reflects actuality, in my experience attorneys have nonetheless not been the creative problem solvers we could be. So consider *OLAI* a kind of creative and technical "boot camp," because **that** is the profession the future impels.

Reality? The world needs legal engineers more than computational mores, more than unduly abstract advisers, and more than competent drafters at the rear end of deals, projects, and conflicts. The world needs legal programming as code, designed in from the design phase. Legal insight on creation. We talk much more about what this means, with specific examples.

While creativity and "legal engineering" are important, mere philosophical screed will skim off the surface of a busy practice like a flat stone skipped over a frozen lake. Undue abstraction will *accentuate* the separation of our profession from others, instead of achieving immersion, admixture. Thus, this book has a far more practical goal: to help you get better at what you *already* do, by leveraging AI platforms and data for legal procedures and strategy. This includes how to use AI and "big data" to (i) better understand, (ii) guide, and (iii) support legal decisions (including to boards of directors, CFOs and other potentially severe or opaque corporate audiences), as well as in drafting, governance, and strategic contexts. Yes, we adopt certain engineering practices (or, at least, "memes"). But we also learn how to apply engineering *systems*—software and hard data—to actual legal work. Using simple tools, we may concurrently improve how law is practiced, and, more importantly, how legal ecosystems are architected, measured, and improved.

Lastly, at the apex of memes and practice, of philosophy and tactics, we address the creation of legal AI and AI governance systems *from the ground up*. One trusts that this third element will help drive not only your clients, but the world at large, to optimize AI. This is what I mean in saying that the world needs "legal engineering" more than computational mores. Law must generally and very literally govern advanced AI. Thus, we explore **both** how law can help AI, and how AI can help you.

That said, while the textual artifact you hold in your hands was induced laboriously over decades of trial and error, it remains just a recipe. It is up to you to bake the cake.

PART I

Tabula Rasa

CHAPTER 1

Essential Codes

As DNA is the code of known life, so law is the code of human society.

Computer code, in turn, was originally fomented to run legal code—amongst other arguments.

To run legal code, computer code had to encompass all states of the world, all of its abstract and physical objects, and it had to describe these things in a highly structured and consistent manner, such that they could be combined and operated upon like quantities in a calculator.

In order to compute complex legal, diplomatic, and scientific arguments, computer code needed mathematical precision but virtually *unlimited* qualitative scope. It was a paradox unrivaled in the history of science, and the challenge was arguably first assumed by a strange autodidact from the 1600s.

A. Binary Dreams: Polymath, Polyword, Pollywog

In 1672, an attorney originally employed to re-write the legal code of the Electorate of Mainz[1] was sent by the Elector

[1] A governmental division of the Holy Roman Empire, nestled within present day Germany.

to the court of Louis XIV, to help resolve certain legal and diplomatic claims.[2] To avoid war.

Despite—or perhaps because of—the scope and difficulty of his actual job, this attorney found time for certain other pursuits; pursuits that inevitably trickled into the broader stream of his professional mission.

An autodidact in mathematics, his collision with numerous intellectuals in Paris (including a stellar and pioneering Dutch astronomer) improved his knowledge up to the state of the art in mathematics, and accelerated him beyond it, into the realm of automated calculation. Amongst his earliest experiments: trying to build the first mechanical calculator capable of multiplication and division—which he accomplished. On the basis of his novel mechanical calculator, and despite his lack of formal training, he was admitted to the Royal Society, one of the world's oldest and most august scientific assemblies.

Our intrepid attorney's rapid success in mathematics contrasted with his rapid failure in diplomacy. Soon after his arrival in Paris, open war broke out between the French and the Dutch. But let us not assume that correlation is causation!

Whatever the practical and immediate causes of war, on an abstract level this attorney's ability to move the dispute forward to proper, logical, or objective resolution was obscured by the very opaqueness and indefiniteness of the legal arguments underlying the positions of both sides to the dispute. Indefiniteness creates conflict. It invites disparate interpretation, and it gives parties the freedom to take extreme positions.

Frustrated, he imagined a radically different method of dispute resolution:

> At the outset of any legal, diplomatic, or scientific dispute the parties would input their evidence and arguments into a machine, then say "Let us compute!" Much as his

[2] See generally Martin Davis, *The Universal Computer: The Road from Leibniz to Turing* (W.W. Norton & Co., 2000), 3-20.

> advanced calculator consumed numbers and symbols to output quantities, this "concept processor" would input predefined conceptual symbols, and output the objectively correct decision, purely through mechanical process.... No more conflicts/wars/lawsuits.

To realize his dream, the attorney knew he needed to create two things:

1. **The Machine**, the dispute resolution processor (which we still quaintly call a "court system"); and, perhaps more importantly,
2. **A Concept Language** able to universally and consistently characterize any mental or material object (which we most closely approximate with so-called natural language,[3] e.g., English, Hindi, Mandarin, etc.).

