

University of Colorado Law School

Colorado Law Scholarly Commons

Publications

Colorado Law Faculty Scholarship

2013

Contract as Pattern Language

Erik F. Gerding

University of Colorado Law School

Follow this and additional works at: <https://scholar.law.colorado.edu/faculty-articles>



Part of the [Banking and Finance Law Commons](#), and the [Contracts Commons](#)

Citation Information

Erik F. Gerding, *Contract as Pattern Language*, 88 WASH. L. REV. 1323 (2013), available at <https://scholar.law.colorado.edu/faculty-articles/111>.

Copyright Statement

Copyright protected. Use of materials from this collection beyond the exceptions provided for in the Fair Use and Educational Use clauses of the U.S. Copyright Law may violate federal law. Permission to publish or reproduce is required.

This Article is brought to you for free and open access by the Colorado Law Faculty Scholarship at Colorado Law Scholarly Commons. It has been accepted for inclusion in Publications by an authorized administrator of Colorado Law Scholarly Commons. For more information, please contact rebecca.ciota@colorado.edu.

HEINONLINE

Citation: 88 Wash. L. Rev. 1323 2013

Provided by:

William A. Wise Law Library



Content downloaded/printed from [HeinOnline](#)

Fri Feb 24 13:37:01 2017

- Your use of this HeinOnline PDF indicates your acceptance of HeinOnline's Terms and Conditions of the license agreement available at <http://heinonline.org/HOL/License>
- The search text of this PDF is generated from uncorrected OCR text.
- To obtain permission to use this article beyond the scope of your HeinOnline license, please use:

[Copyright Information](#)

CONTRACT AS PATTERN LANGUAGE

Erik F. Gerding*

INTRODUCTION

Scholars and practitioners routinely talk about the “architecture” of individual contracts.¹ Many observers have also noted the broad-brush similarity between the drafting of legal contracts and computer programming or coding.² It is strange, then, that contract law scholarship has overlooked part of the landmark literature linking design in architecture and computer code.³ In 1977, Christopher Alexander, a professor of architecture at the University of California, Berkeley, drew

* Associate Professor, University of Colorado Law School. I would like to thank Fred Bloom, Anna Gelpern, Paul Ohm, and Harry Surden for comments on this article. This Article was made possible by a summer research grant from the University of Colorado Law School. The author has no financial interests that are the subject of this work or that influenced this work.

1. For example, Larry Cunningham, who is being honored with this symposium, has written about the architecture of contracts in the context of using XML (extensible markup language) in corporate contracting. Lawrence A. Cunningham, *Language, Deals, and Standards: The Future of XML Contracts*, 84 WASH. U. L. REV. 313, 324 (2006). For a small sample of other recent scholarship discussing the “architecture” of contracts, see Anna Gelpern, *Commentary*, 51 ARIZ. L. REV. 57, 64–65 (2009) (discussing the “architecture” of the standardized form contracts for derivatives); Scott J. Burnham, *How to Read a Contract*, 45 ARIZ. L. REV. 133, 142–45 (2003) (arguing for reading contracts by examining their architectural structure); Robert P. Bartlett, III, *Commentary*, 51 ARIZ. L. REV. 47, 50 (2009) (describing “how the architecture of contracts across a variety of domains” seeks to curb parties taking advantage of one another after the contract has been executed).

2. For an academic article analogizing contract drafting to computer coding, see Henry E. Smith, *Modularity in Contracts: Boilerplate and Information Flow*, 104 MICH. L. REV. 1175, 1190 (2006). The author of one book on contract drafting uses the computer metaphor to make a strong point on writing style. He writes: “[c]ontract prose is limited and highly stylized—it’s analogous to computer code. It serves no purpose other than to regulate the conduct of the contract parties, so any sort of writerly ‘voice’ would be out of place.” KENNETH A. ADAMS, *A MANUAL OF STYLE FOR CONTRACT DRAFTING*, xxvii (2d ed. 2008). For an extreme example that moves beyond analogy and describes contract and other legal rules as a form of computer code, see Alexey V. Lisachenko, *Law as a Programming Language*, 37 REV. CENT. & E. EUR. L. 115 (2012).

3. Some legal scholarship has cited *A Pattern Language*, the inspiration for this essay. See *infra* note 4. However this legal literature focuses primarily on property, urban planning, and assorted other subjects. See e.g., Henry E. Smith, *Property as the Law of Things*, 125 HARV. L. REV. 1691, 1708 (2012) (using pattern language to discuss modularity of property rights); Braham Boyce Ketcham, *The Alexandrian Planning Process: An Alternative to Traditional Zoning and Smart Growth*, 41 URB. LAW. 339, 343 (2009) (discussing lessons of Alexander’s work for city planning).

upon his background in computer science to co-author *A Pattern Language*.⁴ This practical book—together with Alexander's more theoretical companion volume, *The Timeless Way of Building*,⁵ and his other work⁶—provided an influential blueprint for architects, urban planners, and the reading public who sought a more organic, humanistic, and democratic way of designing buildings and cities amidst the failures of modern urban renewal and widespread dissatisfaction with the course of architectural modernism.⁷ This article examines how Alexander's pattern language framework explains how attorneys draft contracts,⁸ including in response to the types of legal design problems illustrated in Larry Cunningham's book, *Contracts in the Real World*.⁹ Moreover, the pattern language rubric explains how individual legal agreements interlock to create complex transactions, and how transactions interconnect to create markets.¹⁰ Furthermore, this pattern language framework helps account for recent evidence, including from the global financial crisis, of failures in modern contract design, even in cases where sophisticated financial firms and their lawyers were the architects.¹¹

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

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

6. *E.g.*, CHRISTOPHER ALEXANDER, *A NEW THEORY OF URBAN DESIGN* (1987); CHRISTOPHER ALEXANDER, *NOTES ON THE SYNTHESIS OF FORM* (1964); CHRISTOPHER ALEXANDER, *THE LINZ CAFÉ* (1981); CHRISTOPHER ALEXANDER ET AL., *THE OREGON EXPERIMENT* (1975); CHRISTOPHER ALEXANDER ET AL., *THE PRODUCTION OF HOUSES* (1985).

7. See Emily Eakin, *Architecture's Irascible Reformer*, N.Y. TIMES, July 12, 2003, <http://www.nytimes.com/2003/07/12/books/architecture-s-irascible-reformer.html> (describing popularity of Alexander, as well as his disputes with architectural establishment); Ketcham, *supra* note 3 (examining implications of Alexander's work on urban planning). Alexander's work meshed with that of other writers in urban planning, such as Jane Jacobs. JANE JACOBS, *THE DEATH AND LIFE OF GREAT AMERICAN CITIES* (1961). Jacobs and her movement fought the urban renewal exemplified by New York City's master planner, Robert Moses, which included razing of neighborhoods for high rise complexes and construction of large new superhighways slicing through the middle of dense urban areas. ROBERT A. CARO, *THE POWER BROKER: ROBERT MOSES AND THE FALL OF NEW YORK 1963–76* (1974).

8. See *infra* Part II.A.

9. LAWRENCE A. CUNNINGHAM, *CONTRACTS IN THE REAL WORLD: STORIES OF POPULAR CONTRACTS AND WHY THEY MATTER* (2012). This book, which prompted this symposium, serves as an accessible and excellent sample of some of the recurrent design problems for all contracts. Cunningham overlays those problems created by client objectives with those imposed by the common law doctrines of contracts. One could read Cunningham's book not only as a supplement to a first year law school course in contracts, but also as a catalogue of the challenges, design flaws, and design failures that transactional attorneys routinely face. See *infra* Part II.B.

10. See *infra* Part II.D.

11. See *infra* Part III.

Alexander and his collaborators described a series of “patterns” or design solutions for buildings that meet specific environmental needs of individuals. A pattern represents an encapsulated abstract or conceptual solution to a recurring design problem. Patterns thus free architects and designers from having to reinvent the wheel; they can use the solutions that evolved over time as designers in the past grappled with, and crafted answers to, similar problems.¹² In Alexander’s work, a pattern describes a particular solution that can be used to plan growth in a particular region, city, or neighborhood, to design homes or other buildings, or to create rooms or spaces within a building. Interlocking individual patterns create larger design patterns, which, in turn, connect to form still larger patterns.¹³ Thus, patterns for rooms and structural elements combine to create design patterns for buildings. Arranged together, patterns for buildings form patterns for neighborhoods. Patterns for neighborhoods join to create patterns for cities and regions. Through scaling and rules that define when patterns fit together, Alexander’s system created a larger “language” for architectural design.¹⁴ Alexander’s architectural ideas inspired computer programmers who borrowed his idea of patterns that solved particular environmental problems and that connected into a

12. Doug Lea, *Christopher Alexander: An Introduction for Object-Oriented Designers*, SOFTWARE ENGINEERING NOTES (ACM SIGSOFT, New York, N.Y.), Jan. 1994 at 39. Lea describes patterns as encapsulated in that they are, “[i]ndependent, specific, and precisely formulated enough to make clear when they apply and whether they capture real problems and issues, and to ensure that each step of synthesis results in the construction of a complete, recognizable entity, where each part makes sense as an in-the-small whole.” *Id.* at 42. Lea also notes that “abstraction” is a critical quality of an Alexandrian pattern. *Id.*

13. ALEXANDER, *supra* note 5, at 186–91. Doug Lea describes two related qualities to Alexander’s patterns:

Openness. Patterns may be extended down to arbitrarily fine levels of detail. Like fractals, patterns have no top or bottom . . . [and] Composibility. Patterns are hierarchically related. Coarse grained patterns are layered on top of, relate, and constrain fine grained ones. . . . Most patterns are both upwardly and downwardly composable, minimizing interaction with other patterns, making clear when two related patterns must share a third, and admitting maximal variation in sub-patterns.

Lea, *supra* note 12, at 42.

