
Computable Contracts

Harry Surden*

This Article explains how and why firms are representing certain contractual obligations as computer data. The reason is so that computers can read and process the substantive aspects of contractual obligations. The representation of contractual obligations in data instead of (or in addition to) the traditional written language form — what this Article calls “data-oriented contracting” — allows for the application of advanced computer processing abilities to substantive contractual obligations. Certain financial contracts exemplify this model. Equity option contracts are routinely represented not as contract documents written in ordinary language — but as data records intended to be processed by computers. The parties incorporate such data as an expression of their substantive contractual memorialization through various processes.

The representation of contractual obligations as data allows for new contracting properties. Among these possibilities is the design of “computable” contract terms. This Article explains how parties can effectively “translate” certain contractual criteria into a comparable set of computer-processable rules. Parties can provide computer systems with existing data that is indicative or relevant to compliance or performance. In this way, certain previously manual comparisons between promised terms and actual party activities can be automated. This can have the effect of significantly reducing transaction costs associated with contract

* Copyright © 2012 Harry Surden, Associate Professor of Law, University of Colorado Law School; Fellow, Stanford Center for Computers and the Law, Stanford Law School; B.A. Cornell University; J.D. Stanford University. Many thanks to the University of Colorado Law School for supporting this work. I am also grateful to Paul Ohm, Deb Cantrell, Sarah Krakoff, Ahmed White, Andrew Schwartz, Phil Weiser, Vic Fleischer, Erik Gerding, and the rest of my colleagues at the University of Colorado for their helpful comments. Also, many thanks to Seema Shah, Andrew Coan, Mary-Anne Williams, Benjamin Johnston, and Michael Genesereth for their insightful comments. Also, thanks to Ashley Boothby, whose hard work and excellent research assistance improved this Article. This Article is dedicated to the memory of Kristen Vannoy who provided invaluable help. Her intelligence, generosity, warmth, and kindness were always appreciated and will be missed.

monitoring and compliance as compared to the traditional written-language contracting paradigm.

TABLE OF CONTENTS

INTRODUCTION	631
I. DATA-ORIENTED CONTRACTING	639
A. <i>Introduction to Data-Oriented Contracts</i>	639
1. What Is a Data-Oriented Contract and How Does It Differ from a Traditional Contract?	639
B. <i>Limits of Computer Processing of Language-Based Contracts</i>	642
1. Limitations in Natural Language Processing	642
2. Statistical Approaches to Computer-Based Analysis of Natural Language	644
C. <i>Data-Oriented Contract Expression — Structured Data</i>	646
1. Data-Oriented Contracting: How Parties Can Formulate Contracts as Data for Computers	647
D. <i>Endowing Data with Legal Substance</i>	651
1. Endowing Data with Shared Meaning.....	651
a. <i>Acquiring Shared Meaning by Adhering to Published Computer Standards</i>	652
2. Electronic Contracting Interfaces	654
3. Incorporating Data as the Expression of the Contract: Threshold Procedural Agreements	655
E. <i>Contract Doctrine and Data-Oriented Contracts</i>	656
F. <i>The Importance of Data-Oriented Expression</i>	657
II. COMPUTABLE CONTRACT TERMS AND CONDITIONS.....	658
A. <i>What It Means To Be Computable</i>	659
1. Example 1: Certain Financial Contract Terms	659
2. Example 2: Geographic Limitations on Online IP	661
3. Example 3: Stanford Intellectual Property Exchange (SIPX)	662
B. <i>Principles of Computable Contracts</i>	663
1. Computer Semantics: Conveying the Meaning of Concepts to Computers.....	665
2. Captured Legal Assertions and Advanced Semantics.....	667
3. Providing a Computer with a Source of Information Relevant to Compliance or Performance.....	671
III. THEORY AND COMPUTABLE CONTRACTING	674

2012]	<i>Computable Contracts</i>	631
	A. <i>Computable Contracting Through the Lens of Contracting Private Lawmaking</i>	675
	1. Contracting as Private Lawmaking	675
	a. <i>Choices in the Form in Which Contractual Obligations Are Expressed</i>	676
	b. <i>Choices in the Criteria, Terms, and Conditions by Which Compliance Is Measured</i>	677
	c. <i>Choices in How Contracting Parties Manage Uncertainty</i>	678
	B. <i>Limitations of Computable Contract Terms</i>	682
	1. Features Common to Computable Contracting Contexts	682
	2. Limitations of Computable Contracting	683
	C. <i>Producing Intelligent Results Without Intelligence</i>	685
	1. Functional View of Intelligent Results.....	686
	IV. THE IMPLICATIONS OF COMPUTABLE CONTRACTING	688
	A. <i>Perceived Benefits of Computable Contracting</i>	688
	1. Reduced Transaction Costs in Contracting	689
	2. New Analytical Properties Gained from Computability.....	690
	3. Permits Autonomous “Computer to Computer” Contracting.....	694
	B. <i>Theory: Transaction Costs, Contracts, and Property</i>	696
	CONCLUSION.....	699

INTRODUCTION

Commercial contracts involve promises under specified terms and conditions. Assessing compliance with these criteria can entail transaction costs.¹ For example, firms may incur costs in monitoring that each side has performed as promised or in orienting their own behavior to accord with terms.² Aggregated across multiple parties,

¹ Jesper Andersen et al., *Compositional Specification of Commercial Contracts*, 8 INT’L J. ON SOFTWARE TOOLS FOR TECH. TRANSFER 485, 485-86 (2006) (noting that contracts require numerous steps, including specifying the contracts, monitoring their performance, analyzing the impact of contracts on prices and supplies, and integrating this information into other operational units such as the supply chain).

² In practice, contracting parties do not necessarily monitor, assess, and enforce every contractual provision to the full extent specified. Rather, they may focus on a certain subset of core terms within a larger contract or under-enforce some formal terms or conditions altogether. For a discussion of the distinction between formal, written but under-enforced contractual provisions, see Lucian A. Bebchuk & Richard A. Posner, *One-Sided Contracts in Competitive Consumer Markets*, 104 MICH. L. REV. 827, 828 (2006) (“The distinction between contracts on paper and their actual

agreements, and transactions, the net costs of conforming to a network of contractual obligations can be significant.³ To manage problems of analogous cost in other contexts, firms have often employed technological solutions.⁴ Is the assessment of contract compliance similarly amenable to automation? Or is there something unique about contracting (and assessing legal obligations generally) that makes automated assessment infeasible?

The conventional view has been that the automation of contract monitoring or compliance is beyond the capability of contemporary technology.⁵ To understand this view, it is helpful to consider contract assessment as consisting of two broad phases: 1) the *understanding* of what has been promised under what conditions; and 2) the *comparison* of what was promised contractually to what has (or has not) happened.⁶

The first perceived barrier relates to computer-based “understanding” of contracts. Firms often memorialize contractual

implementation is one that has received much attention . . .”).

³ ANUJ SAXENA, ENTERPRISE CONTRACT MANAGEMENT: A PRACTICAL GUIDE TO SUCCESSFULLY IMPLEMENTING AN ECM SOLUTION 4 (2008) (“Today, it is not uncommon . . . for a large organization [to have] tens of thousands of legal agreements driving its operations, a situation several times more complex than in the past.”).

⁴ F. ROBERT JACOBS ET AL., MANUFACTURING PLANNING AND CONTROL FOR SUPPLY CHAIN MANAGEMENT 6 (6th ed. 2011); *see, e.g.*, JOEL D. WISNER, KEAH-CHOON TAN & G. KEONG LEONG, PRINCIPLES OF SUPPLY CHAIN MANAGEMENT: A BALANCED APPROACH 333 (2008) (describing the use of technology to solve business logistics problems).

⁵ ENRICO FRANCESCONI, SIMONETTA MONTEMAGNI & WIM PETERS, SEMANTIC PROCESSING OF LEGAL TEXTS: WHERE THE LANGUAGE OF LAW MEETS THE LAW OF LANGUAGE 60-62 (2010) (noting the specific capabilities of automated processing of natural language legal texts); Symposium, *Legal Reasoning and Artificial Intelligence: How Computers Think Like Lawyers*, 8 U. CHI. L. SCH. ROUNDTABLE 1, 19 (2001) (Cass Sunstein argues that, “[A]t the present state of the art artificial intelligence cannot engage in analogical reasoning or legal reasoning.”).

⁶ This somewhat reductionist view of a “contract” and the monitoring of contractual obligations is developed for analytical purposes but should not be taken as a positive description of the complexity of contractual arrangements. For example, the entirety of a “contract” often consists of many legal obligations layered onto those expressed in the writing — some of which are entirely external to the contract writing. *See, e.g.*, RICHARD A. LORD, 1 WILLISTON ON CONTRACTS § 1:1 (4th ed. 2012) (describing a “contract” as “the total legal obligation that results from the parties’ agreement . . . as supplemented by any other applicable laws”). Others have emphasized the distinctions between obligations framed in “formal” contract writings and actual obligations as monitored or enforced. *See, e.g.*, Gillian K. Hadfield, *Problematic Relations: Franchising and the Law of Incomplete Contracts*, 42 STAN. L. REV. 927, 992 (1990) (“[C]ourts tend to view the formal written contract as representing the entirety of the commitments structuring the franchise relationship.”).

arrangements in written documents.⁷ In these texts, the parties communicate what it is they are agreeing to through the medium of language.⁸ The perceived impediment to automation concerns the state of computerized understanding of language-based expression. When people read contract documents they engage high-level language processing abilities to understand the (often complex) contractual obligations.⁹ By contrast, contemporary computer algorithms cannot read or understand even basic written language texts at anywhere near the sophistication exhibited by a person of ordinary literacy.¹⁰

A distinct issue concerns contract performance. Parties often draft contracts with terms deliberately specified at varying levels of discretion, open-endedness, or abstraction to allow flexibility given future uncertainty.¹¹ Computer-based assessment in such scenarios appears problematic. People are able to respond reasonably in contexts involving judgment, abstraction, or uncertainty by employing

⁷ Contracts can be oral but are frequently memorialized on a written agreement in the commercial setting. E. ALLAN FARNSWORTH, FARNSWORTH ON CONTRACTS § 7.2, 191 (1990) (“The parties to a contract often reduce to writing part or all of their agreement, following negotiations during which they have given assurances, made promises, and reached understandings.”).

⁸ Written language is the typical form in which legal obligations generally, and not just contractual obligations, are expressed. See, e.g., Lawrence M. Solan, *Why Laws Work Pretty Well, but Not Great: Words and Rules in Legal Interpretation*, 26 LAW & SOC. INQUIRY 243, 244 (2001) (“[L]aws are expressed in language, and we necessarily use words and rules whenever we use language, whether for legal or for other purposes.”).

⁹ See, e.g., Argye E. Hillis & Alfonso Caramazza, *The Reading Process and Its Disorders*, in COGNITIVE NEUROPSYCHOLOGY IN CLINICAL PRACTICE 229 (David Ira Margolin ed., 1992) (“[A] cognitive process such as reading involves a series of transformations of mental representations On this view, even very simple cognitive tasks will involve various processing mechanisms”)

¹⁰ The study of algorithms permitting computers to understand human language is known as natural language processing (NLP). For detailed explanations of the limits of NLP as of the writing of this Article, see STUART RUSSELL & PETER NORVIG, ARTIFICIAL INTELLIGENCE: A MODERN APPROACH 860-67 (3d ed. 2010); Robert Dale, *Classical Approaches to Natural Language Processing*, in HANDBOOK OF NATURAL LANGUAGE PROCESSING 1, 1-7 (Nitin Indurkha & Frederick J. Damerau eds., 2d ed. 2010); Richard Socher et al., *Semantic Compositionality through Recursive Matrix-Vector Spaces*, in CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING (forthcoming 2012) (noting that particular NLP approaches are limited and “do not capture . . . the important quality of natural language that allows speakers to determine the meaning of a longer expression based on the meanings of its words and the rules used to combine them”).

¹¹ Relatedly, performance itself can occur in contexts of considerable uncertainty, unexpected facts, or unforeseen but desirable exceptions.

sophisticated cognitive processes.¹² By contrast, while contemporary computer systems can perform feats of apparent analytical sophistication in certain scenarios, as of yet, they are unable to act in cognitively demanding contexts at levels anywhere comparable to, for example, trained attorneys. These two technical obstacles — the limitations of current technology in reading language-based contracts and in assessing compliance with abstract, unstated, or flexible criteria — would seem to bar automation in determining contract conformance.

It is possible, however, to represent contractual obligations in forms other than ordinary language.¹³ In particular, parties can express certain contractual terms or conditions as *computer data*. Why express contract criteria in data rather than words? When terms are represented in highly-structured data, computers can process them with a high degree of accuracy as compared to those expressed in ordinary language. This “contract-as-data” approach is not simply theoretical, but can be seen in practice in domains such as finance.¹⁴ In recent years, firms are likely to express the core terms of certain financial contracts — such as agreements to sell currencies at a future date — not as written words on paper, but as computer-readable data records.¹⁵ Financial firms express obligations in this non-traditional form so that they may be easily processed by electronic financial trading systems.¹⁶ When contracting parties express terms as data, to facilitate computer analytics, we might describe such expression as “data-oriented.”

Expressing contracts as data may enable computers to *read* core terms, but what about assessing conformance? Consider a currency

¹² HUBERT L. DREYFUS, WHAT COMPUTERS STILL CAN'T DO 1-5 (1992).

¹³ For example, under basic contract principles, contracts can be expressed by non-language-based conduct. See *Hercules, Inc. v. United States*, 516 U.S. 417, 424 (1996) (internal citation omitted) (“An agreement implied in fact . . . although not embodied in an express contract, is inferred, as a fact, from conduct of the parties”)

¹⁴ See, e.g., DAVID F. DEROSA, OPTIONS ON FOREIGN EXCHANGE 19-20 (2011); FIN. STABILITY BD., IMPLEMENTING OTC DERIVATIVES MARKET REFORMS 39-45 (2010).

¹⁵ These kinds of financial contracts are an example of foreign currency derivative contracts — a family of agreements widely used to manage financial risk. See DON M. CHANCE & ROBERT BROOKS, INTRODUCTION TO DERIVATIVES AND RISK MANAGEMENT 2 (2009); see also Tim Cave, *Over-the-Counter Derivatives Join Electronic Revolution*, FINANCIAL NEWS (May 3, 2010), <http://www.efinancialnews.com/story/2010-05-03/over-the-counter-derivatives-join-electronic-revolution> (noting that in 2008, the members of the International Swaps and Derivatives Association electronically traded over 80% of over-the-counter derivatives contracts).

¹⁶ AYESHA KHANNA, STRAIGHT THROUGH PROCESSING FOR FINANCIAL SERVICES: THE COMPLETE GUIDE § 4.31 (2008).

contract's typical performance obligation — to pay an amount of money on a specified date.¹⁷ In the traditional contracting paradigm, a reasonable means of assessing *prima-facie* performance would be to compare the records of payment transfers from promisor to promisee, to the terms of the deal.¹⁸ If such records were available *electronically*, monitoring the contract could be both more automated and more efficient. The parties might provide data relevant to performance — such as electronic payment records — to their computers. When both the contractual obligation and the information relevant to fulfilling that obligation are in computer-processable form, a computer may compare what was promised to what has occurred as a *prima-facie* indicia of conformance. When systems are designed to produce automated assessments of conformance, we may describe those contract terms as being *prima-facie* “computable.”¹⁹

This Article introduces the theory and methods of data-oriented and computable contracting.²⁰ A “data-oriented” contract is one in which

¹⁷ For simplicity in illustration, this example involves settlement by delivering the underlying currency. However, most derivatives contracts do not actually involve the exchange of the underlying asset (e.g., the foreign currency) during settlement, but rather, involve a cash payment netting the value of the contract based upon the underlying value given the relevant exchange rate. DEUTSCHE BORSE GRP., THE GLOBAL DERIVATIVES MARKET - AN INTRODUCTION 15 (2008), available at http://www.math.nyu.edu/faculty/avellane/global_derivatives_market.pdf.

¹⁸ Indeed, we might reasonably assume that this is the type of evidence that a party would introduce in a legal proceeding to support compliance. See, e.g., *Harvard Drug Grp., L.L.C. v. Senior Respiratory Solutions, Inc.*, No. 09-13083, 2010 WL 148670, at *9 (E.D. Mich. Jan. 13, 2010) (“Defendants were indebted to Harvard Drug for \$151,432.40 Attached [are] nine invoices [including] the due date, the invoice amount, payments made on the invoice, and the balance due on the account.”); KHANNA, *supra* note 16, § 2.4.1.1.3 (“FedWire is used to take money out of the buyer’s account and to put it in the seller’s account.”).

¹⁹ Colloquially, the term “computable” is used when a computer can be given the means to produce a desired result (such as a mathematical computation). Analogously, we can consider the above contract terms to be computable in the sense that a computer received the necessary means to produce a *prima-facie* result concerning compliance. This Article employs a colloquial usage of the term “computable.” This Article does not employ the more formal usage as related to mathematical theory of computability. For more information about this mathematical concept, see MICHAEL SIPSER, INTRODUCTION TO THE THEORY OF COMPUTATION (1st ed. 1996).

²⁰ This work builds heavily on Professor Michael Genesereth’s concepts of Computer Science. See, e.g., Nathaniel Love & Michael Genesereth, *Computational Law*, in PROCEEDINGS OF THE TENTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW 205 (Association for Computing Machinery, Inc. 2005) (discussing an approach to automated legal reasoning focusing on laws, regulations, contract terms, and rules in the context of electronically-mediated actions).

the parties have expressed some part of their contractual arrangement as computer-processable data (as in the currency contract described above).²¹ When a contract term is “computable,” the parties have arranged for a computer to make automated, *prima-facie* assessments about compliance or performance (i.e., as in the comparison of payment terms to payment data). A central theme will be that some — but not all — contractual terms or conditions can be meaningfully represented in terms of data and rules for the purpose of automated assessment. Indeed, these approaches may apply to a relatively small subset of contracting subjects. This Article details those limits. However, in the appropriate context — as in the domain of finance — computer-understandable contracts can significantly reduce particular transaction costs.²² Such widespread reductions can alter laws developed under assumptions about prevailing transaction cost levels.

Part I of this Article explains the concept of “data-oriented” contract expression. This view of a “contract,” as represented as data rather than as (or in addition to) words, deviates significantly from the traditional paradigm in which contractual intentions are communicated using descriptive language.²³ Part I argues that data-oriented contracting has arisen as a way around current limitations in computer-based processing of language (the form in which contracts have historically been expressed). When parties want to apply the analytical capabilities of computers to the substance of their contractual obligations, they must reorient the way in which they express their contractual obligations. That is, they must move from written language toward structured computer data, which is more amenable to computer processing. Part I explains how parties can (and do) endow computer data with shared meaning and the legally substantive significance to effectuate contractual agreements.

²¹ For early recognition that legal obligations could be made computable, see William McGeveran, *Programmed Privacy Promises: P3p and Web Privacy Law*, 76 N.Y.U. L. REV. 1812, 1812-15 (2001).

²² In particular, computer-understandable contracts can reduce those costs associated with monitoring or assessing certain contract terms. See KHANNA, *supra* note 16, at 82 (“Trading derivatives [was] usually phone based, with manual faxing of contracts and consequent input of trade information into computer screens by the middle office. This method of communication created . . . and in general, made the trade lifecycle more prone to errors and delays in settlement.”).