Though the gears would still need to turn, most of the analytical work would *already* have been done by the symbology itself. In other words, the very fabrication of the symbology, and its pre-input **mapping** to real-world concepts, would enable the subsequent **mechanistic** and objective post-processing.

Again, the machine would leverage the preexisting ontological classifications inherent in the computational language, including its foundational definitions (almost like a "calculus" uses its notation to make the calculation easier). Thus, creating this "universal concept language" was the attorney's first major task.

---

[3] Many would call legal language "un-natural language," and thus, attorneys "unnatural language processors." However, the desire to formalize "loosey-goosey" prose into something clear, precise, and objective is actually one of the constant quests that makes the legal profession so closely akin to the academic neighbor addressed in these pages: computer science/code.

Modifying concepts from the ancient Chinese I Ching[4] and other linguistic precedents, he repackaged these symbologies into something like modern binary. Then the attorney began defining concepts from the real world as discrete mathematical entities, all processable by machine. . . .[5]

Thus, modern binary was conceived.

For non-geeks, "binary" refers to a mathematical system using only two symbols—"1" and "0." At time of writing, it is used by all computers to encode data.[6] Each digit is referred to as a "bit," and can be physically implemented in circuitry using logic gates. **This is a telling example of how something profoundly abstract can end up being foundationally practical.** No contemporaneous computer works without instantiating the binary concept.

And not only modern binary! Our intrepid attorney also helped lay the foundation for a host of related first-order concepts in computer science and information theory. See generally Chapter 9.

In other words, in the foment and froth of this person's mind, law and computation were conceived as complimentary systems—the latter to solve the former, amongst other dispute classes. Legal, diplomatic, and

---

[4] Included amongst the five classics of Confucianism, the I Ching is a method or manual of divination based on eight symbolic trigrams and 64 hexagrams, interpreted in terms of the principles of yin and yang. The specific precedent is less important than the idea of creating a scalable method of symbolic manipulation. The full title of Leibniz's original article on the subject is "Explanation of Binary Arithmetic, which uses only the characters 1 and 0, with some remarks on its usefulness, and on the light it throws on the ancient Chinese figures of Fu Xi" (1703).

[5] See additional background in Chapter 9: A Brief History of Law and Computer Science, and cites therein. The author is especially indebted to the work of Martin Davis. See related citations.

[6] Even quantum computers, in early evolution contemporaneously, arguably use a variant of binary code.

scientific disputes weren't particularly differentiated, in this worldview.

The former classification and procedural system—the law—contained the germ, the fricative matrix, from which the latter computational schema evolved. Real-world legal dilemmas, even the statutory ontologies of Mainz, helped generate the theoretical foundations for modern computer science.

True story, if glibly told.

As you may have already grokked, Googled, or guessed, the counselor[7] in question was Gottfried Wilhelm Leibniz, and you may know him as:

- Coinventor of calculus,[8]
- Famous/notorious philosopher,
- Literal and figurative *polymath.*

---

[7] For a more detailed discussion of Leibniz's legal education, scholarship, and legal work generally, see Chapter 9, pp. 104-107, and sources cited therein. Essentially, Leibniz had between two and three degrees in law (depending how you count in polymathematics and philosophy), including a doctorate, and essentially zero degrees in math (unless you count Aristotelian logic, in his first BA). Granted, a "philosophy" degree back then could have covered "natural philosophy" (i.e., studies preceding chemistry or any other hard sciences, as well as mathematics, to some extent). But the point is that the vast bulk of his pedagogy was grounded in law, not math. Ditto for most of his titles, salaried work, etc., excluding genealogical work, aristocratic tutelage, etc. See generally Davis, Ch. 1.

[8] Or not, if you hail from the Isaac Newton–dominated Britain or its post-colonial progeny. Newton and his intellectual allies, including the likes of Voltaire, successfully labeled Leibniz a thief or plagiarist for "stealing calculus" from Newton. Allegedly he was exposed to an early draft of one of Newton's unpublished articles during a diplomatic side trip to England. That is why many of us, including me, don't hear of Leibniz at all until we stumble across him in an old bookstore basement in Palo Alto … or read this book. The Newton–Leibniz conflagration was not dissimilar to the US East Coast versus West Coast rapper conflicts of the late twentieth century.

More relevant to our explorations: Leibniz is considered one of the most important pioneers in what would become computer science, and not just because he worked out the framework and applicational structure for modern binary.[9]

We will expand upon this story and its implications a bit later. But the punch line is that law was at the origin point of modern binary, computational systems and languages. The legal-aristico-politico disputes of the 1700s were the "sand in the gears" of the mind of the man that helped lay the foundations for modern computer science.