14. See ALEXANDER, *supra* note 5, at 187–91; see also *infra* Part I.A. The language metaphor is not a loose one. In fact, Alexander developed his theory of an architectural pattern language by borrowing heavily from Noam Chomsky’s research on linguistics. See TOM TURNER, CITY AS LANDSCAPE: A POST POST-MODERN VIEW OF DESIGN AND PLANNING 30 (1996). Alexander sought to create a “generative language” for design, a system in which particular patterns would serve as a vocabulary and which had rules or syntax establishing when certain patterns fit together to make intelligible “sentences.” ALEXANDER, *supra* note 5, at 183–87; Janet Finlay et al., *Pattern Languages in Participatory Design*, in PEOPLE AND COMPUTERS XVI - MEMORABLE YET INVISIBLE: PROCEEDINGS OF HCI 2002 160, 164 n.1 (Xistine Faulkner et al. eds., 2002) (linking Alexander’s pattern language to Noam Chomsky’s idea of “generative grammar” in linguistics); see also *infra* Part I.B.

larger language to meet their own design needs.¹⁵ The concepts in *A Pattern Language* shaped a generation of new computer languages and approaches to coding, particularly object-oriented programming.¹⁶

Alexander's work also provides a unique lens to look at how transactional attorneys draft contracts. However, the pattern language framework does much more than explain the function of contractual boilerplate or the process of assembling particular contracts. It also describes how individual contract patterns form complex transactional patterns, and how, in turn, complex transactional patterns form complex financial markets. For example, transactional attorneys arrange individual patterns for provisions in legal agreements—e.g., the basic provision establishing the loan of money in exchange for interest and principal repayments, representations and warranties, covenants, provisions defining default, and remedies—to form legal agreements, such as mortgages or bond indentures. Patterns for separate contracts connect to create transactions. For example, a mortgage, note, deed of sale, and other agreements operationalize the purchase of real estate. Patterns for simple transactions fit together to create more complex transactions and even markets. For instance, mortgage documents, pooling and servicing agreements, trust documents, and indentures create mortgage-backed securities.¹⁷ Patterns of complex transactions and markets, in turn, create more complex financial systems. To extend the examples above, mortgage-backed securities form part of a web of connected financial instruments and markets, called the “shadow banking system,” that connect consumer and commercial borrowers to investors in capital markets.¹⁸

By contract “patterns,” I mean an encapsulated solution within a legal agreement (or set of agreements) to a specific legal problem. This

15. See *infra* notes 76–77 and accompanying text.

16. See *infra* notes 76–77 and accompanying text.

17. See generally *infra* Part II.D. For primers on the securitization process, see Erik F. Gerding, *Code, Crash, and Open Source: The Outsourcing of Financial Regulation to Risk Models and the Global Financial Crisis*, 84 WASH. L. REV. 127, 147–51 (2009); Steven L. Schwarcz, *The Alchemy of Asset Securitization*, 1 STAN. J.L. BUS & FIN. 133, 135–36 (1994).

18. See *infra* Part II.D. For some of the recent legal and economic scholarship on this shadow banking system, see GARY B. GORTON, SLAPPED BY THE INVISIBLE HAND 27–28 (2010); Morgan Ricks, *Shadow Banking and Financial Regulation* 3 (Columbia Law & Econ., Working Paper No. 370, 2010), available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1571290; ERIK F. GERDING, LAW, BUBBLES, AND FINANCIAL REGULATION 395–469 (2013); Erik F. Gerding, *The Shadow Banking System and Its Legal Origins* 1, 32–33 (Aug. 23, 2011) (unpublished manuscript) (available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1990816); ZOLTAN POZSAR ET AL., SHADOW BANKING, 8 (2010), available at http://www.newyorkfed.org/research/staff_reports/sr458.pdf.

problem might consist of a need to match the particular objectives of both counterparties in a discrete part of a bargain. The solution might also address a certain feature of the legal environment, such as one of the contract doctrines examined so colorfully in Cunningham's book. Lawyers can repeat and adapt a contract pattern each time a version of that problem, whether miniaturized or supersized, appears. Each contract pattern interlocks, nests, and works together with other contract patterns to solve more complex problems and create more intricate and elaborate bargains.

Moreover, interlocking patterns enables scalability. That is, arrangements of individual patterns form larger patterns, which combine with other patterns to form still larger patterns. Just as Alexander's patterns for rooms create patterns for buildings, which create patterns for neighborhoods and then cities, so then patterns of individual contract provisions form legal agreement patterns, which interlock to create patterns for transactions, which, in turn, mesh to create patterns for markets. Larger patterns solve larger problems and can meet more complex demands of a greater range of counterparties. This scalability differentiates contract design from contract boilerplate. It also highlights how contract patterns are different than other examples of preformulated language in the law, such as writs or pleadings in procedural law.

This essay examines how patterns enable the transformation of contractual provisions into contracts, contracts into transactions, and transactions into markets. Although contract design patterns are broader than contract boilerplate (as described in Part II.C. below), some of the extensive legal scholarship on boilerplate¹⁹ helps explain how contract patterns generate agreements, transactions, and markets. The work of Henry Smith on the modularity of contract boilerplate proves particularly useful in this regard.²⁰ Contract patterns perform several functions. Contract patterns break complex problems and bargains into components.²¹ Attorneys can then repeatedly apply these particular

19. For a sample of this extensive literature, see Marcel Kahan & Michael Klausner, *Standardization and Innovation in Corporate Contracting (Or "The Economics of Boilerplate")*, 83 VA. L. REV. 713 (1997) (analyzing how standardization in corporate contracts results from "learning" and "network" externalities); BOILERPLATE: THE FOUNDATION OF MARKET CONTRACTS (Omri Ben-Shahar ed., 2007); see also Douglas G. Baird, *The Boilerplate Puzzle*, 104 MICH. L. REV. 933 (2006); Michelle E. Boardman, *Contra Proferentem: The Allure of Ambiguous Boilerplate*, 104 MICH. L. REV. 1105 (2006); Robert A. Hillman, *Online Boilerplate: Would Mandatory Website Disclosure of E-Standard Terms Backfire?*, 104 MICH. L. REV. 837 (2006); Margaret Jane Radin, *Boilerplate Today: The Rise of Modularity and the Waning of Consent*, 104 MICH. L. REV. 1223 (2006); Todd D. Rakoff, *The Law and Sociology of Boilerplate*, 104 MICH. L. REV. 1235 (2006).

20. Smith, *supra* note 2.

21. *Id.* at 1176, 1179–80, 1196, 1197.

solutions to similar problems. Patterns also serve as heuristics for attorneys, i.e., devices to estimate quickly whether particular language solves certain bargaining problems, meets client objectives, and will be interpreted by courts in an anticipated manner.²²

Contract patterns, like Smith's modules, allow teams of lawyers to work on different aspects of a contract or transaction simultaneously. Multiple persons can work on the same agreement, Smith explains, because contract modularity restricts the information transfer of certain "boilerplate" provisions. This means that if one lawyer modifies one module of a contract, other modules can remain relatively unaffected.²³

Patterns also enable scalability or the transformation of contracts into transactions and transactions into markets.²⁴ Smith's work on modularity and restricting information transfer comes into play here too. To boil part of his theory down: standardized contract language means that third parties need to incur less cost in valuing either certain contracts or parties to those contracts.²⁵ We can expand Smith's logic to explain how contract patterns enable standardized contracts to be traded on organized financial markets. In extreme examples, contract patterns allow certain debt contracts to become what economist Gary Gorton calls "informationally insensitive" debt.²⁶ Certain financial instruments become informationally insensitive when they are (at least in theory) immune to adverse selection by traders with inside information. Investors can value informationally insensitive contracts at low cost. The ease of valuation, in turn, makes these tradeable contracts both highly liquid and endows them with many of the economic features of money.²⁷

The pattern language framework explains not only how sophisticated contracts function, but also how they fail. The pattern language framework provides a lens through which we can examine recent contracts law scholarship on the failures of sophisticated contract design, including "sticky" contract provisions in sovereign bond agreements,²⁸

22. See Part II.E generally. A view of pattern as heuristic has parallels to Kahan and Klausner's idea that boilerplate provisions in corporate law agreements enable and result from learning effects. See Kahan & Klausner, *supra* note 19, at 729–33.

23. See Smith, *supra* note 2, at 1207; see also *infra* notes 132–133 and accompanying text.

24. See *infra* Part II.D.

25. See Smith, *supra* note 2, at 1210.

26. See Gary Gorton, *Slapped in the Face by the Invisible Hand: Banking and the Panic of 2007*, 7, 9–10 (Nat'l Bur. of Econ. Res., Working Paper, 2009), available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1401882.

27. *Id.* at 9.

28. MITU GULATI & ROBERT E. SCOTT, *THE THREE AND A HALF MINUTE TRANSACTION: BOILERPLATE AND THE LIMITS OF CONTRACT DESIGN* 9–17 (2013) (framing question of "sticky")

“Frankenstein” contracts in mortgage-backed securitizations,²⁹ and the “flash crash” and other episodes of massive losses generated by automated, algorithmic trading on financial exchanges.³⁰ Just as modularity and contract design patterns foster the development of new financial instruments and markets, so too can their features contribute to the unraveling of these markets. For example, by restricting the information content of contracts, patterns and modularity not only midwived the creation of liquid markets for those contracts, they also played a role in the catastrophic freezing of these markets. This essay points to the “shadow bank runs” triggered when modularized and standardized asset-backed securities became almost impossible to value during the financial crisis.³¹ So failures of contract design can have broader social consequences far beyond the private relationship of the two parties to a bargain. More broadly, the failure of contracts can have systemic effects for entire markets when a particular contract enjoys widespread use or when it is so connected to other critical contracts that cascading failures occur.³²

Common threads run through these separate contract failures. These

sovereign bond contracts); *id.* at 33–44 (surveying previous literature explaining contract stickiness). Professors Kahan and Klausner provided an early economic analysis of why contract boilerplate provisions might become “sticky” even if they did not use that term in their work. Kahan & Klausner, *supra* note 19, at 727–36 (analyzing how “switching costs” associated with boilerplate terms and other dynamics may lead to persistence of suboptimal boilerplate).

Continued use of these “sticky” terms might serve the expressive, symbolic, or political needs of clients. See Anna Gelpern & Mitu Gulati, *Public Symbol in Private Contract: A Case Study*, 84 WASH. U. L. REV. 1627 (2006). However, these obsolete terms may also fool those clients into a false sense of security that contracts perform their stated functions, match the parties’ intent, or protect clients’ interests as advertised. See generally Charles J. Goetz & Robert E. Scott, *The Limits of Expanded Choice: An Analysis of the Interactions Between Express and Implied Contract*, 73 CALIF. L. REV. 261, 288–89 (1985) (analyzing consequences when boilerplate language becomes rote and its meaning unintelligible).