²³ As this Article will later detail, it may not be possible to create computable versions of many contractual arrangements using contemporary technology. See *infra* Part III. In other contexts, it is technologically possible but inefficient to do so given the costs. *Id.* The key point is that, notwithstanding these limitations, there is a subset of arrangements that can be represented in data and is amenable to automation. *Id.*

The expression of contract terms as *data* is significant because it enables a suite of novel, computer-based, contracting abilities.²⁴ Part II explains one of these capabilities — computable contracting. As suggested above, parties can make certain contract terms “understandable” to a computer by translating the meaning of the term into a set of consonant computer instructions (e.g., “payment” translates into comparing payment records between parties). In some cases, parties may also enable the computer to “compute” conformance with such terms. They can do this by providing the computer with access to the type of data that is relevant to determining compliance (e.g., access to actual, electronic payment records). When such a link is created, computer systems may make automated, *prima-facie* (legally tentative)²⁵ compliance assessments by comparing contract terms to data indicative of conformance. In some cases, these systems automate straightforward comparisons that were previously done manually. In other cases, the computers can perform advanced analytics infeasible in the written-language-based contracting approach.

Part III explores the limits of data-oriented and computable contracting. These approaches are not suited for contracting scenarios involving significant amounts of uncertainty, abstraction, or complexity. Rather, they are geared toward the subset of contracting in which the application of contract terms is expected to be relatively non-controversial in the ordinary case. Such routine contracting contexts, although perhaps not arresting from a legal analytical perspective, represent domains of substantial commercial significance.²⁶ Part III theorizes the computable contracting concept

²⁴ For early recognition that legal obligations could be made computable, see McGeveran, *supra* note 21, at 1812-15. For descriptions of a variant of this idea in the digital rights management context, see also Stefan Bechtold, *Digital Rights Management in the United States and Europe*, 52 AM. J. COMP. L. 323, 323-82 (2004) (examining this concept in a digital rights management context).

²⁵ The phrase “*prima-facie*” is used to denote that automated assessments will often be “first cut” approximations of an ultimate, legally authoritative determination as to compliance. In litigation, *prima-facie* compliance with a law or contractual obligation is often influential regarding ultimate determinations of liability or legal consequence, but not necessarily determinative. A deviation might occur, for example, if a judge decides that there is a valid exception to a *prima-facie* result that should be considered in the ultimate determination of liability. For example, even if a party can successfully establish the basic elements of the *prima-facie* case, there are defenses (e.g., that the contract violates public policy) that can still defeat the *prima-facie* case and make the contract unenforceable. See RESTATEMENT (SECOND) OF CONTRACTS § 178 (1981).

²⁶ This is indicated by the electronic, standardized financial contracts described

through the lens of choice in contract design. Parties can deliberately architect an increased level of legal determinacy into a contracting arrangement so as to enable computability.

Part V explains the implications of computable and data-oriented contracting, including the ability to perform sophisticated computational analytics unavailable in the traditional written approach. For example, once contractual obligations are represented in a data-oriented manner, a computer system may be able to compare contracting data to manufacturing information to detect and prevent disparities or contradictions, potentially reducing legal risk.²⁷

There are also theoretical implications. Laws often contain implicit assumptions about transaction cost levels. The scope of such laws are thus linked to how costly certain activities are to carry out.²⁸ Computable contracting can reduce certain contracting transaction costs as compared to the traditional paradigm. The technological reduction of transaction cost levels can therefore have effects on the substance of laws. Part V explores how reductions in transaction costs can impact legal scope, using copyright's "fair use" doctrine as an example.²⁹ To the extent that computable contracting (or similar technologies) become more prevalent and alter transaction cost levels, the effective scope of certain laws may change, even as the statutes or doctrine, on their face, appear to remain constant.

earlier. See, e.g., Morten Bech, *FX Volume During the Financial Crisis and Now*, BIS Q. REV., Mar. 2012, at 33, 38 (2012) (noting the daily activity in 2011 on several foreign exchange electronic trading platforms consists of over \$100 billion per day apiece; also noting that this data actually represented only a small percentage of estimated the overall electronic trading daily volume in foreign currencies).

²⁷ For example, consider a scenario in which a firm has contracted to deliver products to two customers on the same date, but the firm has the manufacturing capacity to deliver to only one. In the traditional contracting context, such a disparity between a firm's legal obligations and actual capacity might be difficult to detect.

²⁸ I have written about this dynamic elsewhere as it relates to privacy, and it is applicable to the domain of contracting as well. See generally Harry Surden, *Structural Rights in Privacy*, 60 SMU L. REV. 1605 (2007) [hereinafter *Structural Rights*] (discussing the latent structural costs that help regulate social conduct).

²⁹ One justification for copyright's fair use doctrine is based upon market failures that are assumed to occur due to the high costs of contracting when low-value uses are at stake. See Wendy J. Gordon, *Fair Use as Market Failure: A Structural and Economic Analysis of the Betamax Case and Its Predecessors*, 82 COLUM. L. REV. 1600, 1618 (1982).

I. DATA-ORIENTED CONTRACTING

A. Introduction to Data-Oriented Contracts

A “data-oriented” contract is one in which the parties have expressed one or more terms or conditions of their agreement in a manner designed to be processable by a computer system.³⁰ Typically, the parties express core elements as precisely defined computer data, rather than (or in addition to) a written language document to facilitate computer analysis, automation, or communication of their contractual obligations. Data-oriented contract expression is becoming increasingly common in domains such as financial trading and electronic commerce — in which transacting occurs through electronic interfaces.³¹

More generally, we can understand data-oriented contracting as an approach by which contracting parties can reliably convey information about the *substance* of certain contractual legal obligations to computer systems. This Part explains how contract terms can be expressed as data, and why one would want to do so. In short, contemporary computers are unable to understand contracts written in ordinary language. Parties wishing to apply computer analytics to the substance of their contractual obligations must instead express them as data.

1. What Is a Data-Oriented Contract and How Does It Differ from a Traditional Contract?

It is helpful to highlight some differences between the data-oriented and traditional contracting paradigms. First, data-oriented contract

³⁰ This Article is introducing the term ‘data-oriented’ contract. The term was created for lack of an existing word to describe the concept of expressing contract data using computer-processable, structured data rather than natural language sentences or phrases. Conceptually, data-oriented contracts have existed in some form or another since the 1970’s with firms employing Electronic Data Interchange (EDI) systems to communicate electronically with one another. However, EDI systems are broader and cover a great deal of non-contracting electronic communication between firms. They also tend to be limited to particular types of contracts such as purchase orders. *See, generally*, JANE K. WINN & BENJAMIN WRIGHT, *THE LAW OF ELECTRONIC COMMERCE* § 5-59 (2001) (explaining EDI).

³¹ *See* MARTIJN GROOT, *MANAGING FINANCIAL INFORMATION IN THE TRADE LIFECYCLE: A CONCISE ATLAS OF FINANCIAL INSTRUMENTS AND PROCESSES* 67 (2008); MARTIN CHRISTOPHER SEXTON, *MDDL AND THE QUEST FOR A MARKET DATA STANDARD: EXPLANATION, RATIONALE AND IMPLEMENTATION* 168 (2007) (explaining that over the counter (or firm-to-firm) currency future contracts are frequently expressed in the FpML or similar data-language).

terms differ in the form by which they are expressed. As noted above, in a conventional contract, parties express their contracts using words — written (or spoken) descriptive language.³² Often parties memorialize contracts in a document in which the core aspects of the agreement are described in sentences or clauses.³³ By contrast, in a data-oriented contract, parties express some part of their contract — for example, key terms or conditions — as computer data and rules.³⁴ Thus, in the financial domain, certain contracts are today more likely to have core terms memorialized as structured computer data records rather than as written language agreements.³⁵

To say that a contract is “data-oriented” is not to suggest that every aspect of an arrangement has been represented in computer-processable form. Contractual relationships can consist of a nexus of multiple, sometimes interdependent, agreements. Moreover, contract documents can be complex and, beyond core terms, can cover topics ranging from distributing risk to provisions regarding choice of law.³⁶ The data-oriented label simply suggests that the parties have decided that *some* subset of key terms or conditions would benefit from being represented as computer processable data. Thus, one can consider

³² As the next part will discuss, contracts need not be expressed through language but can also be implied from conduct or action. *See* Part I.B.1.

³³ Technically, the contract can be considered to be larger than the document itself, which is the “writing” or the “formal” contract memorialization, because legal obligations can arise from outside of the four corners of the document itself. Nonetheless, the convention is to refer to the document informally as the “contract” since many, if not most, of the primary obligations arise from the formal terms stated in the writing. *See* LORD, *supra* note 6, § 1:1 (describing the contract as “the total legal obligation that results from the parties’ agreement . . . as supplemented by any other applicable laws”).

³⁴ As this Part will explain, these computer records are given semantic content through some deliberate methods.

³⁵ Thus, in some cases, there is both a traditional language-based contract document and a data-oriented “translation” of those terms; in other cases, the only representation of the contractual obligation is in data-oriented form. Thus, the contract documents exist largely as computer data records — and not as written language documents — in the databases of the financial firms and electronic trading platforms. *See, e.g.*, DAVE CLIFF, DAN BROWN & PHILIP TRELEAVEN, TECHNOLOGY TRENDS IN THE FINANCIAL MARKETS: A 2020 VISION 4, 8, available at <http://www.bis.gov.uk/assets/foresight/docs/computer-trading/11-1222-dr3-technology-trends-in-financial-markets.pdf> (discussing the global financial market’s aggressive adoption of new technologies, including the switch to paperless electronic trading).

³⁶ *See, e.g.*, Robert A. Hillman & Jeffrey J. Rachlinski, *Standard-Form Contracting in the Electronic Age*, 77 N.Y.U. L. REV. 429 (2002) (noting that boilerplate language in contracts often contains language about remote risks that are unlikely to materialize in an given contract).

financial contracts to be data-oriented, if the core terms of a given transaction — such as price, quantity, and payment date — have been represented as computer processable data, even if other topics from a larger contractual framework have not.³⁷

A second difference concerns the intended interpretive audience. Traditional contracts are intended to be read by people — for example, the contracting parties and those who might later adjudicate a dispute, such as a judge. Data-oriented contracts also ultimately need to be understandable by these parties, but they have an additional interpretive “audience” — computer systems. Thus, firms represent standard financial contracts in a highly structured, data-oriented form precisely to facilitate the computer-based formation, trading, and settlement of such agreements.³⁸ A final distinction — data-oriented contracts differ in the way in which they acquire substantive meaning. In a traditional contract, the meaning arises from the shared understanding of the language used by the parties.³⁹ In a data-oriented contract, the parties employ explicit processes — discussed below — to endow data with substantive meaning.⁴⁰

These distinctions highlight an important conceptual difference between data-oriented contracts and the traditional “electronic contracting” literature. That body of scholarship has often explored how contract doctrines designed for a paradigm of paper contracts translate to contexts where such agreements are expressed electronically (e.g., “browse-wrap” web-site agreements or contracting via e-mail.)⁴¹ Data-oriented contracts are different because, not only are they electronic in form, but also they have been purposely oriented

³⁷ Often, the totality of contractual obligations can be considered a hybrid of data-oriented and traditional, language-based expression. Financial electronic trading often involves a series of written “threshold” agreements that support the subsequent electronic-only contracting. See *infra* Part II.D.

³⁸ As this Article discusses later, in some instances, financial firms choose to represent their contract terms in data, not explicitly, but implicitly by a business decision to use a third party electronic trading platform that stores such contract data in electronic form. See CLIFF, BROWN & TRELEAVEN, *supra* note 35, at 10-11.

³⁹ See *Bergholm v. Peoria Life Ins. Co.*, 284 U.S. 489, 492 (1932) (noting that contracts “must be construed according to the terms which the parties have used, to be taken and understood, in the absence of ambiguity, in their plain, ordinary, and popular sense”); *Fla. Cent. R.R. Co. v. Schutte*, 103 U.S. 118, 140 (1880) (“[I]n this, as in other cases of contracts, language is to be given, if possible, its usual and ordinary meaning.”).

⁴⁰ As this Part will explain, these computer records are given semantic content through some deliberate methods.

⁴¹ See Hillman & Rachlinski, *supra* note 36, at 463-64.

for computer-based understandability.⁴² By contrast, email, browser-wrap, and other similar contracts, although “electronic” in form, are intended to be read and understood primarily by people, not computers, and are expressed in ordinary language.⁴³ In most cases this means that *computers* will not be able to understand what such language-based contracts mean. Thus, data-oriented contracts raise distinct issues from contracts that are merely electronic versions of those that were, in the past typically, written on paper.⁴⁴

To fully comprehend data-oriented contracting and why parties are engaging in it, it is helpful to have a basic understanding of limitations in computer-based processing of written language. In short, contemporary technology cannot easily process written language — the form in which contracts (including most “electronic contracts,” such as website terms of service) have traditionally been expressed. To the extent that contracting parties want computers to process the substance of their obligations, they must reorient their contractual expression away from ordinary language and toward highly-structured data — a form more amenable to computer processing.

B. Limits of Computer Processing of Language-Based Contracts

1. Limitations in Natural Language Processing

Computer scientists consider legal documents to be “natural language” texts. In computer science, the term “natural language” is used to refer to the ordinary languages that people use to

⁴² See, e.g., *About the Licenses*, CREATIVE COMMONS, <http://creativecommons.org/licenses/> (last visited Nov 2, 2012) (“In order to make it easy for the Web to know when a work is available under a Creative Commons license, we provide a “machine readable” version of the license — a summary of the key freedoms and obligations written into a format that software systems, search engines, and other kinds of technology can understand.”); *Machine-readable Privacy Policy Statements*, UNITED STATES DEPARTMENT OF COMMERCE, http://osec.doc.gov/webresources/policies/machine_readable_privcy_policy_statements.html (last visited Nov. 2, 2012) (“The standard for machine-readable Privacy Policy . . . enables Web sites to translate their privacy practices into a standardized format . . . that can be . . . automatically interpreted by a user’s [web] browser . . .”).

⁴³ For a comparatively rare work of scholarship that is focused on computer-understandable agreements, see McGeeveran, *supra* note 21, at 1812-13 (describing how it is possible to make website privacy agreements computer understandable).

⁴⁴ Digital Rights Management (DRM) can be understood to be a basic version of the data-oriented approach. While sophisticated rights-expressions languages have been developed, in practice, the level of granularity of expression of DRM terms has been fairly low level as of yet. For a good description of DRM and its applications, see Bechtold, *supra* note 24, at 323-25.

communicate — such as English or French.⁴⁵ The qualifier “natural” is meant to contrast against the highly structured and mathematically based “formal languages” such as those used to program computers.⁴⁶ Thus, written-language texts such as books, law review articles, letters, e-mails, or nearly any document that people use to communicate with one another would be considered examples of natural language expression. By contrast, a computer program would be considered an expression of formal language because it is written in a programming language with a highly constrained, structured, and pre-defined form.

Contract documents are thus natural language texts since they are written in ordinary sentences — rather than in the constrained, precisely-defined, mathematically based forms.⁴⁷ Natural Language Processing (NLP) is the branch of research devoted to enabling computers to be able to read and understand natural language expressions — such as books, e-mails, newspaper articles, or legal texts such as contracts or legislation — that people use to communicate with one another.⁴⁸

This natural/formal dichotomy is useful because contemporary computer systems are comparatively limited in their ability to “understand” natural language documents.⁴⁹ Contracting parties, lawyers, or other readers of documents (including legal texts such as contracts) rely upon high-order cognitive linguistic processing abilities that permit the understanding of complex, novel, abstract, and relatively unstructured natural language sentences.⁵⁰ By contrast, even the most advanced computer-based processing of natural language texts is comparatively limited, and (as of this writing) does not approach the reading and comprehension abilities of an ordinary literate person.⁵¹ To understand why parties sometimes express their contracts as data, it is helpful to have a basic understanding of the

⁴⁵ See RUSSELL & NORVIG, *supra* note 10, at 860-61.

⁴⁶ See *id.* (noting that sometimes, the contrast is made against “artificial languages”).

⁴⁷ One might not think of legal language as “natural” in the colloquial sense, given its highly specialized terminology. However, within a framework in which “natural” largely means “non-mathematical,” we can consider even arcane legal phrases to be “natural” in that they are based upon the conventions of person-to-person language.

⁴⁸ RUSSELL & NORVIG, *supra* note 10, at 860-61.

⁴⁹ FRANCESCONI, MONTEMAGNI & PETERS, *supra* note 5, at 60-62.

⁵⁰ See DAVID CAPLAN, NEUROLINGUISTICS AND LINGUISTIC APHASIOLOGY: AN INTRODUCTION 1-6 (1987); CHRISTOPHER D. MANNING & HINRICH SCHÜTZE, FOUNDATIONS OF STATISTICAL NATURAL LANGUAGE PROCESSING 3-5 (1999).

⁵¹ See MANNING & SCHÜTZE, *supra* note 50, at 3.

limitations of the most successful contemporary techniques for processing natural language.

2. Statistical Approaches to Computer-Based Analysis of Natural Language

Many useful techniques for analyzing natural language documents are based upon statistics.⁵² Within the legal context, parties have successfully used these statistical approaches to automatically assess natural language legal documents. In litigation discovery, for example, parties have used computers to filter out corporate documents likely to be irrelevant to a case, based upon what is written in the document.⁵³ Why not apply these same statistical techniques to read contracts written in English for the purpose of automating assessment with that contract's terms? This part will briefly discuss why the statistical approaches — although useful for the limited task of sorting documents during discovery — are currently inadequate for the more demanding task of understanding of the meaning natural language-based contracts.

Although parties have used automated techniques to assess the words of documents during discovery, it is in a comparatively limited role. In the typical case, parties use computers to winnow large troves of corporate documents into more manageable subsets for subsequent attorney review.⁵⁴ For example, during discovery in a contract case, one party might need to locate within a large trove of corporate documents (e-mails, reports, memorandums, contracts), the small subset that are that corporation's contracts.⁵⁵ Often the computers detect telltale patterns in the language of documents and then use these patterns to flag relevant documents based upon probability indicators. For instance, a computer might detect that documents that contain the phrase "the parties hereby agree" are much more likely to be *contracts* than some other type of document. Using such heuristic patterns a computer might sort contracts from other documents — at

⁵² See *id.* at 4.

⁵³ See, e.g., Joe Palazzolo, *Why Hire a Lawyer? Computers are Cheaper*, WALL ST. J., June 18, 2012, at B1, available at <http://online.wsj.com/article/SB10001424052702303379204577472633591769336.html> (describing how parties are beginning to use computers to automate the sorting of large troves of corporate documents during litigation).

⁵⁴ See *id.*; John Markoff, *Armies of Expensive Lawyers, Replaced by Cheaper Software*, N.Y. TIMES, Mar. 5, 2011, at A1, available at <http://www.nytimes.com/2011/03/05/science/05legal.html> (last visited Apr. 4, 2011).

⁵⁵ See Markoff, *supra* note 54, at A1.

a level of accuracy (say 80%) that is useful to reduce the document trove that requires manual attorney review.⁵⁶

The *reading* of contract language to assess conformance with contract terms is in many ways a much more demanding task. The statistical approaches described employ probabilistic models that estimate meaning based upon patterns and heuristics. Although the computers do examine the language of these documents, the computers do not engage with the underlying meaning of the words at a deep cognitive level, as would a literate human reader.⁵⁷ The somewhat counter-intuitive insight is that a deep conceptual understanding of the meaning of words is not required for a variety of useful document automation tasks; for those, statistical approximations work surprisingly well. However, for the task of reading contracts to determine conformance, the level of accuracy, depth, and sophistication demanded is greater.