The world mostly forgot this. Certainly, no one remembers Leibniz as a "lawyer." And many of his fans from the computer science and mathematics world may violently reject this characterization, *in utero*. But law dominated his formal training, and his original "day job."[10]

Understanding that common "hydrological origin"—the riparian nexus of these two intellectual streams, law and computer science—will help us at the delta, in the fractured

---

[9] See, e.g., Chapter 9 (citing Norbert Wiener). See generally Martin Davis, *The Universal Computer*, pp. 3-20. The "universal characteristic" alone is a useful concept for investigation: "For one could always say: let us calculate, and judge correctly through this, as much as the data and reason can provide us with the means for it." See, e.g., Wikipedia on "Characteristic Universalis" available at https://en.wikipedia.org/wiki/Characteristica_universalis, as of October 2018, citing Strickland 2011 at 355 ("[t]his characteristic consists of a script or language .... The characters would be quite different from what has been imagined up to now. Because one has forgotten the principle that the characters of the script should serve invention and judgment as in algebra and arithmetic"). P.S. Most would reject a citation to Wikipedia as entirely inappropriate to an alleged treatise. True for a legal finding or case. Wildly inefficient for light references to historical or introductory technical matter. The Internet exists. I commend it to the regular use of new attorneys, if only for introductions and as a "finder scope" for source material of denser structure (e.g., proven case law, peer-reviewed technical papers, published patents, etc.).

[10] See, again, Chapter 9, pp. 104-107.

practical domain where you and I have to get real work done for real clients with real problems, typically under severe time constraints.

Thus, the point of this story is not to bore you with history, it is to arm you with actionable data (which happens to be historical). The above intellectual correlations, the "hydrological nexus" between law, computer science, and even AI, are more than just curiosities. They serve as **practical tools** to help you navigate a legal future deeply entwined with AI and empirical data.[11] This future is happening now, and that "happening" will undoubtedly persist through many cycles of time and hype. Thus, leverage your history, and it will pay compounding dividends.

Leibniz imagined dispute resolution machines. Today we are realizing them, application by highly imperfect application. These applications are not the idealized, "Harz Mountain"[12] dreams of a philosopher. They are gritty, messy lines of code, colliding with real-life data and globally networked hardware platforms, and are rife with security, privacy, and legal ethics problems. But they ultimately bring us closer to Leibniz's vision: objective resolution of complex conceptual problems, at machine speed.

Allegedly, this includes blockchain[13] powered "smart contracts." To be determined is whether "smart contracts" ultimately become successful parts of the world's infrastructure, or automated weapons of "math destruction" (as Warren Buffett and others have described credit default swaps and other automated financial tools and option contract cascades),

---

[11] See, e.g., Chapter 10C.

[12] See, e.g., Davis at 1. Leibniz tried and failed to build a complex series of windmills to automatically pump out water from silver mines in present day Germany. Thus, think of this particular over-engineered "Harz Mountain" scheme as kind of a Rube Goldberg machine.

[13] For a skeptical note on a truly promising tech, see "Special K, Meta K" at Chapter 19B.

unchecked and out of control in their consequences. But the vision of these systems is *straight from Leibniz's playbook:* a pre-encoded, executable contract, written as computer software (which software and data are ultimately fixed in binary), is effectuated and resolved automatically, with the results occurring automatically and verifiably through the blockchain.[14] Again:

> At the outset of any legal, diplomatic, or scientific dispute the parties would input their evidence and arguments into a machine, then say "Let us compute!" Much as his advanced calculator consumed numbers and symbols to output quantities, this "concept processor" would input predefined conceptual symbols, and output the objectively correct decision, purely through mechanical process. . . .

No more trouble trying to figure out who is right and wrong about a contract clause. The contracts, in theory, resolve themselves automatically. In fact, in our "smart contracts" use case, the parties need not even say "let us compute." Computation happens automatically and in real time, as the events unfold.

Again, whether anyone has really thought out the consequences of cascades of interdependent and interconflicting automated contracts is another question, for another chapter. (But see the impact of credit default swap collapses and the Great Recession of 2008 for interim fodder, to seed your imagination.) However, automated encoding of real-world

[14] "Blockchain" refers to a distributed database system with encryption (along with, perhaps, an accompanying social movement). A precise definition is beyond the scope of this legal treatise, but blockchain infrastructures are frequently touted as a means to verify transactions without trusted intermediaries, even between two parties that do not know each other.

events and accompanying legal structures, and automated/mechanical resolution flows straight from the high concepts of our 1600s autodidact.

From Leibniz to blockchain there is a straight line.

Introducing *On Legal AI*

This book is designed to make you a more valuable attorney. By introducing the concept of the "legal engineer," and other disruptive principles, it seeks to catalyze your transformation from a "check the box" cost center (the enterprise version of a circulatory hemorrhage), to a core part of product development and business growth. This book is designed to help you change—both in fact and with regard to how others perceive you—from a roadblock to a creator of provable enterprise value. How? We explore how you can improve technical and business outcomes of artificial intelligence and related software efforts. We explore how to reframe your role and retime your interventions. We explore how to translate legal outcomes into other languages: from qualitative to quantitative, and back again. If this is a treatise, it is a fast treatise—one focused on your results, not academic pretensions.

fastcase FULL COURT PRESS

$24.95
ISBN 978-1-949884-07-4
52495>
9 781949 884074