29. The financial crisis revealed a more destructive aspect of increasingly rigidified, modularized, and interconnected financial contracting. Anna Gelpern and Adam Levitin describe how the rigidity of certain contracts involved in a securitization prevented mortgage servicers from renegotiating mortgages with financially strapped borrowers. Anna Gelpern & Adam J. Levitin, *Rewriting Frankenstein Contracts: Workout Prohibitions in Residential Mortgage-Backed Securities*, 82 S. CAL. L. REV. 1075, 1124–27 (2009). This harmed not only the investors in securities based on those mortgages, but also had larger systemic consequences of deepening the “subprime” crisis. *Id.*

30. See Andrei A. Kirilenko & Andrew W. Lo., *Moore’s Law Versus Murphy’s Law: Algorithmic Trading and Its Discontents*, 27 J. ECON. PERSP. 51 (2013) (analyzing use of algorithms in automated trading and its role in both the flash crash and the 2012 trading error that caused losses in excess of \$400 million for Knight Capital); Frank Partnoy, *Don’t Blink: Snap Decisions and Securities Regulation*, 77 BROOK. L. REV. 151, 168–72 (2011) (examining 2010 flash crash).

31. See *infra* Part III.C.

32. See Anna Gelpern, *Financial Crisis Containment*, 41 CONN. L. REV. 1051, 1056 (2009) (discussing abrogation of financial contracts as historic mechanism to containing financial crises).

failures proliferated as contracting became increasingly rigidified,³³ routinized, modularized,³⁴ interconnected,³⁵ and even automated.³⁶ These same dynamics mean that Alexander's ideas in architecture have more purchase for contract design, given that he wrestled with analogous changes in contemporary architecture.³⁷

At first blush, Alexander's description of architectural patterns may appear to have little application to contracts. Indeed, contracts are not buildings, not least because contracts represent social rather than physical constructs (and for all the other reasons explained in Part II.G. below). However, Alexander's pattern language framework also illuminates how modern contract design has failed spectacularly. Alexander's pattern language framework recasts the problems created when interconnected contracts no longer work together, when inflexible contracts do not or cannot adapt to legal or economic shocks, and when automated contracts remove human judgment and generate both agreements and errors at light speed.³⁸ In addition, Alexander's rubric has much to say for the consumer end of the contracting spectrum. His program speaks to problems of consent, equity, and error as contracts of

33. For example, Anna Gelpern and Adam Levitin analyze how the rigidities built into certain contracts that govern a securitization contributed to the severity of the subprime crisis. See Gelpern & Levitin, *supra* note 29; see also *infra* Part III.B.

34. Cf. Smith, *supra* note 2, at 1176 (arguing boilerplate provisions create a modularity in contract design that allows lawyers and clients to manage complexity; boilerplate represents a middle point in the spectrum that runs from "information-rich contract rights limited to a particular deal to simple standardized property rights availing against 'the world.'").

35. See G. Mitu Gulati, William A. Klein & Eric M. Zolt, *Connected Contracts*, 47 UCLA L. REV. 887 (2000) (using metaphor of "connected contracts" to describe how and why firms are organized).

36. The project of automating contract drafting dates back decades. See Patricia Hassett, *Technology Time Capsule: What Does the Future Hold?*, 50 SYRACUSE L. REV. 1223, 1228 n.14 (2000) (providing examples of "legal expert systems" or computer programs designed to draft legal contracts and statutes). An expanding array of contracts is now being written in machine-readable form, so that computers can interpret and perform contract functions. See Harry Surden, *Computable Contracts*, 46 U.C. DAVIS L. REV. 629, 641–42, 647–48 (2012).

37. See Max Jacobson, *Max Jacobson Interviews Christopher Alexander*, 42 ARCH. DESIGN 768, 768 (1971) (quoting Alexander: "One of the most serious difficulties in the environment today is the machine-like character of the buildings that are being made. They are alienating and untouched by human hands."); ALEXANDER ET AL., *THE PRODUCTION OF HOUSES*, *supra* note 6, at 24 (critiquing modern "mass housing": "[p]laced and built anonymously, the houses express isolation, lack of relationship, and fail altogether to help create human bonds . . ."); *id.* at 39–40 (criticizing "efficiency" of current mass housing design for producing alienating buildings and for failing to involve residents in design process). See also ALEXANDER, *THE LINZ CAFÉ*, *supra* note 6, at 85–94 (describing Alexander's attack on Modernism in architecture).

38. See *infra* Part III.A.

adhesion migrate online³⁹ and to a mobile device world.⁴⁰ It also provides a framework for thinking about improving contract design as companies, such as LegalZoom, enable consumers to bypass transactional lawyers and draft and botch their own complex agreements.⁴¹

This essay also briefly considers how Alexander's normative program serves as a starting point for a longer discussion of ways to improve contract design. Alexander's collective work served not merely as an instruction manual—it also represented, in roughly equal measure, hymnal and manifesto.⁴² For Alexander and his collaborators, better design of the built environment served not merely to increase its functionality or aesthetics. *A Pattern Language* and Alexander's other work also aimed to make the design process more *democratic*, by providing lay people with a clear vocabulary, grammar, and syntax to create their own architectural plans.⁴³ Alexander also sought to make individual buildings, neighborhoods, cities, and regions more *harmonious* with one another and with the natural environment.⁴⁴ At the same time, his work underscored the need for a pattern language of design to be *adaptive* to changing social needs and environmental conditions.⁴⁵ Finally, *A Pattern Language* and Alexander's other writings present a thoroughly *humanistic* vision of architecture, in which design meets basic needs of individuals and families rather than those of abstract, mechanistic institutions or ideologies.⁴⁶

39. Cf. Margaret Jane Radin, *Online Standardization and the Integration of Text and Machine*, 70 *FORDHAM L. REV.* 1125, 1135 (2002) (discussing problems of online contracts of adhesion).

40. But cf. Scott R. Peppet, *Freedom of Contract in an Augmented Reality: The Case of Consumer Contracts*, 59 *UCLA L. REV.* 676 (2012) (arguing that technologies of smartphones can enable consumers to manage information asymmetries associated with consumer contracts, rendering many contract law concerns with adhesion contracts less salient).

41. LegalZoom and other companies have enjoyed considerable success in selling “do-it-yourself” legal agreements to consumers covering an array of complex transactions from divorce settlements to the formation of a limited liability company. See Lindzey Schindler, *Skirting the Ethical Line: The Quandary of Online Legal Forms*, 16 *CHAP. L. REV.* 185 (2012) (analyzing online services that provide form contracts under professional responsibility rules for lawyers).

42. For an analysis of the philosophical and ideological implications of Alexander's work, a critique of Alexander's reticence in confronting all of those ideological implications, and a description of the enemies that have challenged Alexander's work, see Kimberly Dovey, *The Pattern Language and Its Enemies*, 11 *DESIGN STUD.* 3 (1990).

43. See *infra* Part I.C.1. Note that Alexander's poetic style makes it difficult to boil his normative arguments down to a handful of points.

44. See *infra* Part I.C.2.

45. See *infra* Part I.C.3.

46. See *infra* Part I.C.4.

These normative elements and, moreover, the descriptive power of *A Pattern Language* provide guideposts for how contract design and drafting can move from teaching a legal skill, which is too often relegated to the outskirts of the legal academy, to studying and addressing complex, cascading, and catastrophic failures of contract design in modern markets, if not a more ambitious, normative program. In short, the increasingly automated, routinized, rigid, interconnected, complex, and opaque nature of modern contracts presents practical problems for meeting client needs and averting systemic contract failures. Moving from a descriptive plane, these failures also pose deeper philosophical challenges to values central to contract law, such as individual consent and equity. On a normative plane, rethinking contract design and drafting in terms of a pattern language may make contracts not only more intelligible, but more closely hew to the human needs and values of the individuals ultimately bound.

This short essay follows the following simple plan. Part I provides an overview of Christopher Alexander's writings on architectural design and explains how *A Pattern Language* created a vocabulary, syntax, and grammar for architects and planners. It also sketches how computer science borrowed pattern language to structure solutions to its own design problems. Part II describes how the design of legal contracts exhibits some of the same logic of a pattern language. It describes how contract patterns work to create contracts, transactions, liquid markets for financial instruments, and entire financial systems. If Part II describes how contract patterns function, then Part III examines how they break down, sometimes catastrophically. This final Part considers how the pattern language framework describes and illuminates recent contract failures and provides some normative guideposts for improving contract design.

I. PATTERN LANGUAGE IN ARCHITECTURE AND DESIGN

A. *Alexander's Approach*

Alexander and his collaborators created a process for creating better buildings and cities by starting with a series of design patterns that meet particular human needs or solve particular problems in a structure.⁴⁷ They isolated patterns that recur throughout rooms (e.g., "Entrance Room," "Zen View," or "Couple's Realm")⁴⁸ or structures (e.g.,

47. ALEXANDER ET AL., *supra* note 4, at x, xiii.

48. *Id.* at xxvii–xxviii.

“Cascade of Roofs,” “Arcades,” “Staircase as a Stage”).⁴⁹ Alexander and his co-authors assert that these patterns represent the fundamental design building blocks from which all cities and structures are formed: “at the larger scale of towns and buildings, the world is also made of certain fundamental ‘atoms’—that each place is made from a few hundred patterns—and that all of its incredible complexity comes, in the end, simply from the combination of these few patterns.”⁵⁰ Alexander believes that the best contract patterns evolved in traditional settings as the people who lived in buildings developed their own design solutions.⁵¹

Yet Alexander’s patterns represent more than isolated and abstract elements. Rather, each design pattern relates to other design patterns. Individual patterns may be composed from other smaller patterns.⁵² For example, Alexander describes how traditional stone houses found in the South of Italy are composed from the following patterns:

- Square Main Room
- Two Step Main Entrance
- Small Rooms off the Main Room
- Arch Between Rooms
- Main Conical Vault⁵³

In turn, Alexander’s patterns can be used to form larger patterns. Those Italian stone houses fit into a larger pattern of the town:

- Narrow Streets
- Street Branching
- Front Door Terrace
- Connected Buildings
- Public Wells at Intersections
- Steps in the Street⁵⁴

In short, Alexander creates a system for placing together individual patterns for a room to create patterns or solutions for the design problems of entire buildings. He sets rules for how mosaics of patterns

49. *Id.* at xxvi–xxvii.