Consider a basic contract-reading task that would be trivial for a sophisticated person, but potentially confounding for a computer employing a statistical algorithm: identifying the names of the parties who are contracting with one another. One issue is that natural languages, such as English, are comparatively less constrained than the formal languages used to program computers.⁵⁸ This wide range of expressive variation in natural language can confound sophisticated algorithms, but can be readily understood by literate human readers engaging cognitive processes.⁵⁹ Consider how, in natural language documents, it is often possible to convey more or less the same idea using one of many linguistic variations. A literate person would be able to recognize the conceptual equivalence of sentences such as “John Smith lives at 415 Broadway Street in Boulder Colorado” and “The resident of house number 415 found on Broadway, a street located in the city of Boulder, in the state of Colorado, is John Smith.”⁶⁰ Such readers use contextual clues and other processes to

⁵⁶ See Thomas Barnett et al., *Machine Learning Classification for Document Review* 1, 7 (June 8, 2009) (unpublished manuscript) (on file with the DESI III Global E-Discovery/E-Disclosure Workshop), available at http://www.law.pitt.edu/DESI3_Workshop/Papers/DESI_III.Xerox_Barnett.Xerox.pdf.

⁵⁷ See e.g., RUSSELL & NORVIG, *supra* note 10, at 860-65 (describing how probability models can approximate natural language to help transform it to formal language).

⁵⁸ See BARBARA H. PARTEE, ALICE TER MEULEN & ROBERT E. WALL, *MATHEMATICAL METHODS IN LINGUISTICS* 93 (1990) (“Natural languages are . . . suited to just about any communicative goal we may have.”).

⁵⁹ See Hillis & Carmazza, *supra* note 9, at 229.

⁶⁰ See Günter Neumann & Gertjan van Noord, *Reversibility and Self-Monitoring in*

understand linguistic formulations that they have never previously encountered.⁶¹

By contrast, since many automated statistical approaches rely upon patterns, such techniques might not be able to recognize the conceptual equivalence between similar sentences at anywhere near the 100% accuracy required for basic contracting tasks like identifying party names.⁶² Contextual signals of meaning or seemingly minor changes in order or subject — that would be trivial for a literate person to understand — can challenge even advanced language processing algorithms.⁶³ If basic contractual tasks — such as simply identifying party names given variations — are challenging to automated algorithms, even more advanced tasks — such as understanding substantive meaning or managing textual, legal, or factual uncertainty — likely pose even greater challenges.

In sum, while computer-based statistical approaches have proven useful in analyzing natural language documents in limited legal contexts (an example being litigation discovery) they are currently insufficient for the demanding task of reading and understanding traditional English-based contracts.⁶⁴ Contemporary computer technology cannot reliably read ordinary language-based expression at comparable levels of conceptual sophistication as those of literate readers.⁶⁵ However, this is not to suggest that complex information — such as those contained within contract documents — cannot be communicated to computer systems. Rather, the point is that contractual information has to be expressed to the computer in a form other than the natural language, English sentences in which they have traditionally been composed.

C. *Data-Oriented Contract Expression — Structured Data*

We have seen that many technological approaches cannot handle the relatively freeform nature of natural language, in which commercial contracts are often expressed.⁶⁶ A conventional view has

Natural Language Generation, in REVERSIBLE GRAMMAR IN NATURAL LANGUAGE PROCESSING 59, 65 (Tomek Strzalkowski ed., 1994).

⁶¹ See Hillis & Carmazza, *supra* note 9, at 229.

⁶² See Prakash M. Nadkarni, Lucila Ohno-Machado & Wendy W. Chapman, *Natural Language Processing: An Introduction*, 18 J. AM. MED. INFORM. ASS'N 544, 544 (2011); Socher et al., *supra* note 10, at 1201 (2012).

⁶³ See Socher et al., *supra* note 10, at 1201.

⁶⁴ See Nadkarni et al., *supra* note 62, at 544.

⁶⁵ See *id.*

⁶⁶ Lawmakers traditionally characterize the activities they seek to regulate,

been that parties who want computerized analysis of their contractual obligations would thus have to await advances in technological processing of natural language. However, in certain circumstances, there is an alternative approach—to reorient contract expression *away from* “natural language” sentences and *toward* a form more amenable to computer processing — precisely defined and highly structured data.⁶⁷ In some respects, this highly constrained format has a much more limited range of expressiveness as compared to natural language. However, for many purposes, contracting parties can, within the strictures of computer constraints, represent sufficiently meaningful information to delineate their contractual intentions.⁶⁸

1. Data-Oriented Contracting: How Parties Can Formulate Contracts as Data for Computers

Computers process information best when it is presented in a structured and precisely defined form. When information has been structured to be amenable to computer processing, computer scientists describe it as “machine-readable.”⁶⁹ Indeed, reformulating information in this way is a basic principle underlying modern computer software.⁷⁰ Programmers create software by using structured

prohibit, or incentivize using written language fixed in documents. One major reason that most laws are fixed in written texts is to satisfy public notice principles. *See, e.g.,* Russell L. Weaver, *Retroactive Regulatory Interpretations: An Analysis of Judicial Responses*, 61 NOTRE DAME L. REV. 167, 168-69 (1986) (discussing how the *ex post facto* clause of the United States Constitution and canons of statutory interpretation are meant to provide citizens with notice of legal requirements by protecting them from laws passed retroactively).

⁶⁷ ROBERT GLUSHKO & TIM MCGRATH, DOCUMENT ENGINEERING: ANALYZING AND DESIGNING DOCUMENTS FOR BUSINESS INFORMATICS & WEB SERVICES 17 (2008) (“To exchange documents, computers . . . require a precise and unambiguous language for describing information.”).

⁶⁸ RUSSELL & NORVIG, *supra* note 10, at 294-95.

⁶⁹ The key behind being machine-readable is that it has a precise, often formally defined, structure. *See, e.g.,* RALPH M. STAIR & GEORGE W. REYNOLDS, PRINCIPLES OF INFORMATION SYSTEMS 104 (9th ed. 2009) (“[M]achine-readable data can be understood and read by computer devices.”).

⁷⁰ Programmers do not write software by describing what they want the computer to do in ordinary natural language. Rather, they use precisely defined and structured formal computer programming languages, which have a limited and pre-defined set of instructions. High-level programming languages — such as C or Java — have precisely defined “syntax” which describes what inputs can be expected. The semantics of the language define the action that will be taken. For each instruction given to a computer, there is roughly a corresponding, unambiguous action that the computer must take. If a computer receives an instruction that is not in its pre-defined list, it will reject it as an error. *See* GARY MARRER, FUNDAMENTALS OF PROGRAMMING:

programming languages to convey information to computers in a precisely defined form that they can readily process.⁷¹ The data-oriented approach essentially adopts this technique to the task of expressing contract terms and meaning.

What does it mean for data to be structured? It means that there is a consistent, defined format for expressing and reading data permitting computer systems — which follow mechanical rules — to unambiguously extract data.⁷² Unlike in natural language, in which subtle contextual clues provide information about meaning and syntax, in structured data information is encoded according to precise and limited set of rules.

Consider some relatively unstructured natural language that might appear in a traditional option contract. An option contract gives one party the right, but not the obligation to buy or sell something (such as the right to purchase 100 shares of Google stock at \$400 per share).⁷³ Typically such contracts have an expiration date after which the option is no longer valid.⁷⁴ Given the freeform nature of natural language, the parties might express this information as: “this option expires on January 18, 2015” or “this contract giving the right, but not the obligation to purchase, shall no longer be valid after the 18th of January in the year 2015.” As described previously, the flexibility of language often poses challenges for automated techniques for processing language.

The data-oriented approach addresses this problem by representing contract terms in a well-defined and consistent form for encoding and extracting these terms. In other words, the data *is structured*. In the structured data approach, the range of expression is constrained as compared to natural language. Consider this somewhat stylized example of how a firm might represent an option contract’s expiration date in a data-oriented, structured form.⁷⁵

WITH OBJECT ORIENTATED PROGRAMMING 37-38 (2009).

⁷¹ ROBERT HARPER, PRACTICAL FOUNDATIONS FOR PROGRAMMING LANGUAGES (forthcoming Nov. 2012) (manuscript at 3).

⁷² See SIMON STOBART & DAVID PARSONS, DYNAMIC WEB APPLICATION DEVELOPMENT: USING PHP AND MYSQL 125-26 (2008) (“The important feature of structured data is that it follows a consistent and predictable format.”).

⁷³ CHANCE & BROOKS, *supra* note 15, at 2.

⁷⁴ See generally Fischer Black & Myron Scholes, *The Pricing of Options and Corporate Liabilities*, 81 J. POL. ECON. 637, 637-38 (1973) (discussing option contracts and the value of acting upon them quickly versus waiting until closer to the expiration date to act).

⁷⁵ For an example of a language for expressing the meaning of financial contracts, see generally Simon Peyton Jones, Jean-Marc Eber & Julian Seward, *Composing*

<Option_Expiration_Date: 01/18/2015>.⁷⁶

This rigid format — with the label “Option_Expiration_Date:” followed by a date on which the option expires — can convey essentially the same information about the expiration date as the descriptive sentences above. By imposing a rigid structure, the parties are simply ensuring that a computer will be able to read and process it accurately. Should the parties agree to always express their option contract’s expiration date using the label “Option_Expiration_Date:” followed by the date upon which the contract expires, they could later instruct a computer to reliably decode this information. Thus, we can consider this contract term to have been made “machine readable.” The parties have encoded it in a pre-defined fashion that allows a computer to reliably extract that information for later processing, as compared to the comparable natural language version.⁷⁷ Presumably, other core terms — such as price and quantity — would be represented in a similar, predefined, and highly structured way (e.g., <Exercise_Price:\$400>).⁷⁸

There are a few points to note about this example of reorienting contractual expression to be computer-processable. First, notice that the label “Option_Expiration_Date” is not just an identifier for a *computer* but has some human-understandable meaning. In other words, a person looking at this data record would likely be able to discern that this data was meant to indicate the date by which the parties intended the contract option to expire. This is a preview of the idea — explored below — that for data-oriented representations to be

Contracts: An Adventure in Financial Engineering, 35 ACM SIGPLAN NOTICES 280 (2000), available at <http://delivery.acm.org/10.1145/360000/351267/p280> (discussing various examples of computer-processing language to express defined financial contracts).

⁷⁶ This data format is loosely based upon XML, but it is simplified for explanatory purposes. It is important to note that for computable contracting purposes, the data can be in any format as long as it is structured. For more information about XML standard formatting. See, e.g., *Extensible Markup Language (XML)*, W3C, <http://www.w3.org/XML/> (last visited Mar. 6, 2012) (providing more information about XML standard formatting).

⁷⁷ Note, this “labeling” approach is not the only way to create structured data. As long as there is a pre-defined, consistent form for encoding or decoding information, a number of approaches may be used. See, e.g., Jones et al., *supra* note 75, at 282-85 (describing an alternative structured approach for setting important dates for option contracts).

⁷⁸ See, e.g., ESSVALE CORP. LTD., *BUSINESS KNOWLEDGE FOR IT IN INVESTMENT BANKING* (1st ed. 2006) (reviewing the data standards for conveying financial contracts); GROOT, *supra* note 31, at 65-68 (detailing the data formats in which common financial contracts are expressed).

useful, there has to be a link between the contracting parties' contractual intentions and the computer-oriented representation.

The second point is that parties wishing to create a data-oriented contract can elect among any number of conceptually similar technologies to produce structured representations of their contractual data.⁷⁹ Thus, some data-oriented contracts might be represented as computer-database records, others using data-description languages, and others using computer programming languages. We need not concern ourselves with the details of the particular technology or format chosen, as long as it is structured enough to be machine readable.⁸⁰ The larger point is that it is possible to express substantive contractual information in a particular form to later be readily processed by a computer system.

The third point is that the structured approach described simply allows for computers to reliably *identify and extract* core contract information. It says nothing about the computer systems understanding the "meaning" of the contract terms (i.e., what an "expiration date" means). The distinct issue of computer-understandable meaning will be explored in Part II. Further, the data-oriented approach does not require that *every* aspect of a contractual arrangement be represented in data. Rather, parties selectively identify portions of contractual arrangements that are both amenable to data-orientation (such as price or quantity), and for which it makes economic or business sense to do so.

Ultimately, when contract data is presented in the highly structured, data-oriented format, rather than in natural language, a computer system can readily and reliably extract core terms. This (partially) overcomes a technological limitation with respect to natural language because the parties are *not attempting to have a computer read* traditional language-based contracts, but rather, reorienting how they express contracts, mindful of the constraints of technology.

⁷⁹ GLUSHKO & MCGRATH, *supra* note 67, at 42-45.

⁸⁰ Contract information may be encoded in a variety of machine-readable formats. Thus, the parties might choose a "markup language" such as XML. They might choose to represent the contract as a database record, in which case the information would be interpretable according to a structured database schema. Or, they might encode the contract within a computer programming language such as C or Java. *Id.*; see also Aleksandra Nenadic & Ning Zhang, *Non-Repudiation and Fairness in Electronic Data Exchange*, in ENTERPRISE INFORMATION SYSTEMS V 286, 290 (Olivier Camp et al. eds., 2004).

D. Endowing Data with Legal Substance

There are some basic questions when expressing contracts as data. First, how does the data acquire the shared meaning necessary to effectuate an agreement? Relatedly, how should such data be interpreted? These issues will be explored below.

1. Endowing Data with Shared Meaning

Upon inspection, a data-oriented contract would appear to be a mix of computer data and computer instructions. It may not be obvious how such computer data can obtain the level of meaning necessary to permit parties to express their contractual intentions to one another. In the traditional contracting paradigm, the process of meaning acquisition is comparatively straightforward. Most contracts use descriptive language to explain what the contract is about, and to delineate the parties' mutual understandings and goals. The meaning of contract obligations often arises from the ordinary, shared meaning of the words (e.g., "I agree to sell you my car for \$500").⁸¹ When more specific meanings are required, parties will often define words within the body of a contract document.⁸² Similarly, in data-oriented contracting, parties must explicitly ascribe meaning to data.

One way that contracting parties can endow computer data with shared interpretations is through a "data-meaning" threshold agreement.⁸³ Threshold agreements are traditional, written language documents that parties agree to before engaging in data-oriented contracting.⁸⁴ Such agreements serve as a legal foundation for subsequent data-oriented contracting. They address, at the outset, important topics such as the meaning of data, or processes for handling unanticipated exceptions.⁸⁵

⁸¹ Sometimes meaning arises from earlier default interpretations provided by external sources, such as courts, administrative agencies, or standards bodies. See Marcel Kahan & Michael Klausner, *Standardization and Innovation in Corporate Contracting (or "The Economics of Boilerplate")*, 83 VA. L. REV. 713, 717-19 (1997) (explaining that contracting parties adopt implicit or explicit definitions from external sources). The term "threshold agreement" is this Article's own.

⁸² See *id.* at 719.

⁸³ The terms "threshold agreement" and "data-meaning" threshold agreement are terms coined by this Article due to absence of existing terminology.

⁸⁴ See Kahan & Klausner, *supra* note 81, at 762-63 (describing standards in contract contexts).

⁸⁵ See, e.g., KATHERINE L. LYNCH, *THE FORCES OF ECONOMIC GLOBALIZATION: CHALLENGES TO THE REGIME OF INTERNATIONAL COMMERCIAL ARBITRATION* 370-79 (2003) (discussing agreements specifying dispute-resolution procedures in threshold

A “data-meaning” threshold agreement provides specific interpretations to be given to data.⁸⁶ Contracting parties can create these agreements to ensure that they have a consistent, shared meaning for data.⁸⁷ For example, if parties are creating a data-oriented equity option contract, they may create a document in which they detail the data format and intended interpretation of the core terms.⁸⁸ They might agree that an option expiration date will always be represented by data that is labeled “Option_Expiration_Date:”, and that any date following this label should be interpreted as the expiration date of the contract. Since contracts ultimately involve agreements between people (or organizations run by people) these agreements provide a link between the parties’ understanding of contractual intentions and computerized representations of those obligations.

a. *Acquiring Shared Meaning by Adhering to Published Computer Standards*

Rather than defining the meaning of contract data themselves, parties might instead incorporate existing data standards.⁸⁹ Data-standards are specifications that provide shared formats for data, and descriptions for how data should be interpreted, and are capable of expressing standard contracting scenarios.⁹⁰ Such standards are often created by a centralized body and are often publically available.⁹¹ This

agreements or terms of service).

⁸⁶ See BRAHIM MEDJAHED & ATHMAN BOUGUETTAYA, SERVICE COMPOSITION FOR THE SEMANTIC WEB 14 (2011).

⁸⁷ See JAMES BEAN, ENGINEERING GLOBAL E-COMMERCE SITES: A GUIDE TO DATA CAPTURE, CONTENT, AND TRANSACTIONS 1-2 (2003) (“[B]oth participants of [an] e-commerce model . . . may operate under a formal or informal contract . . . with agreed-upon terms such as scheduling, pricing, delivery, and support.”).

⁸⁸ See Stephen J. Choi & G. Mitu Gulati, *Contract as Statute*, 104 MICH. L. REV. 1129, 1140-41 (2006).

⁸⁹ Standardized descriptions of concepts in particular domains can be defined in a data ontology. See KHANNA, *supra* note 16, at 73 (“Financial ontologies define the concepts and the relationships . . . pertinent to a particular domain. For example, an ontology for an equity derivative call option will include the concept of strike price, exercise date, underlying asset and underlying asset price, and the price of the call option itself. They are usually written in a standard data format language such as XML.”).

⁹⁰ See GLUSHKO & MCGRATH, *supra* note 67, at 17-20.

⁹¹ An example of standardized data interpretive structures is the Universal Business Language (UBL). UBL sets out standard forms for automating contracting involving data-oriented representations of documents like purchase order contracts. See *id.* at 17.

is an efficient way to endow data with meaning.⁹² For example, electronically exchanged financial contracts are commonly composed according to a set of pre-defined data-standards capable of expressing common financial contracts.⁹³ The benefit of adhering to an existing standard is that: 1) contracting parties need not expend resources creating their own definitions, but rather can borrow predefined formats; and 2) multiple parties can interact with one another using a shared data-language.⁹⁴

One main benefit of the data-standards over the threshold-agreement approach is that, in the former, the parties do not necessarily need to meet initially to sign a written foundational agreement.⁹⁵ Rather, each party can indicate electronically that they are adhering to one particular, shared-data standard. This enables parties to engage in “ad-hoc” automated transactions without having a pre-existing contractual relationship.⁹⁶ Both parties’ computer systems can electronically indicate to one another that they are sharing a common interpretation for the data by reference to a public standard.⁹⁷

⁹² See Nenadic & Zhang, *supra* note 80, at 290 (describing electronic contracting frameworks); see also MEDJAHED & BOUGUETTAYA, *supra* note 86, at 14.

⁹³ There are a number of data standards used in finance for specifying contractual information. These include the FIX protocol, FpML, MDDL, Fin XML, and SWFIT. See, e.g., ESSVALE CORP., *supra* note 78, at 59 (“FpML . . . is an XML message standard for the OTC Derivatives Industry. All categories of privately negotiated derivatives will eventually be included within the standard. The standard is managed by ISDA on behalf of a community of investment banks that make a market in OTC derivatives.”); GROOT, *supra* note 31, at 66-73 (describing the major data standards used in financial contracting); D. Craig Norlund, *Electronic Dissemination of Disclosure Documents*, in PRACTISING LAW INSTITUTE CORPORATE LAW AND PRACTICE COURSE 1999, at 113 (PLI Corp. Law & Practice, Course Handbook Series No. 39, 1999), WL 1093 PLI/Corp 39 (“The FIX protocol is a message standard developed to facilitate the electronic exchange of data related to securities transactions, including indications of interest, orders, fills, executions, allocations and confirmations.”).