50. ALEXANDER, *supra* note 5, at 99–100.

51. ALEXANDER, THE OREGON EXPERIMENT, *supra* note 6, at 38. Nikos Salingaros takes Alexander’s implicit evolutionary view of architecture quite a bit further and develops a Darwinian theory of how architectural design has developed. See NIKOS A. SALINGAROS, A THEORY OF ARCHITECTURE 195–210 (2006); see also *id.* at 21–22 (citing Alexander’s influence).

52. ALEXANDER ET AL., *supra* note 4, at xviii.

53. ALEXANDER, *supra* note 5, at 188.

54. *Id.* at 190–91.

for individual buildings together form patterns or solutions for neighborhoods (e.g., "Eccentric Nucleus,"⁵⁵ "Promenade,"⁵⁶ and "Housing Hill"⁵⁷). In turn, *A Pattern Language* describes rules for joining neighborhood patterns, (e.g., "Mosaic of Subcultures,"⁵⁸ "Local Transport Areas,"⁵⁹ and "Community of 7000"⁶⁰) to create design solutions for cities. Urban and rural patterns, when arranged together, create regions (e.g., "City Country Fingers,"⁶¹ "Agricultural Valleys,"⁶² and "Independent Regions"⁶³). This interlocking structure of patterns provides architects with an endless number of combinations of patterns to use in design. "A pattern language is a system which allows its users to create an infinite variety of those three dimensional combinations of patterns which we call buildings, gardens, towns."⁶⁴

The genius of Alexander's writing is that it creates rules for which patterns belong together in any particular context. Alexander describes when and why particular design patterns fit together and, by implication, when combinations of patterns make little sense:

There is a structure on the patterns, which describes how each pattern is itself a pattern of other smaller patterns. And there are also rules, embedded in the patterns, which describe the way that they can be created, and the way that they must be arranged with respect to other patterns.⁶⁵

Whether patterns make sense together is determined by whether they solve particular human needs and problems posed by the natural and built environment. Alexander explains this, somewhat elliptically, by arguing that architectural patterns work when they are "congruent" or when there is a "fundamental inner connection" between the pattern and the human events that occur within that pattern.⁶⁶ So a successful pattern for a Chinese kitchen stems from a "pattern of relationships required for

55. ALEXANDER ET AL., *supra* note 4, at 150-55.

56. *Id.* at 168-73.

57. *Id.* at 209-14.

58. *Id.* at 42-50.

59. *Id.* at 63-69.

60. *Id.* at 70-74.

61. *Id.* at 21-25.

62. *Id.* at 26-28.

63. *Id.* at 10-15.

64. ALEXANDER, *supra* note 5, at 186.

65. *Id.* at 185.

66. *Id.* at 92-93.

cooking Chinese food.”⁶⁷

B. Vocabulary, Syntax, and Grammar: Linguistics and Design

In creating this system for architectural design, Alexander borrowed heavily from linguistics and mathematics. Indeed, he analogizes his pattern language to logical languages, which contain two features:

1. A set of elements, or symbols.
2. A set of rules for combining these symbols.⁶⁸

Alexander explains that natural languages, such as English, represent more a complex variation of this simple logical language.⁶⁹ The two elements are again present. A natural language consists of words and then rules that outline the permissible arrangements of words.⁷⁰ However, a natural language also defines the relationships between words. Alexander writes, “there is . . . a structure on the words—the complex network of semantic connections, which defines each word in terms of other words, and shows how words are connected to other words.”⁷¹ Just as a natural language contains words and rules of grammar for creating sentences that make sense, so too an architectural pattern language contains both a vocabulary of patterns and syntax for combining those patterns to create buildings that make sense. Alexander finds direct analogues between the systems of natural languages and pattern languages:

<i>Natural Language</i>	<i>Pattern Language</i>
Words	Patterns
Rules of grammar and meaning which give connections [between words]	Patterns which specify connections between patterns
Sentences	Buildings and places ⁷²

The individual patterns in Alexander’s works serve as the equivalent to a “vocabulary” or collection of words in a language. Alexander’s

67. *Id.* at 94.
68. *Id.* at 183–84.
69. *Id.* at 184–85.
70. *Id.* at 184.
71. *Id.*
72. *Id.* at 187.

patterns describe particular design problems and solutions. Alexander's work also contains a "syntax" or system to link together the individual patterns or vocabulary into longer, more complex descriptions of design solutions (that is, buildings, neighborhoods, and cities).

Alexander argues that this pattern language has a "generative" quality.⁷³ It provides individuals with conceptual building blocks and rules to design an infinite number of buildings. Alexander writes, "[A] pattern language . . . gives us the power to generate these coherent arrangements of space. Thus, as in the case of natural languages, the pattern language is *generative*. It not only tells us the rules of arrangement, but shows us how to construct arrangements—as many as we want—which satisfy the rules."⁷⁴ Alexander's work thus builds directly off the linguistic theories of Noam Chomsky, who developed the idea that human languages possess a "generative grammar."⁷⁵

A Pattern Language deeply impressed computer scientists, who faced their own design problems. Programmers adapted Alexander's approach to create "patterns" in computer codes. As with architecture, computer programmers could break complex design problems into components and find a pattern solution for each component. Programmers could then combine each pattern to a particular design problem to create organized, modularized solutions to more intricate problems.⁷⁶ Alexander's writings exerted a particularly strong influence on the development of "object-oriented" programming.⁷⁷

73. *Id.* at 186.

74. *Id.* Alexander connects this generative quality of an architectural pattern language to a human language, "[B]oth ordinary languages and pattern languages are finite combinatorial systems which allow us to create an infinite variety of unique combinations, appropriate to different circumstances . . ." *Id.* at 187.

75. In linguistics, a generative grammar describes a system of rules that relate "signals" (such as words) to the semantic interpretations of those signals. See NOAM CHOMSKY, TOPICS IN THE THEORY OF GENERATIVE GRAMMAR 12 (1978); see also Finlay et al., *supra* note 14, at 163 (discussing connections linking Chomsky's work with Alexander's pattern language with computer languages).

76. JAMES O. COPLIN, SOFTWARE PATTERNS (1996) (outlining Alexander's influence on software programming); Lea, *supra* note 12, at 39.

77. Lea, *supra* note 12. Object-oriented programming is an approach to computer coding which decomposes complex problems into "objects." Object oriented programs encapsulate particular abstract concepts into discrete "objects," with each object formed by a data field. Objects also encapsulate certain procedures (or "methods") for manipulating data. A program functions by these objects or modules doing the problem-solving work, rather than employing a top-down hierarchy of algorithms. Object-oriented programming languages include C++ and Java. GRADY BOOCH ET AL., OBJECT-ORIENTED ANALYSIS AND DESIGN WITH APPLICATIONS (3d ed. 2007).

C. *Alexander's Normative Program*

Alexander's intellectual project extends far beyond merely describing how traditional architecture is made. Instead, he sets out to create both theory and instruction manual for improving architectural design and urban planning. He aims not only to improve the functionality (which is an objective more associated with architectural modernism)⁷⁸ or aesthetics of architecture. It is hard to pin down Alexander's normative program; after all, he famously made the search for a "quality without a name" the centerpiece of his theories.⁷⁹ Yet several normative themes reverberate throughout Alexander's work. He sought to make design more democratic, harmonious, adaptive, and humanistic. Each of these objectives deserves elaboration as they have implications for modern contract design.

1. *Democratic*: The dust jacket cover of *A Pattern Language* states that a primary goal of the book is to enable citizens outside the architecture profession to participate in the process of designing and improving the built environment. The jacket reads:

At the core of these books is the idea that people should design for themselves their own houses, streets, and communities. This idea may be radical (it implies a radical transformation of the architectural profession) but it comes from the observation that most of the wonderful places of the world were not made by architects but by the people.⁸⁰

Alexander and his collaborators write in the body of the book that they seek to give citizens the tools to develop their own pattern languages for designing buildings and cities.⁸¹

2. *Harmonious*: Alexander also sought to develop a means to build individual homes, buildings, neighborhoods, cities, and regions that exist in more harmony both with each another and with the natural

78. GORDON GRAHAM, *PHILOSOPHY OF THE ARTS: AN INTRODUCTION TO AESTHETICS* 174–75 (3d ed. 2005).

79. ALEXANDER, *supra* note 5, at 17.

80. ALEXANDER ET AL., *supra* note 4.

81. *Id.* at xvii. See also ALEXANDER, *supra* note 5, at 167. Alexander argued:

The people can shape buildings for themselves, and have done it for centuries, by using languages which I call pattern languages. A pattern language gives each person who uses it, the power to create an infinite variety of new and unique buildings, just as his ordinary language gives him the power to create an infinite variety of sentences.

Id. See also ALEXANDER ET AL., *THE OREGON EXPERIMENT*, *supra* note 6, at 5, 38–41 (describing "principle of participation" in which users of buildings must make decisions on architectural design); Dovey, *supra* note 42, at 4 ("Many of the patterns imply a kind of democratic, participatory socialism . . .").

environment. Alexander's quasi-mystical prose makes this quality hard to define concretely. He speaks of creating architectural patterns "free from internal contradictions."⁸² He defines architectural patterns as "fabrics of relationships"⁸³ that must be "congruent" with the human events that occur within those spaces.⁸⁴ Alexander sought to address the "inability [of design and engineering] to balance individual, group, societal, and ecological needs."⁸⁵

3. *Adaptive*: Alexander and his collaborators did not see their pattern language as fixed, but rather as fluid and evolving. They likened the pattern language to experiments or "hypotheses of science."⁸⁶ They called their catalogue of 253 patterns in *A Pattern Language* their "best guess as to what arrangement of the physical environment will work to solve the problem presented."⁸⁷ But all these patterns remain "free to evolve under the impact of new experience and observation."⁸⁸ One computer software scholar argues that Alexander aimed to address both: "[a]esthetic and functional failure in adapting to local physical and social environments" and designs that were "ill suited for use in any specific application."⁸⁹ Alexander himself remarked, "one of the characteristics of any good environment is that every part of it is extremely highly adapted to its particularities."⁹⁰

82. ALEXANDER, *supra* note 5, at 26.

83. *Id.* at 89.