⁹⁴ These pre-defined data standards permit financial firms to trade a given standardized financial contract multiple times among multiple parties — as each of the parties are using the same data-formation and interpretive standard. See generally Andrew A. Schwartz, *Consumer Contract Exchanges and the Problem of Adhesion*, 28 YALE J. ON REG. 313, 318 (2011) (discussing the trading of standardized contracts).

⁹⁵ MARTIEN SCHAUB, EUROPEAN LEGAL ASPECTS OF E-COMMERCE 6-9 (2004) (describing how the open architecture of the Internet, combined with data standards, allows for ad-hoc transactions from parties who do not necessarily have any previous relationship).

⁹⁶ *Id.*

⁹⁷ For example, the Danish government requires all invoices requesting payment for government purchases to be submitted *electronically* according to a data standard, so that payment can be automated. Once a firm sells a product to the Danish government, in order to receive payment, they must send a standardized electronic,

In essence, the parties' computers agree, at the time of contracting, that each is using the same interpretive basis.

2. Electronic Contracting Interfaces

How do parties express their contractual intentions as computer-processable data? In principle, firms could write language-based contracts and then translate them into the structured form preferred by computers (e.g., "This contract expires on January 18, 2015" into "<expiration_date: 01/18/2015>"). In practice, this process often occurs more invisibly through the use of electronic contracting interfaces.⁹⁸ Essentially, parties enter contract terms through a computer that allows contract terms to be seamlessly translated into data as they are entered. Similarly, existing contracts expressed as data can be decoded and presented on the screen in human-friendly, readable form. In commercial contracting, such interfaces are often provided by third-party firms used by both buyer and seller as intermediaries.⁹⁹ If contracting occurs through such a third-party interface, the parties may be unaware of the their contract's data format, or that it is expressed in data at all — relying upon the third party firm to manage those details.

On-screen computer interfaces are structured to impose constraints on the way in which contract information is entered. Structured interfaces are familiar to those who have purchased over the Internet (e.g., to capture delivery information, websites require buyers to type the street address in one on-screen box, and the zip-code in another). At a basic level, contracting through such an electronic interface allows a computer to capture core contract information *electronically* at the outset as it is expressed — rather than requiring a subsequent manual conversion from descriptive words to data.

More significantly, the highly-constrained format of such interfaces — requiring the entry of each distinct piece of information (e.g., price,

data-oriented invoice. Such an electronic invoice is simply a data record — it does not resemble a traditional paper invoice, composed of descriptive language (e.g., "Please pay \$20.00 for goods sold. Amount due by August 1, 2012"). Rather a version of the data record instead might look something more akin to "Amount: 20.00; Payment Date: 01-01-2015." Kelly Ng, *Denmark Helps Businesses with National E-Invoicing System*, FUTUREGOV (Jan. 13, 2010), <http://www.futuregov.asia/articles/2010/jan/13/denmark-helps-businesses-national-e-invoicing-syst/> (last visited Mar. 3, 2012).

⁹⁸ See RITENDRA GOEL, E-COMMERCE 17-18 (2007).

⁹⁹ See SCHAUB, *supra* note 95, at 9 (discussing the role of trusted third-party intermediaries). Much of electronic financial contracting is conducted through electronic trading platforms operated by third party firms. JOHN TEALL, FINANCIAL TRADING AND INVESTING 17 (2012).

quantity) in its own separate, labeled on-screen “entry box” — is the key to seamlessly translating contract data from human-understandable to data-oriented form.¹⁰⁰ When a party is required to enter each piece of contact information in its own distinct on-screen box — such as one labeled “price” — the parties are implicitly identifying to the computer how that piece of data should be interpreted (i.e., this information should be stored and interpreted as a price). Thus, the parties are being forced to identify to the computer the precise details of contract data as they enter them (e.g., identifying precisely who parties to the contract are). This partially overcomes the problem, described earlier, in which computers would be challenged in contextually identifying core contract information in the natural language version of such a contract.¹⁰¹ Instead, the constrained conduit by which parties enter information reduces uncertainty in communicating the purpose of contract terms to computers while preserving considerable expressive flexibility.

3. Incorporating Data as the Expression of the Contract: Threshold Procedural Agreements

Parties engaging in data-oriented contracting often use what might be called threshold “procedural agreements” to incorporate the data-oriented expression as their contractual expression.¹⁰² Essentially, these agreements are not necessarily focused on the meaning of data, but rather set up the procedural foundation for future data-oriented or written-language electronic contracting between the parties. Thus, such an agreement might explicitly recognize that the parties are contracting electronically and that the data records should be considered as their contractual expression. Such a procedural agreement might also set up a process for dealing with unexpected

¹⁰⁰ See KHANNA, *supra* note 16, § 2.31 (“An order is created when the trader chooses a security to trade, and inputs details such as the type of price order and quantity. The system then automatically generates a trade ticket, which contains all the basic description of the trade.”).

¹⁰¹ For example, instead of expressing the intention that a contract have an expiration date using written language (e.g., “This contract expires on January 18, 2015.”), the party might instead express equivalent information by “01/18/2015” in the “Expiration Date” field. The party is asserting, at a basic level, that the contract has an expiration date, and at a specific level, that the computer should interpret the entered date as the contract’s expiration date (as opposed to, say, the contract formation date).

¹⁰² For an example of another type of threshold agreement, see Amazon.com Seller Participation Agreement, AMAZON.COM, <http://www.amazon.com/gp/help/customer/display.html?ie=UTF8&nodeId=1161302> (last visited Oct. 1, 2012).

scenarios that are outside of the capacity of the computer systems — such as data-corruption or erroneously executed contracts.¹⁰³ For example, in the case of financial derivatives contracts, parties typically adopt provisions from a standard agreement called the “ISDA Master Agreement.”¹⁰⁴ This agreement sets up a framework that, at the outset, handles various considerations that prove foundational to allowing for data-oriented contracting.¹⁰⁵

E. Contract Doctrine and Data-Oriented Contracts

Does contract doctrine accommodate the expression of contracts as data rather than words? Because contract law has developed within a paradigm in which binding promises are mostly communicated using written or spoken language (e.g., English), it is possible that a data-oriented contract — which is in form, quite different from a traditional written or spoken contract — could be at odds with traditional doctrine. At a minimum, contract laws do not explicitly prohibit expressing contractual obligations in terms of data. More affirmatively, basic contracting principles actively accommodate data-oriented representation.

First, traditional contract doctrines allow flexibility in the form of expression.¹⁰⁶ Although contracts are typically conceived of as being expressed through written or spoken language, contract law also permits non-language-based contracts — those implied from conduct.¹⁰⁷ Thus, contract expression can occur in a form other than traditional written or spoken language.¹⁰⁸ Moreover, contract

¹⁰³ MANAGED FUNDS ASS'N, THE EVOLUTION OF STANDARDIZATION IN THE OTC DERIVATIVES MARKET 1-3 (2009), available at <http://canambar.michaelvacirca.com/wp-content/uploads/2011/05/MFARreporterArticle.pdf>.

¹⁰⁴ *Id.*; VINOD KOTHARI, CREDIT DERIVATIVES AND STRUCTURED CREDIT TRADING Ch. 20 (2011); PHILIP R. WOOD, SET-OFF AND NETTING, DERIVATIVES, CLEARING SYSTEMS § 12-002-003 (2007).

¹⁰⁵ Sharon Brown-Hruska, *The Derivatives Marketplace: Exchanges and the Over-the-Counter Market*, in FINANCIAL DERIVATIVES: PRICING AND RISK MANAGEMENT 21, 24 (Robert W. Kolb & James A. Overdahl eds., 2009); Thomas D'Ambrosio, *Thorny Issues Encountered When Negotiating ISDA Master Agreements*, 3 J. OF SEC. L. REG. & COMPLIANCE 71, 71-76 (2010). For the text of the ISDA Master agreement, see also ISDA Bookstore, INT'L SWAPS AND DERIVATIVES ASS'N, <http://www.isda.org/publications/isdamasteragrmnt.aspx> (last visited Oct. 21, 2012).

¹⁰⁶ To see this flexibility, consider that contract doctrine does not restrict the expression of contractual terms or conditions to an exhaustive set of pre-approved formulations. See RESTATEMENT (SECOND) CONTRACTS § 2 cmt. a (1981).

¹⁰⁷ RESTATEMENT (SECOND) CONTRACTS § 4 (“A promise may be stated in words either oral or written, or may be inferred wholly or partly from conduct.”).

¹⁰⁸ *Hercules, Inc. v. United States*, 516 U.S. 417, 424 (1996) (Rehnquist, C.J.) (“An

principles deliberately employ expansive terminology concerning contract expression.¹⁰⁹ Additionally, as described above, data can be given common meaning sufficient to meet contract's goal of shared understanding of contractual commitments.¹¹⁰ Finally, the Uniform Electronic Transactions Act (UETA) and the Uniform Computer Information Transactions Act (UCITA)¹¹¹ — model legislation concerning issues related to electronic contracting — seem to implicitly acknowledge the possibility of data-oriented contracts.¹¹² A primary unresolved tension may occur in future scenarios where there is both a written and data-oriented representation of the same contractual expression, with interpretations that differ. However, the memorialization and expression of a contract as data, intended primarily for computer processing, does not appear in itself to present a difficulty for general contract doctrine.

F. The Importance of Data-Oriented Expression

Data-orientation is important because it effectively enables new contracting properties as compared to the traditional language-based paradigm.¹¹³ Computers excel at comparing, sorting, organizing, and

agreement implied in fact is 'founded upon a meeting of minds, which, although not embodied in an express contract, is inferred, as a fact, from conduct of the parties showing, in the light of the surrounding circumstances, their tacit understanding.'").

¹⁰⁹ RESTATEMENT (SECOND) CONTRACTS § 4 (asserting that a legally binding promise can be demonstrated by a suitable "manifestation" of an intention to act).

¹¹⁰ See, e.g., *Krasley v. Superior Court*, 101 Cal. App. 3d 425, 431 (1980) ("The essence of a contract is the meeting of minds on the essential features of the agreement."). One might wonder whether a contract composed out of computer data could be similarly amenable to mutually shared agreement. However, as described in Part I, contracting parties can give common, understandable meaning to data by agreeing in advance to specific interpretations and meanings (e.g., via threshold agreements or public data standard protocols).

¹¹¹ UNIF. COMPUTER INFO. TRANSACTIONS ACT § 101, 7 U.L.A. 212 (2002); UNIF. ELEC. TRANSACTIONS ACT § 1, 7 U.L.A. 226 (1999).

¹¹² Neither UCITA nor UETA have been universally adopted, and in many cases of electronic contracting, traditional contract doctrine still governs. As of 2012, UETA has been adopted by forty-six U.S. jurisdictions, but UCITA has only been adopted by two jurisdictions (Maryland and Virginia). See UNIF. COMPUTER INFO. TRANSACTIONS ACT § 206, 7 U.L.A. 305 (2002) (acknowledging implicitly the possibility of data-oriented contracts through the discussion of contracts formed autonomously through electronic agents); UNIF. ELEC. TRANSACTIONS ACT § 14, 7A U.L.A. 38 (Supp. 2000).

¹¹³ Joost Breuker, André Valente & Radboud Winkels, *Use and Reuse of Legal Ontologies*, in *LAW AND THE SEMANTIC WEB: LEGAL ONTOLOGIES, METHODOLOGIES, LEGAL INFORMATION RETRIEVAL, AND APPLICATIONS* 37-39 (Richard Benjamins et al., eds. 2005) (describing how once legislation was modeled as data, computer systems could detect certain contradictions among different legal rules and obligations).

analyzing data. However, as described earlier, the substance of contractual obligations — what it was the parties were contracting about — was effectively inaccessible to most computers when contracts were expressed in “natural language.” However, once contract information has been deliberately formulated to be processable by computers, parties can take apply the analytical abilities of computers to the substance of their contractual obligations. For example, firms can more readily detect conflicts among their legal obligations once they have been represented in data.¹¹⁴ Part IV will discuss these and other potential benefits. However, the next part will focus on one particularly significant ability enabled by data-orientation: computable contract terms.

II. COMPUTABLE CONTRACT TERMS AND CONDITIONS

Once contract terms or conditions have been represented in data-oriented form, parties can take advantage of some novel analytical abilities. One such possibility is the creation of “computable” contract terms. The basic idea behind a computable contract term is to create a series of actionable, computer-processable instructions that approximate what it is that the parties are intending to do in their contractual arrangement. In certain contexts, computer systems can be instructed *how* to assess contract terms in a way that mirrors the parties’ intentions. Further, the parties can sometimes provide the computer with data that is relevant to making determinations of conformance with specified contract terms.

By specifying in computer processable form the core terms of a contract, the *meaning* of these terms, and data relevant to assessing conformance with those terms, contracting parties can sometimes enable automated, *prima-facie* determinations as to compliance.¹¹⁵ The qualification “prima-facie” indicates that such automated assessments of conformance or non-conformance with contract terms may be legally tentative, “first-cut” determinations — rather than legally conclusive outcomes.¹¹⁶ This part explains how to enable computers to

¹¹⁴ See *id.*

¹¹⁵ This Article will examine other examples of law being computed through the logical deduction of rules outside of the contracting context. See *infra* Part V. A familiar example, explored herein, is the personal income tax context, in which tax liability is computed based upon data and rules. For a detailed explanation of this process, see Harry Surden, *The Variable Determinacy Thesis*, 12 COLUM. SCI. & TECH. L. REV. 1, 70-75 (2011) [hereinafter *Variable Determinacy*].

¹¹⁶ Most importantly, “prima-facie” is meant to reflect that an automated determination of compliance may not, in some instances, reflect the ultimate

“understand” the meaning of certain contract terms, and how this can permit automated assessment of conformance.

A. What It Means To Be Computable

This section first introduces the computable contract concept intuitively through basic examples. It then explores the underlying principles more rigorously, presenting illustrations of more complex computable contract terms. The following examples illustrate two main points: 1) it is sometimes possible to “translate” the meaning of contractual concepts into computer instructions that effectively reflect the parties’ contractual intentions; 2) there is sometimes available, in computer-processable form, information that is relevant to determining compliance or non-compliance with contract terms.

1. Example 1: Certain Financial Contract Terms

This Article has previously suggested that many¹¹⁷ electronically traded financial contracts are primarily expressed as data.¹¹⁸ This subpart will explore two previously encountered examples of such contracts with basic terms that can be considered (or have the possibility of becoming) computable.

We have already encountered one example of a financial term that might be considered computable. Recall the earlier discussion of a currency contract in which one party promised to pay another party some amount of currency at some future date.¹¹⁹ We might consider verifiable records of payments between the parties to be informative of performance. If such data were available electronically, computers could be used to make automated, *prima-facie* assessments about compliance with these obligations to pay. In the financial world,

judgment of a legal arbiter.

¹¹⁷ See generally Schwartz, *supra* note 94, at 313 (discussing the trading of standardized contracts).

¹¹⁸ Different financial organizations may have different proprietary formats, but for exchanging trades electronically for a variety of standard financial contracts, firms often employ the standard FIX protocol. ANDREW BRADFORD, *THE INVESTMENT INDUSTRY FOR IT PRACTITIONERS: AN INTRODUCTORY GUIDE* 29 (2008) (“Financial institutions use the industry standard FIX protocol to quickly communicate trades and trade information electronically between exchanges and counter parties.”); see also THE FIX PROTOCOL ORGANIZATION, <http://fixprotocol.org/specifications/> (last visited Mar. 5, 2012).

¹¹⁹ Performance may be conducted, not by delivering the amount of the underlying currency per-se, but by paying the value of the contract as computed based upon the exchange rate in some other currency. See *DEUTSCHE BORSE GRP.*, *supra* note 17, at 6, 15.

something akin to this computation and monitoring of term compliance happens through systems that automate settlement and netting of financial contracts.¹²⁰ In these processes, the amounts owed between firms, based upon the value of these financial contracts, are often automatically assessed and compared against actual payment flows between contracting parties.¹²¹ Such terms may be considered computable in the sense that a computer system was able to compare the terms and obligations of these contracts to data that is relevant to assessing performance — payment records.

Another example of a computable term comes from the option contracts discussed earlier in this Article. Such contracts give the holder the right, but not the obligation, to buy or sell something at a particular price. A characteristic of an option contract is an expiration date after which such options are no longer valid.¹²² Recall that when such a contract is data-oriented, it expresses certain information, such as that contract's expiration date, in structured, computer-processable form. However, data-orientation simply suggests that a computer system can reliably extract and identify the expiration date — more is needed to make this condition actually *computable*. Computability implies that a computer has the ability to engage in some sort of automated comparison between the terms of the contract and relevant information concerning compliance.

To make such an expiration date contract term computable, the parties might “translate” the meaning behind an expiration date into a set of computer instructions that replicate its underlying logic and provide data relevant to conformance with this restriction. The logic behind an expiration date term in an option contract is that the option is no longer valid to execute after the date has passed. The data relevant to make this determination is the expiration date and the date upon which a party attempts to execute the contract. Thus, a comparable computer-based translation might be an instruction to a

¹²⁰ Sometimes the settlement and netting calculation is conducted through a central organization called a central clearing house. See, e.g., DAVID LOADER, *CLEARING AND SETTLEMENT OF DERIVATIVES* 35-36 (2005) (describing the role of the clearing house as acting as a counterparty to both sides of the trade).

¹²¹ In some instances, the amounts owed are directly debited from bank accounts of the contracting parties to reduce risk. See LOADER, *supra* note 120, at 40-45; see also John T. Lynch, Comment, *Credit Derivatives: Industry Initiative Supplants Need for Direct Regulatory Intervention — a Model for the Future of U.S. Regulation?*, 55 *BUFF. L. REV.* 1371, 1395 (2008) (discussing netting).

¹²² For example, such a contract might give the holder the legal right to purchase 100 shares of Google stock at a price of \$400 per share, at any time before January 18, 2015.

computer to compare an *attempted* option execution date to the option expiration date specified as the contract term, and to take some relevant action if the date has passed. Thus, if a party attempts to execute an option on February 1, 2015, but the option expired earlier on January 18, 2015, a system might compute that this date has expired and react appropriately. This might include disallowing execution, or flagging erroneously executed contracts.¹²³

In a functional sense, we might consider such an expiration date term to be computable. The parties were able to provide a reasonable “translation” of the meaning of an expiration date as a set of computer instructions, and then provide the computer with data relevant to compliance (i.e., the actual date of attempted execution).¹²⁴ By comparing the date of execution to the date of expiration, the computer has conducted an automated, *prima-facie* assessment of compliance with this particular term. Such a comparison is of the type that would have been conducted manually had this contract been made in the traditional written paper-contracting paradigm.¹²⁵

2. Example 2: Geographic Limitations on Online IP

Consider another example involving a firm that streams movies to subscribers over the Internet. Imagine that the licensing contract limits streaming to viewers located within the United States.¹²⁶ In the traditional contracting paradigm, the parties might express this condition with language such as “this license only authorizes streaming to subscribers located in the United States.” However, there is a reasonable data-oriented “translation” of this criterion. Firms can make relatively accurate and automated approximations as to the geographic location of viewers based upon the number (“I.P.

¹²³ BRADFORD, *supra* note 118, at 29.

¹²⁴ BENJAMIN VAN VLIET, BUILDING AUTOMATED TRADING SYSTEMS: WITH AN INTRODUCTION TO VISUAL C++.NET 2005, at 148 (2007).