84. *Id.* at 92-94. The following passages underscore how important harmony between the design of spaces and the lives lived in those spaces are to Alexander's vision. Alexander writes of the essential nexus between good design patterns and the human activities and events that happen within those patterns: "[T]here is a fundamental inner connection between each pattern of events, and the pattern of space in which it happens. . . . [E]ach pattern of relationships in space is congruent with some specific pattern of events." ALEXANDER, *supra* note 5, at 92-93. He then describes how advocates for patterns that are "alive," i.e. those that "let our inner forces loose," rather than those that are dead, which "keep us locked in inner conflict." *Id.* at 101. He elaborates:

[A] person is so far formed by his surroundings, that his state of harmony depends entirely on his harmony with his surroundings . . . in some towns, the pattern of relationships between workplaces and families helps us to come to life . . . In other towns where work and family life are physically separate, people are harassed by inner conflicts which they can't escape.

Id. at 106-08. See also ALEXANDER ET AL., THE OREGON EXPERIMENT, *supra* note 6, at 9-11 (discussing the principle of "organic order" that marks good architectural design and use of patterns); ALEXANDER ET AL., A NEW THEORY OF URBAN DESIGN, *supra* note 6, at 2-3, 22 (describing use of architectural patterns to promote "organicness" and to "heal the city").

85. Lea, *supra* note 12, at 39.

86. ALEXANDER ET AL., *supra* note 4, at xv.

87. *Id.*

88. *Id.*

89. Lea, *supra* note 12, at 39.

90. Christopher Alexander, *The Origins of Pattern Theory: the Future of the Theory, and the Generation of a Living World*, IEEE Software (Sept./Oct. 1999), at 71, 74.

4. *Humanistic*: Alexander's writing places human needs, experiences, and emotions at the center of the problems of design. Alexander and his collaborators indicated that the state of architecture as they found it was "fragmented" and "not based on human, or natural considerations."⁹¹ Good patterns, Alexander writes, help an individual achieve personal harmony, which "depends entirely on his harmony with his surroundings."⁹² Alexander places human beings and their needs at the center of architectural patterns.⁹³ He sought to remedy architecture's "[l]ack of purpose, order, and human scale."⁹⁴

II. HOW CONTRACT DESIGN WORKS: THE PATTERN LANGUAGE OF CONTRACTS

A. *Basic Patterns of a Contract*

Legal contracts also exhibit the traits of a pattern language. In designing and drafting agreements, lawyers also use a series of patterns. Each pattern solves a particular problem that the contracting parties face in establishing the terms of their relationship going forward. Often these different patterns correspond to different numbered parts or sections of the agreement. The following patterns appear in most legal agreements:

The preamble: This pattern names the parties to the transaction and establishes the date on which the agreement has been signed.⁹⁵

The exchange: This type of pattern usually appears as a section towards the beginning of the agreement. It establishes the exchange of promises or performance at the core of the contract.⁹⁶ For example, this pattern might cover the sale of an asset for a sum of money, the lease of particular property for future rent payments, or a loan repayable at a certain interest rate repayable in installments over a set term.

The conditions to an exchange: Another group of patterns sets the conditions to one or more party's obligations under the

91. ALEXANDER ET AL., *supra* note 4, at xvi.

92. ALEXANDER, *supra* note 5, at 106. Alexander notes that "stress and conflict are a normal and healthy part of human life," *id.* at 113, but that "a pattern which prevents us from resolving our conflicting forces, leaves us almost perpetually in a state of tension." *Id.* at 114.

93. *See supra* note 37.

94. Lea, *supra* note 12, at 39.

95. TINA L. STARK, DRAFTING CONTRACTS: HOW AND WHY LAWYERS DO WHAT THEY DO 51 (2007).

96. *Id.* at 95 (labeling this an agreement's "action sections").

contract. This pattern might take the form of closing conditions (if there is a time lag between the signing of the agreement and when obligations become effective), conditions precedent, or conditions subsequent.⁹⁷

Representations and warranties: In this family of patterns, the parties make statements of fact, such as those regarding their capacity to contract, financial health, or the quality of the assets involved in the bargain. Should a party's statements be untrue, the other party might have certain rights (specified elsewhere in the contract), such as to cancel the contract, not to perform its obligations, or to seek monetary damages or other remedies.⁹⁸ Indeed, as architectural patterns, contractual patterns must fit together coherently in accordance with rules of syntax.

Covenants: This type of pattern specifies other ongoing agreements the parties make ancillary to the basic exchange. These include agreements to perform certain obligations (positive covenants, such as obtaining insurance) and agreements to refrain from taking certain actions (negative covenants, such as not incurring any other indebtedness).⁹⁹

Default: Many contracts employ one pattern that defines when one or both parties have defaulted on its obligations. This may occur on a violation of the basic exchange, when a particular representation or warranty is untrue, or upon the breaking of a covenant.¹⁰⁰ This particular pattern then meshes with the next pattern.

Remedies: This pattern describes one party's recourse when the other party defaults under the agreement, which might include: ceasing to perform its obligations, terminating the contract, seeking monetary damages, or obtaining injunctive or non-monetary relief.

Termination: This pattern functions to define when the contractual relationship ends, what happens when it ends, and which obligations might continue past termination.

Other patterns, including "boilerplate": Contracts often employ other patterns, such as recitals that appear at the beginning of the

97. *Id.* at 133.

98. *Id.* at 113 (introducing representations and warranties); *id.* at 159–60 (listing misrepresentations among triggers for default provisions, rights of a party to terminate contract, and other remedies).

99. *Id.* at 125.

100. *Id.* at 159–60.

contract before the operative provisions and provide context for why the parties are entering into the agreement.¹⁰¹ A definitions section provides the meanings of terms used in multiple places in the agreement.¹⁰² Finally, the end of many contracts contain general (often “boilerplate”) provisions, such as specifying which jurisdiction’s law governs the contract, where and how disputes relating to the contract will be resolved, how the contract may be amended, and whether third parties have rights under the contract.¹⁰³

Many practitioner’s manuals and model agreements serve a similar function of *A Pattern Language*, namely to provide a guide to negotiating and drafting patterns and outlining a syntax for fitting the patterns together. These manuals specialize in particular types of transactions, such as mergers¹⁰⁴ or loan agreements¹⁰⁵ in business transactions.

B. Environmental and Legal Design Problems: Cunningham’s Contribution

Lawyers practice their craft by modifying provisions culled from previous agreements to meet their client’s particular objectives, as well as to match the terms of the negotiated deal. The overlapping and conflicting objectives of the parties represent the *environmental design problems* to which contract patterns must respond. Provisions borrowed from previous agreements into which one or both of the current parties had entered (contractual precedent) may provide both negotiating leverage and comfort that those patterns had satisfactorily addressed these environmental problems.

Lawyers earn their keep not merely by acting as scriveners, but by negotiating and drafting in the shadow of complex legal regimes. Particular bargains may not be enforceable or may trigger a host of legal consequences. Larry Cunningham’s *Contracts in the Real World* serves as a user friendly guide to some of these *legal design problems* created by the types of common law doctrines found in a first-year law school

101. *Id.* at 38, 60–62.

102. *Id.* at 73–78.

103. *Id.* at 167.

104. *See, e.g.*, JAMES C. FREUND, *ANATOMY OF A MERGER: STRATEGIES AND TECHNIQUES FOR NEGOTIATING CORPORATE ACQUISITIONS* (1975).

105. *See, e.g.*, LEE C. BUCHHEIT, *HOW TO NEGOTIATE EURO CURRENCY LOAN AGREEMENTS* (2d ed. 2000); ANTHONY C. GOOCH & LINDA B. KLEIN, *DOCUMENTATION FOR LOANS, ASSIGNMENTS AND PARTICIPATIONS* (1996).

contracts course. These include legal rules for when certain bargains are unenforceable,¹⁰⁶ the limits on particular contractual remedies,¹⁰⁷ and how courts will interpret contractual language.¹⁰⁸ Like Alexander with his photographs and drawings of cathedrals, town squares, and house floor plans, Cunningham illustrates these legal design problems with colorful examples from court cases, including staples of the first year Contracts course¹⁰⁹ and modern parables such as the contract travails of rap star Eminem.¹¹⁰ Cunningham's book could be read in conjunction with contract patterns that map onto the various legal design problems he identifies. For example, he outlines a typical parole evidence controversy and discusses how a merger clause can reduce the risk of a court looking outside of a written contract to prior or contemporaneous agreements.¹¹¹

Contract patterns operate to solve problems and exploit opportunities in more complex regulatory regimes. For example, tax lawyers often develop standardized and modularized financial instruments and transactions that can help clients lower their tax rate while achieving the same economic benefits. Lawyers engaged in tax planning thus also employ contract patterns.¹¹²

C. *Contract Patterns Compared to Boilerplate*

Contract patterns bear a strong resemblance to contract boilerplate. However, my definition of a contract pattern is more expansive than boilerplate. I define a contract pattern as an encapsulated solution captured in legally enforceable language (whether in a single provision in a single contract or in an entire series of contracts) to a particular legal problem. That legal problem might be the difficulty of matching the objectives of the two contracting parties in a discrete bargain (an environmental design problem). The problem might also be a legal design problem, as described above.

106. CUNNINGHAM, *supra* note 9, at 35–58.

107. *Id.* at 84–108.

108. *Id.* at 126–47.

109. *See, e.g., id.* at 63–64 (providing an account of *Sherwood v. Walker*, 33 N.W. 919 (Mich. 1887)).

110. CUNNINGHAM, *supra* note 9, at 126–29 (describing the controversy over licensing rights to rap music).

111. *Id.* at 132–34.

112. *See* Dan L. Burk & Brett H. McDonnell, *Patents, Tax Shelters, and the Firm*, 26 VA. TAX REV. 981, 995–97 (2007) (describing modularity of tax planning in context of tax investment patents).

A contract design pattern is conceptual. It might not take the exact same form—the rote incantation of the same words—in every contract. Boilerplate, by contrast, uses the same written formulation—the same sequence of words—again and again. Scholars generally use boilerplate to mean particular provisions, sections, or clauses in an agreement, but contract patterns describe much more than parts of legal agreements.