¹²⁵ KHANNA, *supra* note 16, at 82.

¹²⁶ This example is roughly modeled on the example of Netflix, whose video streaming service was initially only available within the United States, and which automatically blocked viewing from abroad based upon IP address detection. See The Associated Press, *Netflix to Stream Movies in Canada Later This Year*, SEATTLE TIMES (July 19, 2010 12:41 PM), http://seattletimes.com/html/business/technology/2012395469_apusnetflixcanada.html; Chris Griffith, *Media Streams Spark Piracy Row Over Copyright*, AUSTRALIAN (June 21, 2011 12:00 AM), <http://www.theaustralian.com.au/australian-it/media-streams-spark-piracy-row-over-copyright/story-e6fmgakx-1226078817583>. Please note that I do not have knowledge of the details as to how Netflix actually agreed or implemented their contractual term. Rather, the scenario is suggestive of one that would be consistent with computable contracting practices.

Address”) that is assigned to each computer on the Internet.¹²⁷ The parties might agree that such an assessment, while imperfect, is an efficient and sufficiently accurate representation of their intended condition of performance and geographic location and is preferable to the traditional written clause.¹²⁸

One reason that the parties might prefer this computer-based expression of a contract term to its language-based analog is the efficiency by which the streaming firm can “compute” compliance by instructing their computers to detect (or automatically exclude) non-conforming users based upon their geographic location. The interesting point is that the parties might specifically agree, up front, that the contract condition is sufficiently met by *automated* geographic detection — even if such a process is occasionally incorrect relative to a user’s *actual* geographic location. In this way the parties have contractually stipulated a computable contract condition, indicating that the imperfect, but automatable, substitute should be understood to constitute contractual compliance. In this case, *prima-facie* compliance with such a term can be specifically made computable by reference to relevant data (e.g., detected geographic location) and *ex-ante* agreement by the contracting parties.

3. Example 3: Stanford Intellectual Property Exchange (SIPX)

Another example of computable contract terms comes from the Stanford Intellectual Property Exchange (SIPX) pilot program, which allows for the specification of computable intellectual property licensing terms.¹²⁹ Stanford University has used this technology to

¹²⁷ The firm’s computers might examine data about the user’s Internet Protocol (IP) address and credit card billing country to make an automated assessment as to the likely location of the user. Online services can use the IP address to make a reasonably accurate assessment of a user’s geographical location. When a computer connects to the Internet, it is assigned an IP address. Users access the Internet by connection to an Internet Service Provider (ISP), which is often located relatively near to the user. It is possible to link the user’s IP address to a particular ISP, thus making a determination as to that ISP’s geographic location. In this way, it is possible — at a high level of granularity — to make a sufficiently accurate but imperfect assessment as to the likely geographic location of the user.

¹²⁸ See Bamba Gueye et al., *Leveraging Buffering Delay Estimation for Geolocation of Internet Hosts*, in NETWORKING 2006: NETWORKING TECHNOLOGIES, SERVICES, AND PROTOCOLS, PERFORMANCE OF COMPUTER AND COMMUNICATION NETWORKS, MOBILE AND WIRELESS COMMUNICATIONS SYSTEMS: 5TH INTERNATIONAL NETWORKING CONFERENCE, COIMBRA, PORTUGAL, MAY 2006: PROCEEDINGS 319, 321-23 (Fernando Boavida ed., 2006).

¹²⁹ See THE STANFORD INTELLECTUAL PROPERTY EXCHANGE PROJECT, <http://sipx.stanford.edu/> (last visited Oct. 22, 2012).

implement a service that allows students to print course materials for which they have been licensed.¹³⁰ Under systems such as SIPX, copyright holders might specify relatively complex copyright licensing terms capable of automated resolution by computer systems.

For an illustration, imagine that the University wishes to reduce its academic licensing fees by matching academic resources more closely with those units likely to use those resources. For example, in exchange for a reduced licensing fee, the University might negotiate access to engineering journals — not for all students, but to the subset of students most expected to use them (e.g., engineering students or students currently enrolled in engineering courses).¹³¹ In this imagined contract, the copyright holders limit the license for engineering related resources to students who are engineering majors or are studying an engineering course.

In the SIPX system, such terms can be considered computable. In the traditional paradigm, the parties might have expressed this condition using language such as, “this license limits use of engineering materials to students in engineering related courses of study.” However, there is a computable translation of this condition that can be specified by the license holder using the SIPX system. The parties might link to the University’s student information database, which stores information about students’ courses of study. Thus, when a given student attempts to access or download an engineering article, the system might cross reference data about that student’s major and courses and automatically indicate (or enforce) *prima-facie* conformity or violation with the specified condition.

B. Principles of Computable Contracts

The above examples illustrate the meaning of the computable contract terminology. In other domains, the term “computable” is used to describe contexts where a computer has a process for determining some result through calculations.¹³² Analogously, we can

¹³⁰ See MEDIA X AT STANFORD UNIVERSITY, <http://mediax.stanford.edu/pod-s11> (last visited Mar. 5, 2012).

¹³¹ To be clear, I am not arguing that this licensing arrangement is superior or even good. We could imagine good reasons for non-engineering students to have access to engineering journals to foster cross-disciplinary scholarship. Rather, it is an example of an arrangement that is more feasible once a contractual arrangement has been rendered computable, which is more difficult to achieve in the traditional, written context.

¹³² My usage in this Article is more informal and colloquial and thus distinct from the technical usage of “computability” in *computability theory* in computer science. In

consider the contract terms from the examples to be computable in the sense that the parties have provided the computer system with the means and the data to make automated, *prima-facie* assessments of conformance with certain contract terms. As will be discussed shortly, the major implication is a significant reduction of particular contracting transaction costs where computable terms are possible.¹³³

The examples also illustrate the major principles underlying computable contract terms. First, some contract term or condition must have been capable of translation into a consonant set of computer instructions (e.g., the concept of an expiration date as a rule to comparing dates). Second, there existed data relevant to compliance with contract terms that was available in computer-processable form (e.g., electronic payment data records) and that the parties could provide to the computer for automated comparisons.

Clearly, these constraints are limiting. Therefore, a large number of contract topics, subjects, or criteria will not be computable in this sense. Some contract criteria will be discretionary or incapable of being definitively or usefully measured. Part III discusses these limitations. However, in certain contexts, there *will* be terms amenable to the computable paradigm. In those contexts, by linking computer-processable terms to data about the world, parties can enable automated comparisons between what was promised contractually and what actually has or has not happened.¹³⁴ The following sections generalize about how these principles might be extended to other contracting scenarios.

the sense used in this Article, “computable” simply means that the legal result is able to be automatically generated by a rules-based process. *See, e.g.*, NIGEL CUTLAND, COMPUTABILITY: AN INTRODUCTION TO RECURSIVE FUNCTION THEORY 8 (1980) (“When an algorithm or effective procedure is used to calculate the values of a numeric function, then the function in question is described by phrases such as [computable].”). This is distinct from a more technical and precise usage in computability theory. For an introduction to computability theory, and that distinct usage, see MICHAEL SIPSER, INTRODUCTION TO THE THEORY OF COMPUTATION 1-10 (2d ed. 1997).

¹³³ *See* discussion *infra* Part V.

¹³⁴ Moreover, in each of these examples, the data records that constituted contractual representations were situated within an electronically mediated context. “Electronically mediated” simply refers to the fact that the interactions occurred using computer systems (e.g., via electronic financial trading systems or online services) rather than in a commercial context completely unconnected to computers and data (i.e., an oral contract to purchase an automobile). *See, e.g.*, Love & Genesereth, *supra* note 20, at 205 (describing “an approach to automated legal reasoning focusing on semantically rich laws, regulations, contract terms, and business rules in the context of electronically-mediated actions.”).

1. Computer Semantics: Conveying the Meaning of Concepts to Computers

This sub-part explains the general theory by which parties convey the “meaning” of contractual terms to computer systems.¹³⁵ In computer science, the process or theory of conveying meaning to computer systems is known as “semantics.”¹³⁶ The background problem is that computers, on their own, do not understand the meaning of words or concepts¹³⁷ comparable to the way people are thought to understand meaning — at a deep cognitive level.¹³⁸ How do we then convey “meaning” to computer systems that do not have cognitive processes for understanding concepts? The answer is that we can often provide systems with computer instructions that allow them to react appropriately *as if they did* understand the meaning of words.¹³⁹ Often, this functional approach to approximating meaning is sufficiently robust for a computer to reflect the desired intentions of those using the computers, even if the computer does not understand meaning at a deep conceptual level.¹⁴⁰

To tell a computer what a word means, in many cases, is to provide a translation between a given word and a set of computer instructions producing outputs that are consistent with what *a person* would understand the word to mean.¹⁴¹ For example, contemporary

¹³⁵ A more robust review of computer semantics is beyond the scope of this Article.

¹³⁶ The term semantics has several distinct, technical uses within computer science. For the purposes of this Article, I will employ an informal usage of the word, which implies providing a linkage between a concept that is understandable to a person and a set of instructions or actions that a computer can carry out. See, e.g., CARL A. GUNTER, SEMANTICS OF PROGRAMMING LANGUAGES: STRUCTURES AND TECHNIQUES 9-10 (1992) (“A crude view is that the semantics of a programming language . . . is the mapping . . . from the program written by a human to the target executed by a computer.”).

¹³⁷ HUBERT DREYFUS, ON THE INTERNET 20 (2008) (“[C]omputers . . . don’t have bodies, don’t share our world, and so don’t understand the meaning of our documents and websites.”).

¹³⁸ Hillis & Carmazza, *supra* note 9, at 229.

¹³⁹ DAN JURAFSKY & JAMES H. MARTIN, SPEECH AND LANGUAGE PROCESSING: AN INTRODUCTION TO NATURAL LANGUAGE PROCESSING, COMPUTATIONAL LINGUISTICS, AND SPEECH RECOGNITION 2 (2d ed. 2009) (“What distinguishes . . . language processing applications from other data processing systems is their use of knowledge of language.”).

¹⁴⁰ See, e.g., Larry Hardesty, *Computer Learns Language by Playing Games*, MIT NEWS (July 11, 2011), <http://web.mit.edu/newsoffice/2011/language-from-games-0712.html> (describing how computer software created at MIT developed rules that allowed it to assess text-based instructions for a computer game and improve its ability to win the game based upon approximate rules derived).

¹⁴¹ LAURA C. RIVERO, JORGE H. DOORN & VIVIANA E. FERRAGGINE, ENCYCLOPEDIA OF

computers would not, on their own, understand the meaning of the word “print.” Yet when we issue a print command to a computer a series of appropriate reactions occur — such as the printing of a document on a nearby printer — that are in line with our understanding of the word “print.” This happens because, at some point, computer programmers effectively provided a translation between the concept “print” and a sensible, computer-instruction-based process that results in the printing of documents at a printer. At the risk of oversimplification, this is often the general approach by which people convey meaning to computers — we use computer rules and data to associate words (e.g., “print”) with a set of outputs (e.g., a printer producing a document) that are sensible given the understood meaning of a particular word.

This general translation and association approach is an implicit premise behind much of computer programming.¹⁴² This point is important because one might assume — given limitations in contemporary technology with respect to deep conceptual understanding — that the conveyance of meaning to computers, in any form (such as that required to impart information about the meaning of contract terms) would not be possible. However, the insight from the approach just described is that it is sometimes possible to *functionally* convey meaning to computers, and that functional translation may be sufficient for particular computing purposes — including creating computable contract terms. In other words, given a particular concept, there may be a functional, computer-processable “translation” of the meaning of that word, if one can find a set of computer instructions, or data-relationships, that produce output that is consistent with what a person with a deep conceptual understanding would expect.

This strategy of association between meaning and computer instructions explains how computers react sensibly to make certain contract terms computable. The previous section described data-oriented options contracts, in which contract terms — such as option expiration dates — were expressed as structured computer data (e.g., <Option_Expiration_Date: 01/15/2018>). Upon encountering data labeled “Option_Expiration_Date”, a computer can be provided with a

DATABASE TECHNOLOGIES AND APPLICATIONS 599 (2006) (“Semantic information management [involves] capturing the precise meaning of data in agreed upon terms.”).

¹⁴² See HARPER, *supra* note 71, at 3 (“Programming languages are languages, a means of expressing computations in a form comprehensible to both people and machines.”).

deliberate set of instructions for how to sensibly react to that type of data in a way that effectuates what the parties intended. The computer can be, for example, instructed to initiate a comparison between the contract expiration date and the date upon which one party has attempted to execute the contract. If the expiration date has passed, the execution of the option can be disallowed — in accordance with the terms.

In this way, we can think of the meaning of an “expiration date” to have been functionally conveyed to the computer system. This permits the operation to be automated. It is not that the computer, upon encountering contract data labeled “Option_Expiration_Date” calls upon a deep conceptual meaning framework to understand the parties intended the contract to expire. Rather, the computer has been provided with a set of automatable computer instructions resulting in outputs that are consistent with what a person, with a conceptual understanding of the term, would have intended — an approximation of meaning based upon functional output.

In sum, the computable contracting approach permits computers to make automated, *prima-facie* assessments of conformance with certain contract terms. In order for the computer to make these assessments, the parties need to convey the meaning of the terms to the computers. One way that parties can convey the meaning of contract terms to computers, which lack deep conceptual abilities, is to create computer-based approximations resulting in the type of automated reactions to a particular contract term that a person who did have a deep conceptual understanding of that contract term’s meaning would expect.

Thus, parties wishing to create computable contract terms can sometimes (but not always) devise a set of computer instructions that act as functional translations of contract terms. This permits automated comparisons, which can be consistent with the limitations that the parties intended to convey, but which employ computer processes to lower certain transaction costs. In short, the reason that a computer system “knows” how to go about comparing contract terms to relevant data in a reasonable, but automated, way — despite lacking advanced cognitive processes — is that it has been told through a set of computer instructions how to make comparisons that effectuate the parties contractual intentions.

2. Captured Legal Assertions and Advanced Semantics

For explanatory purposes, the previous examples utilized contract terms with basic numerical comparisons. Computable contract terms

are, however, capable of more sophisticated expression. Computers can be told the meaning of the words directly, or they can be told *how* to find information necessary to determine what a word means. This point is illustrated by the earlier mentioned SIPX system — the Stanford-based project allowing for computer-processable copyright licensing terms. On SIPX and similar platforms, it is possible to design contract terms that are more expressively complex, yet still automatable.

For example, recall the earlier SIPX hypothetical in which Stanford University reduced its academic licensing fees by focusing electronic resources upon those students most likely to use them. The University might, for instance, link the access to law-related electronic resources — such as Hein Online’s law journal archive — to those students pursuing “law related courses of study.” How might conformance with a contract term as abstract as “law related courses of study” be made automatable? Such a criterion appears decidedly more complex and judgment-oriented than earlier examples involving comparisons of expiration dates or payments. One approach is to provide the computer system with data about *how* to decompose a concept as abstract as “law related courses of study” into a series of computer rules that are more actionable.

For example, the University might create a computer-processable definition for “law related courses of study” by leveraging its existing university database in which majors, courses of study, and courses are stored. In such an approach, the University might flag specific majors — such as traditional pre-law majors (e.g., political science, history, and economics) — and graduate law study — as “law related courses of study.” Both licensor and licensee might approve such a list. The labeling of data of particular majors and degrees is key because it will allow a computer to automatically translate the “law related courses of study” criterion into an enumerated and actionable list of majors and courses of study. In other words, when the computer needs to determine what “law related course of study” means for the purposes of conformance with contract terms, it now has been given the instructions to determine how to retrieve what this means. The computer need only access the university database to obtain a list of courses or majors that have been previously demarcated as meeting this requirement.

Once such a computer-processable definition has been provided, determination of conformance with the “law related courses of study” term requires only a short chain of automated analysis. Consider a student attempting to access a law journal through the system. The system might first identify that student’s major by referencing the

university's student information database. The computer can then determine, on its own, the meaning of "law related courses" by extracting the list of courses of study that have been demarcated as "law related." At this point, the computer can compare the student's major with the list of approved courses of study and majors to determine if there is a match. If the accessing student is a political science major, and that major is on the list of "law related courses," there is a match. In this way, *prima-facie* conformance with the terms as complex and discretionary as "law related course of study" can be automated. The key is to create a computer-processable decomposition for those terms and provide the computer with instructions for how to retrieve that meaning.

There are a few points to emphasize about this example. The first is simply to reiterate the main idea — that parties can, in certain circumstances, express conceptually sophisticated contract terms that are also automatable in assessment through the use of advanced computer semantic techniques. Thus, automated contractual assessments are not limited to the simple numerical comparisons used for illustration at the outset, but can encompass abstract or complex criteria. This is possible because parties can sometimes encode knowledge or information about the way concepts are related — in a form that the computer can usefully harness.¹⁴³

Second, it is sometimes possible to automate the assessment of contract terms that appear to require judgment. This seems at odds with this Article's previous assertions about the limits of contemporary technology in understanding abstract criteria and exercising professional level discretion. The key is understanding what this Article calls "captured legal assertions." The general idea behind a "captured legal assertion" is to have a qualified *person* — someone who has the ability to employ professional or subject matter judgment — and have that person apply that judgment to a set of facts. We might sometimes "capture" the *results* of that judgment as computer-processable data. This was illustrated by the way "legal course of study" was defined. In that example, we imagined that some competent person — such as a university administrator — used their professional and cognitive abilities to make a judgment as to what constituted law-related study (e.g., political science or law school

¹⁴³ Indeed, the provision of explicit data representing hierarchies and relationships among concepts or entities is an underlying concept behind "semantic web" technologies. See A.J. Gerber, A. Barnard & A.J. Van der Merwe, *Semantic Web Status Model*, NINTH WORLD CONFERENCE ON INTEGRATED DESIGN AND PROCESS TECHNOLOGY 473-74 (2006).

study) from the entirety of possible majors and degrees. The results of that judgment were then captured in a computer processable list (i.e., a database in which law-related majors were flagged).

Another example will help convey the concept of a captured legal assertion — the results of a legal judgment stored as a computer-processable data. Imagine a contract in which a term restricts the parties to using, for some purpose, only retirement plans that are “qualified” under section 401 of the tax code.¹⁴⁴ The term “qualified” — like “legal course of study” — seems to require some degree of professional judgment and legal analysis that might be beyond the capability of contemporary computer technology — and hence beyond the realm of automation. However, to get around this, an attorney or government official might assess the universe of retirement plans and create an enumerated list of plans that are deemed to be qualified under the provisions. Thus, the results of a judgment — such as whether a plan is “qualified” or not — has been made by a person, and the determinations are “captured” as data in a way that a computer could then usefully harness.

The insight is that in such circumstances, we are not requiring the computer *itself* to make a judgment about the application of discretion-oriented contract term — something likely beyond contemporary technological ability. Rather, parties can sometimes effectively give the computer access to the results of an earlier judgment made by a person who does have the capacity to make such an assessment, by expressing those results as data. Thus, when the computer encounters the phrase “legal course of study,” it is not itself considering different undergraduate majors to determine whether they have traditionally been considered law related. Rather, it is employing a previous judgment, made by a competent person, whose results have been captured in data. Once captured as data — as in an enumerated list or logical relationship — a computer can automatically translate such a discretion-based term into something more actionable (political science, history, economics, graduate law study). Once a person has “decomposed” an abstract or discretionary concept such as “legal course of study” and captured it in a concrete, data-oriented structured form, a computer can often harness this prior judgment in order to automate assessment with these criteria. Such an approach is not perfect, but it may be sufficient for certain contracting purposes.