D. The Hierarchy of Contract Patterns: From Contracts to Transactions to Markets

A pattern language describes not only the way that individual provisions form legal contracts, but, moreover, the way in which lawyers arrange particular legal contracts together to create transaction structures. Consider a simple purchase of a residential home. Parties often effect this transaction with a purchase and sale agreement (governing the purchase of the property), a note (governing the financing of the purchase), a mortgage (relating to the security interest of the lender in the property), a deed of trust, and other ancillary agreements (such as insurance) and disclosures.¹¹³ These individual agreements must cohere, just as smaller architectural patterns for buildings must join to fit a larger building complex or neighborhood.

Individual agreements also form patterns in more complex, commercial transactions. The sale of a business might involve more than a basic agreement covering the purchase (such an asset purchase agreement or merger agreement).¹¹⁴ The transaction might also require organizational documents (article of incorporation or bylaws for a corporation).¹¹⁵ If the acquisition is being financed, the pattern for the acquisition agreements must mesh with the appropriate pattern for a loan financing (such as a credit agreement for a bank loan, an indenture for bonds, a security agreement for an interest in collateral, etc.) or an equity issuance (changes in a company's organizational documents, a stock

113. ALEX M. JOHNSON JR., UNDERSTANDING MODERN REAL ESTATE TRANSACTIONS (3d ed. 2012). See *id.* at 58 (describing purchase and sale agreement); *id.* at 119–21 (describing uses of note and mortgage); *id.* at 129 (explaining deed of trust); see also *id.* at 87–97 (describing other documentation for closing a real estate transaction); *id.* at 13 (describing broker listing agreement).

114. See FREUND, *supra* note 104, at 139, 147–61 (describing acquisition agreement and its component parts). Employment agreements are another example of documents critical to many acquisitions. *Id.* at 399–418.

115. See *id.* at 105–07, 109–11 (describing corporate law mechanics of a merger); *id.* at 78–79 (describing use of specially created subsidiaries to effect acquisition transactions); see also ROBERT CHARLES CLARK, CORPORATE LAW, §§ 10.3, 10.4, 418–37 (1986) (describing various merger structures, including the use of specially created subsidiaries).

subscription agreement, etc.),¹¹⁶ as well as any requisite disclosure to investors.¹¹⁷

Patterns for financing may join together to form more complex credit transactions. Consider a securitization, in which an originating lender sells pools of mortgages or other loans to trusts or other investment vehicles, which then issue asset-backed securities to investors. A securitization funnels the cash investors use to purchase their securities back to originating lenders, who can make fresh mortgages or other loans. When mortgage borrowers make payments on their loans, the cash streams funnel through the investment vehicles to make payments to the holders of asset-backed securities.¹¹⁸

Securitization thus joins the module or suite of residential mortgage agreements to the module or pattern of agreements involved in issuing bonds. Securitization includes a series of other patterned agreements, such as a pooling and servicing agreement. Under this agreement, a financial firm acts as agent for the trust (i.e., the investment vehicle) in collecting payments on the underlying mortgages or loans and enforcing the rights of the trust vis à vis the borrowers on those mortgages or loans. As we will see in a moment, the relationship between this particular agreement and the other patterns in a securitization began to break down during the financial crisis. This failure evidenced a problem in the pattern language and contract design of securitization.¹¹⁹

Securitization, in turn, forms one of the components of a larger web of financial instruments and markets that connects consumer and commercial borrowers to investors in capital markets and links different financial intermediaries to another. This web of instruments—asset-backed securities, asset-backed commercial paper, repurchase agreements (repos), money-market mutual funds, and credit derivatives—provides a network for providing credit and transferring

116. EDWIN L. MILLER JR., *MERGERS AND ACQUISITIONS: A STEP-BY-STEP LEGAL AND PRACTICAL GUIDE* 90–115 (2008) (describing issues and mechanics of different forms of financing for an acquisition, including equity and debt); 295–301 (describing legal issues and mechanics of leveraged buyouts); see also WILLIAM J. CARNEY, *MERGERS & ACQUISITIONS: THE ESSENTIALS* (2009) (explaining merger and acquisition transactions); 39–41 (discussing financing of transaction); 235–37 (describing leverage buyout transactions, including documentation and legal issues); 241–45 (describing securities law issues and documentation for acquisitions financed through equity or notes).

117. See FREUND, *supra* note 104, at 67 (describing disclosure documents required (at that time) for many acquisitions), 427 (describing proxy statement needed to obtain shareholder consent).

118. Gerding, *supra* note 17, at 147–51; Schwarcz, *supra* note 17, at 135–36.

119. See *infra* Part III.C.

credit risk.¹²⁰ Many scholars see the evolution and market freeze in this system as central to understanding the global financial crisis.¹²¹

For this essay, the insight is more basic: individual provisions form patterns for legal agreements, patterns of legal agreements arranged together form patterns for more complex transactions and financial instruments, and patterns of transactions and instruments are essential to the construction of larger financial systems and markets.

E. The Functions of Contract Patterns: How They Work

The modularity and standardization of contracting design patterns allow lawyers and clients to achieve economies of scale and reduce transaction costs for complex market transactions.¹²² Constructing contracts from modified, pre-formulated patterns lowers transaction costs in several ways. Most obviously, lawyers must spend less time drafting and negotiating.

Contract patterns potentially reduce transaction costs by serving several different kinds of functions. First, they may act as *heuristics* for legal analysis of contract terms under conditions of legal risk and uncertainty.¹²³ Patterns give lawyers comfort that particular provisions “work”—that is, they achieve the business objectives of clients, are internally consistent, and run a lower risk of unforeseen consequences under various legal regimes.¹²⁴ Patterns perform a checklist function for

120. See GERDING, LAW, BUBBLES, AND FINANCIAL REGULATION, *supra* note 18, at 397–418.

121. See Gorton, *supra* note 26.

122. See Stephen J. Choi & G. Mitu Gulati, *Innovation in Boilerplate Contracts: an Empirical Examination of Sovereign Bonds*, 53 EMORY L.J. 929, 936 (2004) (surveying literature that contract standardization is driven by high volume players looking to achieve economies of scale); Smith, *supra* note 2, at 1187–88 (describing how modularity in contracts allows contracting parties to save on transaction costs). This phenomenon meshes with the influential theory of the firm literature, in which entrepreneurs face a choice in deciding how to assemble products or services: they can either “make” (produce a given product or service within the hierarchy of a firm) or “buy” (buy the necessary input products or services in a market). See Ronald H. Coase, *The Nature of the Firm*, 4 ECONOMICA 386 (1937). All things equal, by reducing the transaction costs of purchasing products and services in a market, contract design patterns shift the calculus of entrepreneurs towards markets and away from expanding the size of their firms.

123. Labeling contract provisions as heuristics is not new. See, e.g., Clifford W. Smith & Jerold B. Warner, *On Financial Contracting*, 7 J. FIN. ECON. 117, 123 (1979). Gulati and Scott survey the literature on how “sticky” contract boilerplate may reflect satisficing. See GULATI & SCOTT, *supra* note 28, at 37–38. But see Jean Tirole, *Cognition and Incomplete Contracts*, 99 AM. ECON. REV. 265 (2009) (analyzing how contract drafters use heuristics to determine when not to address certain future contingencies and to leave contracts “incomplete”).

124. Kahan and Klausner note this “drafting efficiency” as one subset of “learning benefits” or “learning externalities” that accompany the use of boilerplate. Kahan & Klausner, *supra* note 19, at 720–21.

transaction lawyers with basic issue spotting. Patterns also may give lawyers greater comfort that courts will enforce the contract language and interpret it in an expected manner.¹²⁵

Second, patterns may also economize on the ex ante costs of interpreting contracts, whether by the parties, a court, or a third party.¹²⁶ Henry Smith analyzes how modular contracts restrict information flow.¹²⁷ The resultant move towards standardization means that interpreting a contract requires less time in analyzing the particular parties and the context of the transaction.¹²⁸

Third, contract patterns serve to economize on bargaining.¹²⁹ They may provide shortcut arguments during negotiation with other parties. In other words, they can serve as bargaining precedent (either for a particular counterparty or as an industry standard). Patterns also can help in “bargaining” with courts. Widespread use of contract patterns may have the collective effect of being perceived as “too big to fail”; in other words, lawyers may believe that a court could not invalidate a particular

125. See Kahan & Klausner, *supra* note 19, at 722.

126. Again, this insight is not new, but can be found in the germinal work of Professors Kahan and Klausner. They argue that use of boilerplate leads to another “learning benefit” or “learning externality” of greater certainty as to how courts or other lawyers, professionals and investors will interpret standardized contract terms. See Kahan & Klausner, *supra* note 19, at 720–22 (discussing how boilerplate may have been tested in judicial precedent); *id.* at 723–24 (discussing how boilerplate reduces the cost of providing advice or evaluating securities or investments). Kahan and Klausner then describe an additional series of positive “network externalities” that might accrue with the more of widespread use of particular boilerplate terms. Greater use of a term lowers the cost of legal and other advice as more professionals gain expertise with the term. *Id.* at 726. Network benefits also accrue to firms that employ a boilerplate term after a judicial opinion has upheld or interpreted that term favorably. *Id.*

127. See Smith, *supra* note 2, at 1187–88.

128. See *id.* Lawyers and law firms may thus enjoy an advantage as collectors and aggregators and tailors of patterns. They may have a comparative advantage in understanding how and when to apply patterns and to fit them together for particular situations. They may also have a better ability to predict how courts and regulators might interpret patterns. See generally Kahan & Klausner, *supra* note 19, at 736–39 (analyzing role of lawyers, securities underwriters and other “contracting agents” in promoting the diffusion of learning benefits with boilerplate terms, coordinating contract choices, and enabling cross-subsidization to promote learning and network externalities associated with particular boilerplate). Cf. Victor Fleischer, Regulatory Arbitrage, 89 TEX. L. REV. 227, 239–40 (2010) (describing crucial role of lawyers and law firms in creating and implementing regulatory arbitrage strategies for clients). At the same time, the organizational dynamics of law firms as “assembly lines” may explain the “stickiness” of contract provisions that make little sense. Barak Richman, *Contracts Meets Henry Ford*, 40 HOFSTRA L. REV. 77 (2011).