Third, we can think of such computer-processable information as effectively enabling *new* types of contractual comparisons that would

¹⁴⁴ See, e.g., 26 U.S.C.A. § 401 (West 2010) (discussing qualified pension, profit-sharing, and stock bonus plans).

have been infeasible in the world of paper contracts and manual contract monitoring. While in principle, in the traditional contracting context, an employee charged with monitoring licensing compliance with such a complex licensing agreement could have performed the same suite of cross referencing of student major information and licensing restrictions, the cost would have been so great in terms of time, effort, and aggregation so as to render it impractical. Thus, the very creation of computer-processable versions of contract term information creates differences in *kind*, and not just degree, in the types of contracting analytics that are possible due to decreases in transaction costs enabled.¹⁴⁵

In sum, this subpart has primarily dealt with the conveyance of the meaning of contractual terms so that computer systems can functionally operationalize them in a way that is consistent with what the parties intended. The next subpart explores the ways in which computers can be provided with information upon which they can make automated assessments of conformance or non-conformance.

3. Providing a Computer with a Source of Information Relevant to Compliance or Performance

When parties determine compliance with contract terms, we can think of them as seeking information about what has or has not happened in the world. Information relevant to making such compliance or monitoring determinations is increasingly available in the form of computer-processable data. This is part of a larger trend in which corporate data, overall, is being created, collected, and stored in electronic form.¹⁴⁶ The important realization is that parties can sometimes take advantage of this available body of data to provide the final link in facilitating, automated *prima-facie* assessments of compliance with certain contract terms. Thus, contracting parties may be able to both specify their obligations in machine-processable form and provide the computers with data relevant to determining conformance with these terms. This part explores this final principle of computable contract terms.

¹⁴⁵ See *supra* note 143 and accompanying text.

¹⁴⁶ RALPH M. STAIR & GEORGE W. REYNOLDS, FUNDAMENTALS OF INFORMATION SYSTEMS 4-6 (2011) (citing the increasing use and storage of electronic data overall); Vasupradha Vasudevan & H.R. Rao, *E-Discovery and Health Care IT: An Investigation*, in ETHICAL ISSUES AND SECURITY MONITORING TRENDS IN GLOBAL HEALTHCARE 92 (Steven A. Brown ed., 2011) (estimating that 99% of business information is stored in electronic format).

To understand this point, it is helpful to restate the view of contract compliance articulated in the introduction. In that characterization, contract monitoring or assessment consisted of two broad phases: 1) the understanding of what has been contractually promised; and 2) the comparison of what has been promised, to what has or has not actually occurred. This framing makes sense because, at a high level, we can think of certain contract obligations as general expressions of a future state of affairs and the legal consequences that result if the promised state does not come to pass. Thus, while the previous part explained how to it is sometimes possible to translate what was promised into machine-understandable form (Phase 1), this section explains how a computer can be enabled to make automated, *prima-facie* assessments of conformance, by providing it with relevant data about the world (Phase 2) by which it can make comparisons.¹⁴⁷

Providing data to the computer that is relevant to conformance with terms is thus crucial to the computable contracting concept. Clearly for many types of contracts and the specified terms, the obligations will not be assessable or measurable using data.¹⁴⁸ Putting those relational or discretion-oriented terms or conditions aside — the subject of Part III — let us, instead, focus on the subset of contract terms which may be profitably assessed by data comparisons.

We have already seen several examples of this principle of providing computers with data relevant to determining conformance with terms. In the option expiration date example, the computer was provided with relevant data — the date at which a party attempted to execute the contract. In the currency contract example, the computer was provided with electronic records of payments between the parties. In the example of the geographically restricted Internet video streaming, the computer was provided with data as to the users approximate geographic location. In the SIPX content licensing example, the computer was provided with data about student majors and courses of study. The uniting principle behind these examples is that the parties made available to the computer data, in computer-processable

¹⁴⁷ See Nir Oren et al., *Towards a Formalisation of Electronic Contracting Environments*, in COORDINATION, ORGANIZATIONS, INSTITUTIONS AND NORMS IN AGENT SYSTEMS IV 156, 157-58 (Jomi Fred Hübner et al. eds., 2009).

¹⁴⁸ Computability of contract terms seems to presume a model in which legal outcomes can be constrained to be usefully processed by a computer system. Cf., e.g., Anthony D'Amato, *Can Any Legal Theory Constrain Any Judicial Decision?*, 43 U. MIAMI L. REV. 513, 513-15 (1989) (expressing skepticism about the degree to which legal outcomes may be constrained); Duncan Kennedy, *A Semiotics of Critique*, 22 CARDOZO L. REV. 1147, 1147-52 (2001) (demonstrating that statements that appear to be superficially constrained often mask implicit contextual and threshold decisions).

electronic form, information relevant to making *prima-facie* compliance determinations with contract terms, in a given instance.

Consider another example from a type of contract known as a “service level agreement” (SLA). In such an arrangement, the provider of some service contractually promises to its customers that the service will be up, running, and available at a pre-defined, measured level over a particular period of time.¹⁴⁹ Web-hosting firms (companies in the business of hosting the websites of other companies) and other providers of online services often enter into such agreements with their customers, whose businesses often depend upon a high level of availability for those services.¹⁵⁰ Thus, a web-host might agree that a customer’s website will be available for at least 99.5% of the total minutes over a 30 day period.¹⁵¹ Web-hosting companies typically keep records of website uptime and down-time already. To compute compliance with this obligation is a matter of providing the computer with access to this data, to permit automated comparisons of the actual number of minutes of downtime as recorded to the promised number of minutes.¹⁵²

The important point to draw from this example is that contracting parties can often harness business data that they previously collected for other reasons and repurpose this information to enable automated contract assessments. In the example just mentioned, the web-hosting firms were already collecting and storing data about their uptime for the purpose of operating their business. It was therefore less of an effort to establish a link between their contractual obligation whose criteria was dependent upon this uptime, because the necessary data already existed. Similarly, in the case of the SIPX, the university was already storing data about student class and major enrollment in structured, semantically-labeled form.

¹⁴⁹ See, e.g., AKHIL SAHAI & SVEN GRAUPNER, WEB SERVICES IN THE ENTERPRISE: CONCEPTS, STANDARDS, SOLUTIONS, AND MANAGEMENT 229 (2005) (describing the components of service level agreements); George S. Geis, *Business Outsourcing and the Agency Cost Problem*, 82 NOTRE DAME L. REV. 955, 986-87 (2007) (describing the features and content of service level agreements).

¹⁵⁰ See SAHAI & GRAUPNER, *supra* note 149, at 69.

¹⁵¹ See, e.g., *Google Apps Service Level Agreement*, GOOGLE APPS (last visited Sept. 28, 2012, 2:52 PM), <http://www.google.com/apps/intl/en/terms/sla.html> (promising an uptime percent for the service of 95%-99.9% and the consequences for falling below these terms).

¹⁵² For example, over a thirty-day period there are 43,200 minutes (60 minutes/hr * 24 hrs/day * 30 days). 99.5% availability (0.5% downtime) is no more than 216 minutes of the customer website being inaccessible out of the 43,200 minutes total over that period.

This parallels a trend in which the data that companies gather and store during the course of operating is becoming easier to access and repurpose electronically.¹⁵³ Although firms have been collecting and storing data relevant to their business for many years, until recently, from a practical perspective, it has been difficult to actually access that data to utilize for other purposes. However, emerging technologies such as “service oriented architectures” are increasingly making such corporate data practically more accessible for other applications, increasing the possibility of linking relevant data to create computable contract terms.¹⁵⁴

In sum, contracting parties can sometimes create computer-based contracts in which conformance with certain terms is *prima-facie* computable by the system itself. This applies not to all contract terms or subjects, but to some subset for which there are computer-based translations of the terms and relevant data. In some cases, to make a contract term computable is simply to automate an otherwise straightforward compliance comparison that would have previously been done manually in the traditional written contracting paradigm. In those cases, the significance is a substantial reduction in monitoring and compliance transaction costs where possible. In other cases, parties can create novel computable contract terms of sophisticated expressive range by leveraging far-flung data stored electronically, enabling automating contracting analyses that were practically infeasible in the traditional contracting model.

III. THEORY AND COMPUTABLE CONTRACTING

Let us summarize the central claim. In certain instances, it is possible to convey the meaning of contractual obligations to computers, and parties can automate the assessment of these terms. The key point: in order for computers to “understand” legal obligations, the creators of those obligations must deliberately reformulate them in forms that computers are able to process (e.g., structured data and computer instructions), dictated by technological constraints.¹⁵⁵ The fact that *any* automation is possible is somewhat

¹⁵³ See, e.g., T. Hau et al., *Where to Start with SOA: Criteria for Selecting SOA Projects*, in PROCEEDINGS OF THE 41ST ANNUAL HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES 314 (2008); *Service Oriented Architecture: Features and Benefits*, THE OPEN GROUP, http://www.opengroup.org/soa/source-book/soa/soa_features.htm (last visited Oct. 19, 2012).

¹⁵⁴ See Love & Genesereth, *supra* note 20, at 205-06.

¹⁵⁵ Parties often engage in such translations indirectly by transacting through third party electronic contracting platforms. See GARY P. SCHNEIDER, ELECTRONIC COMMERCE

surprising given assumptions about the limitations of current technology. The important qualification: the computable contracting approach works only when matched to an appropriate context. Primarily, this is the subset of contracting that is not expected to, in the ordinary case, necessitate abstract reasoning or legal analysis.

A. *Computable Contracting Through the Lens of Contracting Private Lawmaking*

This Part will argue that we can understand computable contracting through the theoretical lens of “contracting as private lawmaking.” To make a contract with computable terms requires a series of deliberate choices. These include, for example, the decision to express contract terms as computer processable data, and to hinge contract performance on criteria that are measurable and computable, over alternative criteria that may be more flexible, but less automatable. Choosing such computable criteria often involves tradeoffs. By consciously agreeing to accept attendant tradeoffs *ex-ante*, parties may be able to reorient their arrangements to make them computable. We can thus think of parties as deliberately architecting their contracting parties to *make them* computable so as to accommodate their business needs.

1. Contracting as Private Lawmaking

Contracting parties have been likened to “private lawmakers” and contracting as “private lawmaking.”¹⁵⁶ Under this view, contracting parties can be thought as “private legislators” who — through their contracts — create tailored laws to which only they themselves are bound.¹⁵⁷ This is reflective of a larger principle of U.S. contract law favoring choice and adaptability in contracting.¹⁵⁸ An overarching policy is to permit contracting parties the flexibility to tailor their contractual arrangements to meet their particular needs.¹⁵⁹ We can use

248-51 (9th ed. 2010).

¹⁵⁶ See Robert W. Gordon, *A New Role for Lawyers?: The Corporate Counselor After Enron*, 35 CONN. L. REV. 1185, 1208 (2003).

¹⁵⁷ We can think of contractual obligations as private laws in the sense that they have been designed by the parties (and not by a public legislature), and that they only bind the contracting parties themselves.

¹⁵⁸ LORD, *supra* note 6, § 1:1 (“[A] contract enables parties to . . . tailor their affairs according to their individual needs and interests . . .”).

¹⁵⁹ *Id.*

this framework of choice to understand the computability of certain contracting terms.

a. Choices in the Form in Which Contractual Obligations Are Expressed

This concept of choice in creating legal obligations is important to understanding computable contracting. As described in Part I, the key to having computers be able to process contractual obligations involves an intentional deviation from the traditional paradigm in which such obligations are expressed as written language. Rather, the creators of computable obligations must express their intentions in a unique form — structured data and computer instructions — for the express purpose of working within the constraints of contemporary computer technology. However, since contracting parties are private lawmakers, they have flexibility in how they choose to express their laws. This ability to tailor the reification of obligations as data can be explained by the policy favoring flexibility in formulating contracts to meet party needs.¹⁶⁰

The ability for parties as private lawmakers to express their obligations as data has theoretical implications. As I have written elsewhere, there are subsets of *public law* in which computer-based “translations” of legislation have emerged.¹⁶¹ For example, in the personal income tax context, private firms have created software that certain taxpayers can use to compute their tax liability. The tax code is promulgated in ordinary language, and we can think of such software as containing a computer-understandable “translation” of the meaning and logic of certain subsets of that statute. Firms that create such software seek to faithfully replicate the substance of the tax provisions to ensure that automated assessments of tax liability are accurate.

However, there is an implicit question of the authoritativeness of such software-based translations. Private firms that sell the software typically create such translations. In principle, a more authoritative

¹⁶⁰ See, e.g., Alan Schwartz, *The Default Rule Paradigm and the Limits of Contract Law*, 3 S. CAL. INTERDISC. L.J. 389 (1993) (discussing the conventional view, also known as the default rule paradigm, and that default rules should solve contracting problems).

¹⁶¹ See Surden, *Variable Determinacy*, *supra* note 115, at 1-5 (describing the federal income tax code as a set of legal criteria amenable to representation in a computer model); see also Intuit, *Importing into TurboTax*, TURBOTAX.COM, <http://turbotax.intuit.com/support/iq/Import/Importing-into-TurboTax/GEN12086.html> (last updated Sept. 14, 2012) (“You can import from many financial institutions, such as your payroll provider or brokerage firm . . .”).

computer-based translation would originate from the lawmaking body itself — Congress. For various economic and practical reasons, this may not be feasible, so instead, there are publically produced laws and privately produced computer-based “translations” of those laws.¹⁶² By contrast, in the computable contracting context, such a disparity between the creator of the legal obligation and its computer-oriented translation potentially disappears. Contracting parties are private lawmakers and lend the authoritative imprimatur to the computer-based versions of their legal obligations, since it is they who are producing them.

b. Choices in the Criteria, Terms, and Conditions by Which Compliance Is Measured

There is a second dimension of choice along which contracting parties, as private lawmakers, can tailor their contractual obligations to facilitate computability. Contracting parties, like public legislators who craft legal obligations, can elect among various criteria to use to measure conformance.¹⁶³ In general, those specifying legal obligations often can choose from criteria offering greater or lesser discretion in flexibility when applied. As described in Part I, to make contract terms computable, contracting parties generally have to elect well-defined criteria that can be measured automatically by reference to external data.

In the public law context, the traditional theoretical dichotomy has been between formally realizable “rules” — legal categories with bright line, measurable metrics (e.g., a law with 65 mph speed limit) or flexible legal “standards” (e.g., a law requiring safe driving).¹⁶⁴ The choice confronting contracting parties in electing computable contract criteria is similar but there is a theoretically distinct dimension than in the “rule” versus “standard” analysis. While it is true that contract terms that are computable most often resemble “rules” under this framework, the unique dimension is that technology meaningfully

¹⁶² This is not a unique feature to computable lawmaking. The same phenomenon occurs nearly any time a private attorney gives an interpretation of a law. In that case, a private party is relying on the non-authoritative interpretation, by a private attorney, of a public law. There will often be some uncertainty. However, for an example of a tax calculator originating from the government, rather than from a private firm, see *Tax Calculators, Tables and Rates*, STATE OF CALIFORNIA FRANCHISE TAX BOARD, https://www.ftb.ca.gov/online/Tax_Calculator/ (last visited Nov 2, 2012).

¹⁶³ See Louis Kaplow, *Rules Versus Standards: An Economic Analysis*, 42 DUKE L.J. 557, 557 (1992).

¹⁶⁴ See Duncan Kennedy, *Form and Substance in Private Law Adjudication*, 89 HARV. L. REV. 1685, 1685-86 (1976).

increases the “realizability” of such rule-based criteria.¹⁶⁵ In the computable contracting context, the parties are choosing to deliberately link their rule-based criteria to a source of data relevant to making such a determination, often nearly instantly. By contrast, in the public law context, a lawmaker could in principle promulgate a law with a rule-like criterion that is realizable in theory, but for which the data for making such realizations is, as a practical matter, unavailable. The important point is that in the computable contracting context, the explicit linking of contract criterion to data upon which such criterion can be decided meaningfully accentuates the decisional divide between legal standards and rules.

In sum, because contracting parties are like private lawmakers, largely in control of both the *substance* and the *form* of their contractual obligations, there is the freedom to design them so as to be computable. Parties have the flexibility to explicitly agree in a threshold agreement to make their contractual obligations dependent upon any reasonable criterion, including measurable, automatable data. These choices are important because to make contractual terms computable and take advantage of any attendant benefits, the contractual terms have to be deliberately architected so as to work within the constraints of computer technology.

c. Choices in How Contracting Parties Manage Uncertainty

Contracting parties can also make choices that affect the level of legal uncertainty concerning their arrangements. We can think about legal uncertainty in terms of predictions about future decisions from authoritative legal decision-makers (such as judges).¹⁶⁶ In this view, there is more legal certainty when we expect an *ex-ante* prediction about a legal outcome to match an *ex-post* determination by an authority, and less legal certainty when the convergence between prediction and ultimate outcome is unreliable.

Parties who create computable contract terms likely want automated assessments of conformance with contract terms to be in line with any future judgment of compliance made by a judge, should it be litigated. To the extent that such automated assessments are consistent with later decisions, there will be greater confidence in computable contracts as a legal tool that the parties can rely upon. Conversely, to

¹⁶⁵ See *id.* at 1687 (describing formal realizability and the degree to which a law can be definitively decided by reference to external metrics).

¹⁶⁶ See Oliver Wendell Holmes, *The Path of the Law*, reprinted in 78 B.U. L. REV. 699, 700 (1998) (explaining the view of legal analysis as predicting how an authority will determine how the law applies to a given scenario).

the extent that there is significant divergence between automated assessments, and a typical judgment (e.g., judges are constantly “second guessing” and “correcting” automated assessments *ex-post*), automated assessments will be less reliable.

To capture this potential for divergence, this Article has used the phrase “prima-facie” to indicate the legal tentativeness associated with automated assessments. Said differently, the result of the mechanical comparisons of computable contracting, may not, in some instances, reflect ultimate authoritative judgments by a legal decision-maker, such as a judge, at the conclusion of litigation. In this usage, “prima-facie” can be understood as an expression of a degree of legal uncertainty as to the conclusiveness of an automated assessment. The computable contracting approach employs rather straightforward, deductive methods of determining conformance with contract terms — automated comparison between well-specified terms, and data. Such a formal, rules-based approach does not take into account a wider range of considerations upon which more holistic, nuanced assessments might be made.¹⁶⁷

We could imagine a judge during litigation coming to a result different than an automated *prima-facie* analysis that was based upon data comparisons. Such a divergence might occur if the judge takes into account a broader set of facts that, on the merits, made the data-driven result appear inequitable. If a computer comes to one assessment as to conformance with contract terms, and a judge comes to a different one, there is divergence between prediction and outcome. If it were the case that *prima-facie* assessments were routinely being challenged by dissatisfied parties *ex-post*, and routinely changed by decision-makers *ex-post*, the predictive power of *automated prima-facie* assessments would lose their force and efficiencies would be lost. The benefit of automated *prima-facie* comparisons between contracting criteria and data is precisely that *prima-facie* results are believed to only very infrequently diverge from an authoritative determination by a decision-maker.

However, parties can deliberately calibrate the level of legal uncertainty associated with a given contract. They can make choices in their contracting structure to *increase* the relative legal certainty of their automated *prima-facie* determinations. Contracting parties, as private legislators, can choose to engage in what I have called

¹⁶⁷ See, e.g., Frederick Schauer, *Formalism*, 97 YALE L.J. 509, 510 (1988) (noting that formalist decision-making involves “screening off from a decisionmaker factors that a sensitive decisionmaker would otherwise take into account”).

elsewhere “deliberate forbearance” of *ex-post* corrections.¹⁶⁸ This involves creating a contractual framework in which *prima-facie* assessments control, and in which the possibility of *ex-post* corrections for meritorious exceptions to *prima-facie* assessments is reduced. Parties can thus decrease the likelihood of divergences between *ex-ante* automated assessments, and *ex-post* authoritative judgments.

One way to do this is to specify, in a threshold agreement, the deliberate and joint intention of the parties to be bound to *prima-facie* results. In other words, the contract explicitly stipulates that the parties are voluntarily forbearing from the possibility that a legal-decisionmaker should make an *ex-post* correction, even when an automated, *prima-facie* assessment produces a factually incorrect or unfair result. The parties signal that they are “tying their hands” for the sake of *ex-ante* efficiency and to increase their ability to *rely upon* automated *prima-facie* assessments. By indicating, contractually, the parties’ desire to have the *prima-facie*, automated assessments be authoritative, they are reducing the likelihood of an *ex-post* divergence from the automated assessment of conformance. They are, in effect, architecting their contractual framework so as to increase the legal certainty of their automated assessments by agreeing, up-front, to be held to the *prima-facie* automated conclusions.

The earlier example of geographic detection illustrated the point of deliberate forbearance in *ex-post* corrections. There, the parties wanted geographic location to be a condition of contracting, but they preferred *automated* detection of location for its efficiency. The problem is that the technology for detecting location is occasionally incorrect relative to *actual* location. Thus, the parties explicitly stipulated to adhere to the automated results even when factually incorrect relative to a user’s actual location. They did this to increase the legal certainty of the automated assessments, so that both parties could rely on automation, without being concerned that a later legal decision-maker would overturn the *prima-facie* analysis in light of a holistic assessment. By signaling, *ex-ante*, a deliberate acceptance of automated *prima-facie* conclusions, the parties were explicitly forbearing in accepting the occasional incorrect automated assessment for the sake of efficiency and certainty.

There are a few points to clarify. Such a policy of forbearance requires the judiciary to honor such forbearance requests *ex-post*.¹⁶⁹

¹⁶⁸ See Surden, *Variable Determinacy*, *supra* note 115, at 70-75.

¹⁶⁹ See, e.g., LORD, *supra* note 6, § 31:5 (“If freedom of contract means anything, it means that parties may make even foolish bargains and should be held to the terms of their agreements . . . [The court’s role] is not to redistribute these risks and

The temptation is for one party with a particularly meritorious *ex-post* set of facts, who had previously agreed to a policy of forbearance for efficiency purposes, to opportunistically undo that *ex-ante* agreement. When it appears that the parties had equal bargaining power, and the *ex-ante* decision to engage in forbearance was well understood, courts should respect these decisions and refuse *ex-post* corrections.¹⁷⁰ As such, it is worth reemphasizing that this Article has only discussed *commercial* contracting involving sophisticated firms. Different considerations apply in the *consumer* context, and this Article should not be read to imply anything about consumer contracting.

Second, to say that there is deliberate forbearance is not to say that significant *ex-post* exceptions are not necessarily provided for. Indeed, in the example of financial contracts, the parties use threshold agreements to set up *ex-post* processes to deal with *serious* issues that go awry. As described earlier, parties creating data-oriented and computable contracts need to prepare for and specify a non-automated process for handling significant exceptions or unanticipated circumstances. The expectation is that exceptions will be rare, and the threshold for receiving *ex-post* judicial correction will be much higher.

Third, such agreements represent a tradeoff between *ex-post* correction, and an *ex-ante* efficiency of *automated* assessments. This is similar to the type of deliberation that public lawmakers engage in — electing under and over inclusive “legal rules” at the expense of more accurate, but less administrable, “legal standards.” However, this Article is not *advocating* for the adoption of *ex-ante* efficiency and computable terms over other considerations. Rather, it is noting that there is a genuine tradeoff in terms of *ex-post* flexibility, accuracy, and ability to take into account other valuable and relevant information.

In sum, parties can make deliberate contracting choices to increase the *ex-ante* legal certainty of their automated assessments. They can partially architect the degree to which decision-makers might entertain exceptions *ex-post*. By opting to reduce the possibility of *ex-post* correction, the parties can actually increase the usefulness of their automated assessments. Parties who create computable contracts want *prima-facie* assessments to be accurate reflections of an ultimate authoritative decision so that they can rely upon the speed by which computers can compute these assessments. By opting into the

opportunities as [it sees] fit, but to enforce the allocation the parties have agreed upon it.”).

¹⁷⁰ It is important to qualify that these ideas are only geared toward the *commercial* context involving sophisticated firms, and say nothing about an outcome in the *consumer* context.

computable paradigm, they are consciously tolerating a tradeoff in *ex-post* flexibility for *ex-ante* efficiency. Parties can communicate to a legal decision-maker, later in time, their deliberate agreement, *ex-ante*, to be bound by the automated *prima-facie* assessments, even if such conclusions are occasionally incorrect on the merits.

B. Limitations of Computable Contract Terms

The computable contracting approach is limited to a subset of contracting scenarios. Because the framework uses automated comparisons to assess compliance, it is implicitly limited to contexts in which such straightforward evaluations are sufficiently useful to meet the contracting needs of the parties. For many complex-contracting arrangements, such a model may be inadequate. However, the more surprising point is that some contracting scenarios *actually are* close enough to this paradigm so as to render the computable model valuable. Some represent domains of commercial significance such as the financial contracts described previously. This part will delineate both limits and core characteristics of the computable approach.

1. Features Common to Computable Contracting Contexts

The earlier computable contracting examples share some common features. First, the “topic” of those contracts concerned readily identifiable subjects. For example, option contracts and option expiration dates represent specific, identifiable subjects. By contrast, other contracts routinely reference non-identifiable, or highly abstract topics (e.g., “any agency of competent jurisdiction”).¹⁷¹ Second, the contract terms at issue referenced well-defined and measurable properties of those contracting subjects. In other words, the automated assessments turned on distinct attributes whose *value* could be clearly and non-controversially identified. For example, a “major” is a measurable property of a given student, and an expiration date is a measurable property of a particular option contract. Generalizing, to

¹⁷¹ See, e.g., THE BEAR STEARNS COMPANIES INC. & J.P. MORGAN CHASE & CO., AGREEMENT AND PLAN OF MERGER BY AND BETWEEN THE BEAR STEARNS COMPANIES INC. AND J.P. MORGAN CHASE & CO. 33 (2008), available at <http://graphics8.nytimes.com/images/blogs/dealbook/BSmerger.pdf>. An agreement as complex and necessarily abstract as the Bear Stearns/J.P. Morgan merger agreement, in which party risks and understandings constitute a significant portion of the document, is an example of a highly abstract agreement that would not likely be amenable to the computable contracting approach.

the extent that a contract references identifiable subjects (e.g., particular books, students) and measurable properties of those subjects (e.g., geographic location, date, major), it is more likely that such a contract term could be made computable.

Second, the contracts involved terms for which the anticipated frequency of meritorious exceptions to general rules was expected to be relatively low. For example, it is precisely because the application of the “option expiration date” criteria was expected to be non-controversial in the ordinary case — comparing one date to another — that it made sense to *automate* such a comparison. Were it otherwise — if date comparisons routinely brought uncertainty and contestability — then the automated comparisons would be little more than the starting point for *ex-post* contestation of *prima-facie* results. This would undermine the point of the process and the economies of scale due to efficiency. Thus, contracting criteria likely to involve *ex-post* contestability are poor candidates for computability.

2. Limitations of Computable Contracting

Relational contracting arrangements can be problematic for the computable approach. Charles Goetz and Robert Scott have written that, “[a] contract is relational to the extent that the parties are incapable of reducing important terms of the arrangement to well-defined obligations.”¹⁷² Thus, contracting parties may sometimes specify contract terms at a high level of generality to allow for flexibility and discretion in assessing future conformance.¹⁷³ Flexible criteria may be preferable in scenarios likely to involve unpredictable facts, when performance is best evaluated holistically, or when up-front specification is possible but costly.¹⁷⁴ Because the computable approach involves automated comparisons between well-specified criteria and relevant data — and not abstract generalizations — if a scenario requires abstraction, *ex-post* flexibility in assessing facts, or the exercise of professional judgment— it would be inapt from a computable contracting standpoint.¹⁷⁵ Similarly, the computable

¹⁷² Charles J. Goetz & Robert E. Scott, *Principles of Relational Contracts*, 67 VA. L. REV. 1089, 1091 (1981).

¹⁷³ See *id.* at 1090-92.

¹⁷⁴ See *id.* For a related idea in the rules versus standards context, see Kaplow, *supra* note 163, at 559 (describing that the essential difference between a “rule” and a “standard” is the degree to which a law has its content specified *ex-ante* or *ex-post*).

¹⁷⁵ See, e.g., ZHONGZHI SHI, *ADVANCED ARTIFICIAL INTELLIGENCE 4-5* (2011) (noting that contemporary artificial intelligence systems cannot perform even at the level of a three year old child in many routine tasks).

model presumes terms whose content are both capable of being captured and measured using data. On the whole, abstract or flexible contract terms are less amenable to such measurement.

There is another limit. The computable contracting model is implicitly premised upon the primacy of the formal obligations that have been communicated to the computer system. However, it is important to distinguish between formally specified obligations and other external considerations that may be necessary to determine conformance with legal obligations.¹⁷⁶ In some contexts, the information specified in a formal contract document may be the primary source for assessing obligations.¹⁷⁷ In others, considerations external to the document — for example, business norms, previous transactions between the parties, *ex-post* assessment, or regulatory frameworks — may be more relevant to determining conformance.¹⁷⁸ The significance of such external factors may vary depending upon a given contractual arrangement.¹⁷⁹ For example, Gillian Hadfield has identified contexts where factors external to the four corners of a reified document are significant in determining conformance.¹⁸⁰ To the extent that there are significant external factors that are relevant to determining contract compliance, but that are not available to the computer system, then automated comparisons will not be useful indicators of actual conformance.

Computable contracting is also implicitly premised on a model in which computers “assess” or “determine” conformance. However, ever since the Legal Realist era, it has been understood that the application of legal criteria to facts often masks an underlying process of *ex-post* policy-balancing or the resolution of competing, but conflicting, interests among societal actors.¹⁸¹ Thus, legal determinations, in some cases, look less like discerning “objectively right legal answers” and

¹⁷⁶ See Ian R. Macneil, *Contracts: Adjustment of Long-Term Economic Relations Under Classical, Neoclassical, and Relational Contract Law*, 72 NW. U. L. REV. 854, 863-65 (1977) [hereinafter *Contracts: Adjustment*].

¹⁷⁷ See Robert E. Scott, *A Theory of Self-Enforcing Indefinite Agreements*, 103 COLUM. L. REV. 1641, 1647 (2003).

¹⁷⁸ See *id.* at 1645.

¹⁷⁹ See, e.g., Ian R. Macneil, *Relational Contract Theory: Challenges and Queries*, 94 NW. U. L. REV. 877, 896 (2000) [hereinafter *Relational Contract Theory*] (distinguishing between transactions that are more discrete and self-contained, and those more strongly linked to a larger iterative, social and commercial context).

¹⁸⁰ See Hadfield, *supra* note 6, at 992.

¹⁸¹ See, e.g., Gary Peller, *The Metaphysics of American Law*, 73 CALIF. L. REV. 1151, 1152-54, 1242 (1985) (noting that legal realists exposed that many superficially formal legal decisions actually involved *ex-post* policy balancing).

more like policy-making or dispute resolution.¹⁸² Thus, some contracting criteria will be set up with the expectation that legal authorities will not be finding objectively “right answers,” but rather balancing reasonable, though conflicting, interests on the part of the contracting parties or other actors that are best analyzed *ex-post*.¹⁸³ Thus, to the extent that contract terms tend toward the latter — in which the contractual exchange would benefit from an *ex-post* weighing of competing interests and the parties elect judgment-oriented contractual terms to reflect this recognition — this will, again, be less amenable to the computable approach

A final limitation concerns contracting contexts involving considerable legal uncertainty. The computable model presumes that *prima-facie* legal assessments will be usefully determined by comparing data. However, in particular contexts, there may be significant uncertainty about governing laws or relevant facts. Indeed, instances in which there is considerable legal or factual uncertainty are those in which lawyers are often brought in for their analytical expertise. Thus, the question may arise: How does the computable contracting approach manage the automation of contracting in contexts of legal uncertainty or necessitating professional judgment? The simple answer: it does not. To reemphasize the major point, that is not the subset of contracting that the computable approach is for. Rather, the focus is on a different subset of contracting expected to be relatively more determinate and which useful *prima-facie* legal conclusions are reasonably ascertained by comparing criteria to data.

C. Producing Intelligent Results Without Intelligence

It should be apparent that the computable contracting approach described herein does not involve replicating, in computer systems, cognitive processes exhibited by attorneys. Rather, it is based upon creating computer-based rules that lead to reasonable, *prima-facie* assessments in appropriate contracting contexts. This raises a question: how is it that, if computers are unable to exhibit the advanced cognitive, problem-solving, and professional judgment abilities that are routinely evinced by attorneys, can they can produce useful automated *prima-facie* legal assessments at all? This part explains the underlying principle permitting computable contract terms: “non-intelligent” computers can sometimes be programmed to

¹⁸² See Surden, *Variable Determinacy*, *supra* note 115, at 72.

¹⁸³ See, e.g., Macneil, *Contracts: Adjustment*, *supra* note 176, at 866 (describing the role of architects in construction contracts in balancing interests).

produce, by various rules-based or statistical processes, what would be considered “intelligent results.” Once this principle is understood, it helps explain both the possibility and constraints of the approach.

1. Functional View of Intelligent Results

In many contexts, it is possible for computers to produce *results* that would be considered “intelligent” even while lacking the higher-order cognitive skills associated with people. To understand this, it is helpful to distinguish a common view. In a view that is closely aligned with the public imagination, computers are thought to only be able to deal with abstract problems — such as legal analysis — if they are able to replicate in computer-form the high-level cognitive abilities or abstract reasoning skills of people.¹⁸⁴ “Artificially intelligent systems,” under this view, must replicate and instantiate to varying degrees the thinking facilities of humans, such as the ability to engage in abstract thought, carry on arbitrary, intelligent conversations, read arbitrary texts, or understand concepts at a deeper level.¹⁸⁵ However, it is well understood in the field that artificial intelligence (AI) research has not yet produced, and is not necessarily near producing, computers with artificial, human-level cognition.¹⁸⁶

However, under an alternative view, we might evaluate a system’s “intelligence” primarily based upon the quality of the output produced.¹⁸⁷ If a computer system produces results that most people would consider accurate, helpful, and useful, this approach would consider the system to be “intelligent,” even if the “output” came about through processes that do not approach actual human cognition.¹⁸⁸ The insight is that the first view contains an overbroad assumption — if a task *appears* to require human-level cognition and intelligence — such as legal analysis — then only computer systems that replicate such cognitive processes will be able to perform it. However, if one takes the overbroad view, one is likely to overlook a subset of contexts that routinely demand human cognition as a general matter (such as legal analysis), but may not require cognition in every instance (e.g., straightforward contract comparisons).

Most successful contemporary AI systems in use work by producing what appear to be “intelligent” results on the basis of non-cognitive

¹⁸⁴ See RUSSELL & NORVIG, *supra* note 10, at 1-5.

¹⁸⁵ See *id.* at 2-3.

¹⁸⁶ See *id.* at 27.

¹⁸⁷ See *id.* at 4-5.

¹⁸⁸ See *id.* at 26-27.

processes.¹⁸⁹ For example, modern airplane auto-pilot systems are capable of landing airplanes in difficult conditions such as fog. There, they often meet or exceed human performance, even though such systems do not have a meaningful understanding of abstract concepts like “airplanes,” “runways,” “fog,” or “airports.”¹⁹⁰ Following the latter view, we can consider a machine to be successfully “intelligent” if it produces what people would consider “accurate” or “useful” results, meaning results that approach or exceed that which would have been produced by a person performing the same task.¹⁹¹ Under this position, we can use a similarly situated person, and their expected results, as a comparator and metric for gauging good, automated outcomes. The key insight is that there is a class of tasks that superficially appear to require intelligence or cognition but for which computers can perform useful activities that approach or exceed the output of people through the use of computer models based upon rules or statistics.

For example, playing chess or answering trivia questions seem to call upon the higher-order cognitive, abstract reasoning, and problem solving skills in human players.¹⁹² However, IBM has created computers that can produce output in these arenas that meet or exceed human players using rules, data, and statistics. They do not replicate human cognitive processes.¹⁹³ Similarly, translation appears to be a task deeply connected to the human understanding of the meaning of language. However, Google Inc. has created computer-based translation systems able to produce surprisingly good results without replicating human-level linguistic abilities.¹⁹⁴ This approach to automated translation is mostly statistical in nature. “Google Translate” — and other similar approaches — work in part by

¹⁸⁹ See *id.* at 28-29.

¹⁹⁰ See, e.g., BARNES WARNOCK McCORMICK & M. P. PAPADAKIS, AIRCRAFT ACCIDENT RECONSTRUCTION AND LITIGATION 128-29 (2003) (“[W]hat has occurred is that some airplanes are certified to autoland essentially blind! As a result of using these systems, the pilot has been taken out of the control loop and has become a system monitor or a computer manager . . . Such systems exist today that allow landings in fog.”).

¹⁹¹ See RUSSELL & NORVIG, *supra* note 10, at 26-28.

¹⁹² See Aditya Kalyanpur et al., *Leveraging Community-Built Knowledge for Type Coercion in Question Answering*, in THE SEMANTIC WEB — ISWC 2011 145 (2011); *IBM’s Watson Computer Takes the Jeopardy! Challenge*, IBM CORPORATION, <http://www-304.ibm.com/businesscenter/cpe/html0/211168.html> (last visited Mar. 6, 2012).

¹⁹³ See EKATERINA OVCHINNIKOVA, INTEGRATION OF WORLD KNOWLEDGE FOR NATURAL LANGUAGE UNDERSTANDING 215-20 (2012).

¹⁹⁴ See CLAUDE SAMMUT & GEOFFREY I. WEBB, ENCYCLOPEDIA OF MACHINE LEARNING 913 (2011).

leveraging vast amounts of documents that have been previously translated by people from one language to another.¹⁹⁵ For example, the United Nations frequently translates official documents into multiple languages using professional translators. This corpus of translated documents has become available in electronic form.¹⁹⁶ Such systems analyze these documents to create sophisticated statistical models of the likely meaning of various phrases and are able to produce surprisingly good translations — simply by using probabilistic models.¹⁹⁷

The important point is that, for certain types of tasks, it is possible for contemporary computer systems to produce intelligent-seeming results by relying upon rules-based and statistical approximations, and not upon automated processes replicating human-order cognition. This more nuanced view is key to understanding both the possibilities and limitations of computable contracts. In most of the cases in which computable contracting is possible, the computers are simply engaged in a class of comparisons with outcomes that can be determined by processes that do not require higher-order cognitive or legal analytical skills.

IV. THE IMPLICATIONS OF COMPUTABLE CONTRACTING

Because the range of benefits of machine-processable obligations may not initially be obvious, this part begins by considering in more detail some of the more nuanced advantages of computable contracting. The section that follows will address some of the theoretical implications.