129. Cf. GULATI & SCOTT, *supra* note 28, at 34 (surveying literature on learning and network externalities associated with contract boilerplate); *id.* at 36 (arguing that a party who may not insist on idiosyncratic change to boilerplate for fear of sending negative signal about its future behavior); *id.* at 37–38 (surveying literature on how parties may resort to boilerplate because of satisficing in negotiations and otherwise).

contract provision because it would thus invalidate a huge swath of contracts in the marketplace.¹³⁰

Contract patterns thus offer many of the same benefits of “modularity” that Henry Smith associates with boilerplate language. Patterns aid in decomposing complex problems. They provide a middle ground between information-rich, completely bespoke contract terms and more standardized rules of property.¹³¹ Placing contract provisions in modules reduces the ripple effects of revising or committing an error in a particular provision.¹³² Contract patterns also allow the duties of drafting and negotiation of agreements to be broken up in space (for example, with different specialists in a law firm focusing on different aspects of a merger agreement) or in time (with different provisions being drafted in the past and then pulled off the shelf and re-assembled).¹³³

F. Trading Contracts

We can extend Smith’s theory of how modularity limits the information flow of boilerplate provisions to explain how contract patterns ultimately form patterns for financial markets. Smith argues that by standardizing language, boilerplate essentially strips out information from parts of contracts. This has the somewhat counterintuitive effect of making those provisions more valuable, because standardized language means that counterparties (or third parties) must invest less time to determine what a provision means. By extension, they need invest less time in evaluating the contracting party and how it might interpret or apply the standardized term.¹³⁴

130. Historical financial crises bear witness that this gamble does not always pay off. See Gelpert, *Financial Crisis Containment*, *supra* note 32, at 1056 (describing historical instances in which governments rewrote private contract provisions because of financial crisis conditions).

131. See Smith, *supra* note 2, at 1176.

132. *Id.* at 1188–91.

133. *Cf. id.* at 1180–85.

134. *Id.* See also Kahan & Klausner, *supra* note 19, at 723–24. Kahan and Klausner explain how boilerplate lowers the cost of valuation for investors:

[T]he use of a common term reduces the expense that investors and securities analysts incur in evaluating a firm’s securities and comparing them to alternative investments. This reduced cost increases the liquidity of a security, thereby reducing the issuer’s cost of capital. If a term is commonly used, the cost and effort entailed in understanding the term and its impact on value can be spread over many investments.

Id. When more firms employ a particular boilerplate term, they enjoy “network benefits.” *Id.* at 725–27. This stems from the greater “availability of a large number of investors and securities analysts who will learn how to price a firm’s securities at later public offerings and on the secondary market.” *Id.* at 726.

Smith's logic can be extended even further to explain how standardized contracts can become fungible enough for investors to trade on an exchange. Smith's work thus dovetails with the research of economist Gary Gorton. Gorton argues that certain financial instruments, such as senior asset-backed securities, can become "informationally insensitive."¹³⁵ This means that no trader can earn additional returns by trading the instruments based on inside, non-public information.¹³⁶ Conversely, other traders will no longer fear being at an information disadvantage in the marketplace.¹³⁷ Investors can easily price informationally insensitive instruments.¹³⁸ This promotes the formation of deep and liquid markets for these types of instruments with many buyers and sellers.¹³⁹ Information insensitivity and liquidity mean that these instruments begin to assume many of the economic features of "money."¹⁴⁰

The creation of liquid markets to trade certain contracts, such as asset-backed securities, stems in part from interlocking contract patterns. Asset-backed securities are formed by a web of dozens of contracts from indentures to thousands of mortgages. Investors need not expend enormous amounts of effort evaluating the particular provisions of each of these contracts or of the system as a whole because of the use of contract patterns. Investors can assume that certain mortgage patterns will mean certain things to other investors or, heaven forbid, in a court of law. Investors can assume that contract patterns of mortgages will fit together with patterns of other agreements, like indentures, to create easy-to-value asset-backed securities. Of course, the crisis revealed that investors make these assumptions until they do not (a topic discussed in Part III).

Note that this information stripping or hiding feature of contract patterns and contract modularity has a direct analogue in object-oriented computer programs, another branch in Alexander's intellectual genealogy.¹⁴¹ Object-oriented programs often hide key details of a code

135. Gorton, *supra* note 26, at 4–7.

136. *Id.* at 7.

137. *Id.*

138. *See id.*

139. *Id.*

140. *Id.* Instruments become more like money when they serve as a medium of exchange, a unit of account, and a store of value. *See* N. GREGORY MANKIW, *MACROECONOMICS* 75–77 (5th ed. 2003).

141. Lea, *supra* note 12, at 44–45.

from the user of information.¹⁴² When solving problems with the program, the user or programmer needs only to deal with a more abstract layer or module of the code, rather than delving into (and possibly debugging) all the details of the program.¹⁴³

For now, note how contract patterns, standardization and modularity had important benefits for the development of securitization markets. Investors purchasing asset-backed securities needed to invest fewer resources in acquiring information about pools of supposedly standardized mortgages.¹⁴⁴ This feature enabled investors to trade these asset-backed securities more easily, sell them to be re-securitized (creating new layers of asset-backed securities), or pledge them as collateral for extremely short-term loans such as repos.¹⁴⁵ In other words, contract patterns formed patterns for markets, which formed patterns for entirely new financial systems.

G. *The Limits of Metaphor: Contracts Are Not Buildings*

Of course, differences abound between contracts and a contractual pattern language and the built environment and an architectural pattern language. Where buildings stand or fall based on the natural forces of load, contracts are purely social constructs. Unlike buildings, contracts, by their nature, can never have a sole architect. Contracts arise only out of that so-called “meeting of the minds” of at least two parties. Often (but not always) this occurs after some negotiation between the parties.¹⁴⁶ Contracts only bind when individuals make decisions to comply with them or when courts enforce their terms. The shape of contracts—what they mean—comes from human interpretation. As those interpretations change, contracts change. The existence and effects of contracts may thus be contested and change over time.¹⁴⁷

Moreover, one of the central goals of Alexander’s architecture, to present a “morphologically and functionally complete” language, cannot be accomplished in contract design. Alexander describes this goal for an

142. MEILIR PAGE-JONES, FUNDAMENTALS OF OBJECT-ORIENTED DESIGN IN UML 12 (2000).

143. *Id.* at 12–14.

144. See Gorton, *supra* note 26, at 7, 9.

145. See *id.* at 14.

146. Some contracts, such as adhesion contracts, involve no negotiation between parties; one party drafts an agreement and presents it to another on a “take it or leave it” basis. See *infra* Part III.E.

147. For a famously provocative judicial opinion using linguistic scholarship to argue that the meanings of words and contracts are fluid and change over time, see *Pacific Gas & Electricity Co. v. G. W. Thomas Drayage & Rigging Co.*, 442 P.2d 641 (Cal. 1968).

architectural pattern language thus:

The language is a good one, capable of making something whole, when it is morphologically and functionally complete. It is morphologically complete, when the patterns together form a complete structure, filled out in all its details, with no gaps. And it is functionally complete when the system of patterns has that peculiar self-consistency in which the patterns, as a system, generate only those forces which they themselves resolve—so that the system as a whole, can live, without the action of self-destroying inner conflicts.¹⁴⁸

Contracts can never achieve these particular goals. A long literature in law and economics explains how contracts are fundamentally “incomplete”; that is, they cannot specify a rule for every future contingency that may arise affecting the relationship between the contracting parties.¹⁴⁹ Nonetheless, despite all these differences between the functions of, and forces acting upon, physical architecture and legal/social contracts, many of Alexander’s concepts map quite nicely from architectural design to contractual design.

III. WHEN CONTRACT PATTERNS FAIL

Not only does Alexander’s pattern language provide a description of how lawyers draft contracts, but both his description of patterns and his normative program also provide insights into how contract design has failed and how it might evolve to address these failures. Indeed, contract design faces many of the same technological and social forces that shaped twentieth century architecture. Contracting is becoming increasingly automated and rapid. Contracts are becoming increasingly complex and interconnected. At the same time, contract design has suffered from increasing rigidity and design failures, both small and systemic.

The following paragraphs sketch out some of the lessons of viewing contracts as pattern language for some of the contract design failures identified in recent contracts scholarship. Alexander’s description of the syntax and grammar of a pattern language provides a useful lens through which to view these failures. In addition, his normative goals—creating a more harmonious, humanistic, adaptive, and democratic design—

148. ALEXANDER, *supra* note 5, at 316.

149. For a definition of incompleteness in contracts, see Oliver Hart & John Moore, *Incomplete Contracts and Renegotiation*, 56 *ECONOMETRICA* 755 (1988). See also Oliver D. Hart, *Incomplete Contracts and the Theory of the Firm*, 4 *J.L. ECON. & ORG.* 119 (1988).

provide guideposts for how contract design and drafting might change. The aim of this final Part III is not to match Alexander's mystical style or his manifesto prose. Instead, it is to underscore the rich implications of his approach for modern contract design. My objective is limited. Rather than suggest sweeping reform of contract law doctrine, I seek only to propose a new lens for looking at contract design and its failures.

A. Increasing Automation and Flash Crashes

Contract drafting has become increasingly automated as many lawyers move from word processing to software programs for drafting complex legal agreements.¹⁵⁰ At the extreme, attorneys have begun writing contracts in computer-readable form, so that machines can interpret and follow legal agreements.¹⁵¹ As different stages of the contracting process—from drafting to interpretation and compliance—increasingly involve machines, the risks of severe mistakes multiply. This risk became most evident in the high-frequency algorithmic trading on financial markets that triggered the 2010 flash crash and the 2012 losses of Knight Capital.¹⁵² These losses occurred because of errors in the syntax of the contract patterns that financial firms use to make financial trades.¹⁵³ Even human trading can trigger stock market-crashes, but markets now face the risk of catastrophe due to small shocks triggering cascading failures of automated contracting. Removing humans from the critical stages of contracting creates the potential for catastrophic failure of contract design to adapt to changed environmental conditions. This meshes with the conclusions of economist Amar Bhidé, who argues that when financial conglomerates began to automate decisions on financial risk-taking, they removed the critical element of human judgment and set the stage for the global financial crisis.¹⁵⁴

Yet automation may trigger smaller scale failures as well. A software bug in an online contract market for airline tickets might cause significant losses for companies or travelers. Similarly, parties may not correctly tailor a complex contract written with the aid of contract drafting software to their individual circumstances.

150. Kevin E. Davis, *Contracts as Technology*, 88 N.Y.U. L. REV. 83, 117 (2013).

151. See Surden, *supra* note 36.

152. See Kirilenko & Lo, *supra* note 30.

153. Cf. Partnoy, *supra* note 30, at 171 (examining 2010 flash crash); Kirilenko & Lo, *supra* note 30, at 64 (analyzing use of algorithms in automated trading and its role in flash crash and 2012 trading error that caused losses in excess of \$400 million for Knight Capital).

154. AMAR BHIDE, A CALL FOR JUDGMENT: SENSIBLE FINANCE FOR A DYNAMIC ECONOMY (2010).

B. *Frankenstein Contracts*

The syntactical errors of algorithmic contracting have been more quickly resolved than the failures of other contract patterns in financial markets. The global financial crisis revealed a modern danger that routinized, rigid, and modularized complex financial contracts cannot easily adapt to economic and legal shocks and may thus deepen systemic financial crises. Anna Gelpern and Adam Levitin describe how rigidity built into the terms of pooling and servicing agreements for mortgage-backed securities prevented mortgage servicers from agreeing to restructure mortgage loans.¹⁵⁵ Restructuring might have lowered the defaults of mortgage borrowers and the ultimate losses to investors in the affected mortgage-backed securities. Conversely, a lack of flexibility in these contracts helped cement a collective failure of financial institutions to address underwater mortgages, which deepened the systemic financial crisis.¹⁵⁶

Gelpern and Levitin outline several different kinds of rigidity built into these pooling and servicing agreements. One was contractual: the agreements limited the discretion of the servicers to modify loans to mitigate opportunism by the servicers at the expense of investors.¹⁵⁷ Another rigidity was functional: the multiple investors in these bonds could not overcome the collective-action problems to waive the necessary contract provisions. These collective action problems became even more severe because many of the bonds were re-securitized and held by other investment trusts with multiple beneficial owners.¹⁵⁸ As with the flash crash, mechanistic contracting created overly rigid contract patterns that could not adapt to changing environmental conditions with catastrophic consequences.

C. *Modularity, Information Loss, and Bank Runs*

The modularity and interconnectedness of asset-backed securities also had other consequences for the financial crisis. The information stripping described in Part II.F above that enabled the development of

155. See Gelpern & Levitin, *supra* note 29, at 1087–89.

156. *Id.* at 1124–27.

157. *Id.* at 1091–93 (describing how interplay between statutory and contractual provisions in these agreements circumscribed ability of servicers to modify mortgages). Gelpern and Levitin also describe how these contractual limitations on the discretion of servicers reflected a structural rigidity: the need to comply with legal rules to make the mortgages “bankruptcy remote” from the potential insolvency of the original mortgage lenders. *Id.* at 1093–98.

158. *Id.* at 1098–1102.

securitization and shadow banking markets also contributed to their collapse. When the mortgages underlying asset-backed securities began defaulting in waves, the consequences of this information loss from modularizing contracts became manifest. Investors in mortgage-backed securities could not easily discern whether their particular instruments were affected by mortgage defaults. This problem was compounded for investors in second and third layers of asset-backed securities that were based on those mortgage-backed securities. Similarly, lenders who extended credit based on asset-backed collateral could no longer evaluate their credit risk adequately. Consequently, the information loss and valuation problems caused markets in asset-backed securities, repos, and other instruments to freeze. Complex financial markets suffered shadow banking runs.¹⁵⁹ The modularity and information loss created by rigid contract patterns both fathered and ultimately froze these markets.

Scholars have only begun to worry about a larger class of “systemic contracts,” the widespread use of which might cause financial markets to buckle during an economic shock.¹⁶⁰ Widespread use of a particular contract pattern that cause massive losses among numerous financial firms increases systemic risk by exposing firms to common shocks.¹⁶¹ This provides an extreme example of excessive uniformity in contract terms.¹⁶²

D. “Sticky” Contracts and Boilerplate Language

This failure of contract patterns to adapt occurs even when highly sophisticated attorneys in high-stakes transactions have the ability to make appropriate changes in response to legal shocks. For example, a number of scholars have documented how lawyers failed to modify key provisions in sovereign bond indentures to reflect seismic shifts in the

159. See Gorton, *supra* note 26. This same information loss triggered runs on other instruments in the shadow banking system. See GERDING, BUBBLES, FINANCIAL REGULATION, AND LAW, *supra* note 18, at 452–54; see also Gary Gorton & Andrew Metrick, *Securitized Banking and the Run on Repo*, 104 J. FIN. ECON. 425 (2012).

160. See John H. Cochrane, *Lessons from the Financial Crisis*, REGULATION, 34, 36–37 (Winter 2009–2010) (discussing “systemic contracts” that when widely offered by financial institutions increase systemic risk); see also Gelpert, *Financial Crisis Containment*, *supra* note 32.

161. Systemic risk has been defined as “the risk . . . of breakdowns in an entire system, as opposed to breakdowns in individual parts or components.” George G. Kaufman & Kenneth E. Scott, *What is Systemic Risk, and Do Bank Regulators Retard or Contribute to It*, 7 INDEP. REV. 371, 371 (2003). One way systemic risk increases is when multiple financial firms have exposure to common economic shocks, which would cause them to fail simultaneously. *Id.*

162. See Kahan & Klausner, *supra* note 19, at 734 (discussing how boilerplate may lead to negative externality of excessive uniformity in contract terms).

applicable legal rules.¹⁶³ “Sticky” contract terms represent an example of a larger puzzle of routinized contract terms in sophisticated transactions that appear to be unenforceable, incomprehensible, or illogical.¹⁶⁴ Although continued use of these terms might serve the expressive, symbolic, or political needs of clients,¹⁶⁵ they may also fool those clients into a false sense of security that contracts perform stated functions or protect their interests as advertised.¹⁶⁶

E. Contracts of Adhesion in a Digital and Mobile World

As consumer contracting increasingly migrates to online click-through contracts and to mobile devices, the classic concerns surrounding adhesion contracts¹⁶⁷ become both magnified and less visible to individuals. Lack of consumer understanding and consent and unequal bargaining power all become more pronounced in an electronic environment in which the pace of contracting accelerates, business can alter contract terms rapidly, and the ability and propensity of individuals to read and process those terms on a screen diminishes. This set of problems has led to a rich literature in privacy law scholarship, as well as numerous high-tech solutions (such as the use of avatars on websites and various augmented reality mechanisms) to alert individuals to what rights they are contracting away.¹⁶⁸

It would be a mistake to dismiss these sci-fi solutions to contracts of adhesion. They highlight a very real need to consider the human element in policies with respect to consumer contracting; patterns in consumer contracts must be carefully designed for individuals to be able to process their import and effectively consent to their terms. Hi-tech devices and designs may be able to make contract patterns more intelligible and more important contractual terms appear more salient.

163. *E.g.*, GULATI & SCOTT, *supra* note 28.

164. *See generally* GULATI & SCOTT, *supra* note 28, at 33–44.

165. *See also* Gelpert & Gulati, *Public Symbol in Private Contract*, *supra* note 28, at 1711–14. This vision of what a contract does builds off the earlier work of Mark Suchman. *See* Mark C. Suchman, *The Contract as Social Artifact*, 37 L. & SOC'Y REV. 91 (2003).

166. *See* Goetz & Scott, *supra* note 28. Scholars have long argued that boilerplate can lead to excessive use of suboptimal terms. *See* Michael Klausner, *Corporations, Corporate Law, and Networks of Contracts*, 81 VA. L. REV. 757 (1995); Kahan & Klausner, *supra* note 19, at 734.

167. The decade-old problems of consumer contracts of adhesion were revisited and recast in Todd D. Rakoff, *Contracts of Adhesion: An Essay in Reconstruction*, 96 HARV. L. REV. 1173 (1983).

168. *See e.g.*, Peppet, *supra* note 40; M. Ryan Calo, *Against Notice Skepticism in Privacy (and Elsewhere)*, 87 NOTRE DAME L. REV. 1027 (2012).

F. Democratization

The user interface of contract design has become increasingly important even outside contracts of adhesion. Hand-in-glove with increasing automation comes the increasing democratization of the contract drafting process. Consumers can now download wills, trusts, organizational documents for corporations and other business entities, residential leases, marital settlements, and other agreements online from providers such as LegalZoom.¹⁶⁹

While removing lawyers from the process may make contract drafting more affordable and democratic, this trend also brings significant risks for individuals. Individuals may lack the expertise and awareness of what the provisions in form contracts mean, how they should be adapted to individual situations, and how they might become inappropriate, overly restrictive, or obsolete given changed circumstances.

CONCLUSION

Seeing contracting and contract design in terms of a pattern language offers insights into what contracts and their drafters do. The pattern language framework also sheds new light on recent contract failures. From the flash crash to Frankenstein contracts to online privacy agreements, contract patterns have become overly rigid and routinized. Automation, interconnectedness, and contractual complexity have created a disjointed syntax. This means contract patterns no longer perform their stated roles, parties can no longer effectively consent to contract terms, and parties cannot adapt their contractual relationships to economic and social shocks, big or small.

I leave for later work important questions such as the appropriate roles of legislatures, regulatory agencies, and courts in addressing broken contract patterns, whether policymakers should require or prohibit particular contract patterns or mandate disclosures to consumers or investors regarding those patterns, or whether courts should adopt particular interpretative rules for contract patterns.¹⁷⁰ The task this article undertook was more modest: to sketch out how and why the pattern

169. See Davis, *supra* note 150, at 117.

170. Scholars have begun asking these questions with respect to contract boilerplate. See, e.g., Boardman, *supra* note 19 (analyzing interpretative rule that courts apply to boilerplate in insurance contracts); Hillman, *supra* note 19 (analyzing whether mandating disclosure with respect to boilerplate in online contracts might have perverse effects); Rakoff, *supra* note 19 (questioning embedded assumptions in boilerplate scholarship that courts and regulators should have limited roles).

language of contract works and to identify when this pattern language fails. A more ambitious normative program for contract patterns remains for another day. For now, one message remains: improving contract design—to become more adaptive, democratic, harmonious, and humanistic—is too important to view as mere skills training.