A. *Perceived Benefits of Computable Contracting*

While there are several perceived benefits to formulating contractual obligations in data-oriented, machine-processable form, the advantages of such a data-focused reformulation on legal analytics are somewhat subtle. These benefits include the ability to: 1) reduce transaction costs in creating, monitoring, and reacting to obligations; 2) use new properties for analyzing contractual arrangement that are only possible when they exist in machine-processable form; and 3)

¹⁹⁵ See *Inside Google Translate*, GOOGLE, <http://translate.google.com/about/> (last visited Mar. 6, 2012).

¹⁹⁶ See SAMMUT & WEBB, *supra* note 194, at 913.

¹⁹⁷ See *id.*

enable autonomous, computer-to-computer, contracting. I will examine each in turn.

1. Reduced Transaction Costs in Contracting

Computer-processable contractual obligations can generate economic efficiencies when deployed appropriately.¹⁹⁸ Firms can employ such computable contracts to reduce the transaction costs of creating and resolving those contractual criteria and conditions amenable to computability. In the traditional paradigm, there are often significant costs associated with bargaining and assessment/enforcement of contract terms.¹⁹⁹ Creating data-oriented contracts in which the terms are selectable and adjustable dynamically, and computable contract in which compliance with terms can be assessed on a *prima-facie* basis, can reduce transaction costs. The financial industry and the computable, standardized financial contracts exemplify this dynamic. These contracts contain a number of relatively routine terms and conditions — such as the price, quantity, and expiration date of agreements to buy and sell financial instruments. Such terms are relatively straightforward in the sense that — in terms of legal risk and uncertainty — we imagine that these are not typically contestable in a considerable percentage of cases.²⁰⁰

Assessing when and how these contracts should apply involves the examination and comparison of their various terms and conditions. The automation of comparisons that are not legally complex or contested can reduce transaction costs.²⁰¹ We can see this by reference to a metric: the transaction costs incurred by an employee, in the pre-electronic era, assigned to evaluate such contracts as applied. This manual process would have presumably involved the reading and understanding of key terms by the employee and the acquisition of information to make decisions about when and how they should be implemented. It is true that, even in manual terms, the comparison of straightforward criteria such as date and amounts are relatively slight for the trained employee. However, computers are able to execute

¹⁹⁸ As will be shortly discussed, it is not always appropriate or economically efficient to create computer processable versions of contractual obligations.

¹⁹⁹ See, e.g., Gordon, *supra* note 29, at 1613 (“[A]t times bargaining may be exceedingly expensive or it may be impractical to obtain enforcement . . .”).

²⁰⁰ VAN VLIET, *supra* note 124, at 148.

²⁰¹ See DEROSA, *supra* note 14, at 20 (“Electronic trading has greatly enhanced the price discovery process in foreign exchange. A consequence is a great narrowing of the width of the bid-ask spread . . .”).

these same comparisons at rates that are significantly faster than employees.²⁰² These slight transaction costs become significant when they are, as in the financial industry, multiplied across many such contracts at any given time. Economies can be gained by creating computable versions for relatively routine assessments of legal criteria across multiple contracts.²⁰³ However, the creation of computable contracts has its own costs in terms of technological infrastructure, so under many scenarios it may not be efficient to do so, even when possible.²⁰⁴

2. New Analytical Properties Gained from Computability

A subtle but perhaps more interesting benefit of computable contracts is found in the novel analytical properties that emerge once contractual obligations are represented in computable form. The properties include, for example, the ability of a firm to compare their outstanding legal obligations to one another to detect contradictions.²⁰⁵ In other instances, once legal obligations are represented in terms of data, they can serve as “inputs” to be analyzed within the existing systems that many firms use to manage their operations.²⁰⁶ This sub-part will illustrate the principle that, once legal obligations are formed in terms of structured data that has been given machine-processable meaning, they can be compared, processed, summarized, and manipulated by computer systems, just like other,

²⁰² KHANNA, *supra* note 16 at 82 (noting that manual entry of contract and trade information can create delays and backlogs).

²⁰³ A parallel exemplar of efficiency gain has been seen outside of contracting, in the personal income tax realm. There, portions of the tax code have been rendered into computer-rules through popular programs such as TurboTax. A significant amount of the data for making routine financial assessments about items such as financial trades is available in the form of semantically labeled data. See Surden, *Variable Determinacy*, *supra* note 115, at 70-75 (describing the federal income tax code as a set of legal criteria amenable to representation in a computer model).

²⁰⁴ These costs include the technology infrastructure (i.e., databases and systems) to support computability, as well as the costs involved in “translating” the obligations into processable form. Thus, computable contracts appear to be most efficient when there are large numbers of standardized transactions that justify the infrastructure costs. In other words, one reason that computable contracts may not be widely applicable to many contracting scenarios is that the cost associated with creating computable versions may not be justified for specific, one-off contractual arrangements.

²⁰⁵ See Breuker et al., *supra* note 114, at 38-40 (discussing technologies to detect inconsistencies and contradictions in legal obligations).

²⁰⁶ See SAXENA, *supra* note 3, at 25-30.

more familiar pieces of corporate data (e.g., accounting and revenue data).²⁰⁷

The earlier SIPX example illustrated the new capabilities that emerge when contractual obligations are expressly created in a data-oriented form. These abilities are best understood by contrast against their written-language, “traditional contract” counterparts. As noted, when contractual meaning and intentions are expressed in the conventional form, in which the terms, conditions, and intentions are expressed in descriptive language intended to be read by people, the underlying contractual meaning is effectively inaccessible to a computer system. In the SIPX example, Stanford *translated* its content-licensing agreements with academic publishers into a data- and rules-oriented, machine-processable form. Because of this data-oriented expression of contractual terms, they were able to compare and contrast the meaning of their agreements across multiple, disparate licenses, using the processing and analytic abilities of computers. When the meaning of the contract was expressed in terms of written, descriptive sentences readable by the contracting parties, the transaction costs involved effectively prohibited such comparisons for anything beyond a few agreements.

This ability to computationally compare the *substantive content* of contractual licenses is illustrated by the “duplicative license” scenario. In this example, consider several different university units (e.g., libraries, academic departments) that had separately negotiated licensing agreements for academic materials. In several cases, these agreements overlapped, conferring duplicative licenses. In the traditional written-language contract context, conditions such as duplications or contradictions among legal obligations are difficult to detect. These can become lost and obscured among the contracts and licenses located in the filing cabinets and computers of those who negotiated them. However, once these legal obligations are made explicit and represented in terms of data, they are no longer effectively lost in the paper. Rather, computers can efficiently find and compare these legal obligations as data objects themselves and detect such duplication.

For example, we could imagine two pieces of data: 1) “All University students are licensed to engineering publications from Elsevier under the Library Licensing”; and 2) “All Engineering Students are licensed to Engineering publications from Elsevier”.

²⁰⁷ See IVAN MARKOVIC, SEMANTIC BUSINESS PROCESS MODELING 149-150 (2009) (describing how formal expression of business policies and rules allows for their verification against actual business processes).

Framed in data terms, a computer could easily detect such a duplication. In the traditional paper context, the same obligation could become lost among the many agreements with different publishers, for different types of academic materials, conducted by the separate units on campus. The important point is that once legal obligations are represented as data in themselves, they are capable of being compared and analyzed computationally in new and useful ways.²⁰⁸

Similarly, once contractual obligations become represented in terms of data, they can be used as “inputs” to be processed and can interface with other computer systems.²⁰⁹ Such computer systems can use this “legal data” to process the impact of *legal obligations* upon the commercial processes these systems manage. For example, large firms routinely employ Enterprise Resource Planning (ERP) computer systems in order to organize, plan, and manage all aspects of operating a firm.²¹⁰ These software systems store, organize, and model the data that represents a firm’s operations. Such an ERP system might include, for example, data about suppliers of materials to build products, or computer models of processes by which products are manufactured.

In modern ERP systems, nearly any aspect of the operation of a firm is capable of being stored and represented in terms of data and rules, from coordinating supply chains to the management of a firm’s employees and human resources. Since an important part of any firm’s operations are its contractual and other legal obligations, it would be useful to be able to similarly model such legal obligations. A firm could then use them as inputs to other computer systems — such as those that manage manufacturing or human resources. Used as inputs, the legal obligations could inform or constrain relevant decisions that might be affected by them, such as manufacturing, purchasing, or regulatory compliance. The representation of legal obligations in terms of data permits existing legal obligations to be analyzed alongside and in conjunction with other types of commercial data.

It is perhaps easier to consider this point regarding legal obligations by reference to an analogous category of “non-legal” commercial rules. Many sophisticated firms use computers to implement “business policies,” which are more or less computerized rules used to guide, constrain, and ensure that these automated systems process data in a

²⁰⁸ SAXENA, *supra* note 3, at 23-30 (illustrating different ways in which computer systems can analyze and manage contract terms expressed as data).

²⁰⁹ JENS KAPPAUF, BERND LAUTERBACH & MATTHIAS KOCH, LOGISTIC CORE OPERATIONS WITH SAP: PROCUREMENT, PRODUCTION AND DISTRIBUTION LOGISTICS 67-68 (2011).

²¹⁰ MARKOVIC, *supra* note 207, at 70-80.

way that is consistent with company goals or objectives.²¹¹ Thus, for example, a particular business rule may constrain manufacturing output of a certain good based upon certain stock levels of supplies that are obtainable.²¹² This description of business rules and policies — which are already routinely used in the corporate setting to automate particular commercial operations — resembles this Article’s characterization of computable contractual terms and conditions. In a similar manner, the firm’s *contractual legal* obligations, if represented in terms of data and rules, can form a set of inputs that can constrain or inform particular decisions involving licensing, manufacturing, purchasing, payments, or human resources based upon legal commitments.

For example, manufacturing firms are better able to manage legal risk by representing their legal obligations computationally. Imagine a firm in which two salespeople have contracted to supply one-hundred widgets to two different customers on the same date (i.e., two-hundred widgets total). Assume further that the company only has the manufacturing capacity to produce one-hundred widgets by the contract date. In the traditional, paper-oriented world, this mismatch between legal obligation and capacity might be difficult to detect across a multiplicity of complex contractual arrangements with many such customers. However, when such contractual terms and conditions are represented in terms of computer-processable data, it is easy for a computer system to detect such an over-commitment. Because the legal obligations are data objects, they are capable of being compared and computationally analyzed in a way not realistically possible when these promises are buried among provisions of a large, written-language contractual document. The ability to automatically detect or prevent such difficult-to-fulfill contractual commitments can reduce legal risk.

Another novel property is that computable contracts are capable of being analyzed across multiple chains of analysis that would be hard for a person to follow. A simple example of a chain was illustrated by the earlier example of a licensing condition allowing access to materials only to “students who have majored in engineering.” This requires a short chain of analysis: the computer system must first identify a student’s major, and on the basis of this major, check data to

²¹¹ See *id.* at 79 (“A business policy is a high level directive that exists to control, guide, and shape how an enterprise realizes its courses of action. Business policies define what is allowed or not allowed, and direct or specify constraints on how it should be done.”).

²¹² See *id.*

see whether this major has been demarcated using data as an “engineering major.” Such is the example of a deductive chain — using multiple sources of data to engage in logical deduction to produce a result. For example, Student A is listed as a “physics” major, “physics” is listed as an “engineering major,” therefore the computer system can, through deductive logic applied to data spread across multiple sources, quickly arrive at conclusions that would be somewhat cumbersome for a person to determine.

In principle, such deductive chains are capable of more sophistication. Thus for example, a commercial contract might consider a U.S. customer in breach of a contractual arrangement if payment had not been made within thirty days, whereas an international customer might have ninety days to make payment. A computer can perform this analysis, applying information from customer data that indicates whether customers are foreign or domestic, and cross-referencing their payment information and contract status date. Such deductive chains are automatable when contractual obligations are made computable.

Another example of a computable contract serving as “input” comes from the financial domain. There, financial contracts — represented as data — serve as “inputs” to the computer models that financial firms used to conduct their trading. Firms use such computer models to automate their trading and to model the state of their financial positions and risks. When their financial contracts are in computable form, the computer systems can read the data and automatically assess what equities the firms have, for example, the legal obligation to buy or sell. Such systems can automatically determine whether such contracts are worth exercising or not based upon market-value data. The important point is that because the contractual terms and obligations have been represented in data-oriented, semantically significant form, they can interface and be used as data for other, unconnected systems in a way that obligations written as descriptive language cannot. This ability to represent legal contractual obligations as data-objects that can be compared, processed, and inputted into other systems, is perceived as another significant benefit of computable contracting scenarios.

3. Permits Autonomous “Computer to Computer” Contracting

Finally, the computability of contracting is useful because it allows for computers to engage in *autonomous* computable contracting. Autonomous contracting essentially means that computer systems — acting as agents of human parties and subject to predefined rules and

constrictions — enter into contracts with other computers.²¹³ Such autonomous contracting, while currently fairly basic, is increasingly becoming an important part of electronic commercial dealings.²¹⁴ For example, financial firms program computer models to engage in the automated trading of securities. These contractual arrangements are entered into automatically from one computer system to another, without human intervention.²¹⁵ A financial firm might program a trading algorithm with a strategy to purchase certain securities on the basis of data, and then autonomously enter into those contractual arrangements with the computer of another firm. Similarly, the purchase and pricing of certain advertisements on the search site Google is negotiated autonomously, between computers.²¹⁶

To avoid confusion, let me emphasize the distinction between *autonomous* computable contracting and computable contracting in general. In the autonomous context, not only is the contract computable (expressed in terms of data and rules), but also, the computer systems *themselves* are engaging in contracting automatically, without human intervention. By contrast, computable contracting covers any sort of contractual arrangement in which the terms of the contract have been represented in terms of data and rules, regardless of whether it was a person who entered into the arrangement and then chose to represent it contractually, or if it was an autonomous computer system that entered into the arrangement. This Article is primarily concerned with the principles of computable contracting, however the contract came to be entered into. However, the two concepts are related because before a computer system can autonomously enter into a contractual arrangement, that arrangement must *first* be capable of being represented in a computer-processable and interpretable form.

²¹³ Hal Varian, *Computer-Mediated Transactions*, in *THE NEXT DIGITAL DECADE: ESSAYS ON THE FUTURE OF THE INTERNET* 240-43 (Berin Szoka & Adam Marcus eds., 2010).

²¹⁴ Kevin J. O'Brien, *Talk to Me, One Machine Said to the Other*, N.Y. TIMES (July 29, 2012), <http://www.nytimes.com/2012/07/30/technology/talk-to-me-one-machine-said-to-the-other.html> (estimating that the amount of machine-to-machine communication will eventually outnumber human-to-human communication over the world's wireless networks).

²¹⁵ Varian, *supra* note 213, at 240.

²¹⁶ See Google Adwords, Setting up automated rules, available at <http://support.google.com/adwords/bin/answer.py?hl=en&answer=2472779> (last visited Oct. 19, 2012).

B. *Theory: Transaction Costs, Contracts, and Property*

There are theoretical implications should data-oriented and computable contracting use expand. The scope of certain laws can depend heavily upon prevailing transaction cost levels. When transaction costs broadly decrease, the *effective* scope of certain laws can change as well. This scope change occurs even though the legal text and doctrine appear to have not changed at all. Data-oriented and computable contracting approaches have the potential to alter the substantive scope of the laws in which they are employed. This is because they are transaction-cost-reducing technologies capable of potentially broadly reducing transaction cost levels. To the extent that the scope of a given law is implicitly linked to assumptions about transaction cost levels, changes in those prevailing levels can result in *substantive* shifts in apparently unrelated laws.

To illustrate this argument, I will use copyright law's fair-use limitation as an example of a doctrine whose scope is partially linked to prevailing transaction cost levels. Should computable or other technological contracting become more prevalent, substantive shifts in relative scope are possible in this, and other similar, areas of law whose scope is linked to contracting transaction costs.

To understand how the scope of a law can depend upon transaction cost levels, consider the general contours of copyright's fair use doctrine.²¹⁷ Copyright holders of creative works (such as movies or books) can normally forbid others from making copies (or engaging in other uses of these creative works) without authorization.²¹⁸ To reproduce a copyrighted work, a third party must normally seek authorization from the copyright holder or risk copyright infringement.²¹⁹ An unauthorized use of a copyrighted work would normally constitute copyright infringement and subject the user to copyright's various remedies.²²⁰ However, if the reproduction of a work qualifies as a "fair use," it does not constitute copyright infringement, even if the user does not obtain authorization.²²¹ Thus, under fair use, a literary critic could reproduce part of a novel's text in a critical review without permission, and such an unauthorized duplication of the text would not constitute infringement.²²² There are

²¹⁷ See 17 U.S.C. § 107 (2012).

²¹⁸ *Id.* § 106.

²¹⁹ *Id.*

²²⁰ *Id.* §§ 501-506.

²²¹ *Id.* § 107.

²²² *Id.*

multiple policies animating fair use, and the copyright statute enumerates several factors to determine whether an unauthorized use constitutes a non-infringing “fair use.” The justification for fair use has also been linked to transaction costs.

In a well-known paper, Wendy Gordon linked part of the scope of fair use to transaction costs and market failures.²²³ When transaction costs inhibit authorizations through the market, Gordon argues that certain unauthorized uses should be excused under fair use.²²⁴ In such a scenario, Licensee A might be willing to pay for a low-valued use, but the transaction costs of bargaining and agreeing to authorization are high relative to the value of the use. Because parties generally won’t spend significant resources contracting over things that are not worth that much, the feasibility of agreements arising in the market are low. In such contexts, Gordon argues that courts should excuse these unauthorized uses.²²⁵ There are positive social benefits to such uses, and we should not let society be worse off simply because there are transaction costs inhibiting explicit authorization. Rather, courts should allow such uses to occur absent permission by deeming them fair uses.²²⁶

The scope of fair use is thus partially dependent upon prevailing transaction cost levels. If the doctrine excuses infringements based upon transaction cost levels, then the effective scope of the doctrine will alter as transaction costs change. If transaction costs are high, the domain of otherwise infringing uses that courts will excuse under this “market failure” justification will be larger. This fair use approach excuses socially beneficial uses whose market authorizations are being inhibited due to transaction costs. Thus, if there are greater transaction cost levels in society, there will be more of these inhibited authorizations. Conversely, if transaction cost levels decrease, the class of uses that can be excused for inability to efficiently contract for permission will shrink.²²⁷ Thus, even though the doctrine may appear superficially to be constant, the *effective substantive scope* — the

²²³ See generally Gordon, *supra* note 29, 1618 (analyzing fair use in relation to transaction costs and market failures).

²²⁴ See *id.* at 1616.

²²⁵ *Id.*

²²⁶ *Id.* at 1635-37.

²²⁶ *Id.*

²²⁷ Of course, the domain of uses that are justified or excused on other bases may be unaffected. But if we consider the scope of fair-use to be the set of uses that are excused or justified by any fair use policy, the net scope may alter as transaction costs alter.

domain of uses actually excused under the doctrine — can alter due to exogenous sources as transaction cost levels change.

It is worth noting that this transaction cost level/substantive scope dynamic is generalizable along several dimensions. First, fair use was meant to be an example of just one legal doctrine whose scope is partially linked to transaction cost levels. There are other laws whose scope is explicitly or implicitly linked to assumptions about these levels. For example, Richard Posner and William Landes (following Coase) argue that many property laws can be partially justified by assumptions of prohibitive transaction costs involved in mass contracting.²²⁸ Although a detailed exposition is beyond the scope of this Article, it is worth noting the relationship between computable and data-oriented contracting and substantive scope more generally. It is helpful thus to conceive of transaction costs as more than simply manifestations of inefficiency.²²⁹ We should also understand them to have a functional, regulatory role because assumptions about what is possible given prevailing transaction cost levels often shapes explicit legal scope.

To the extent that laws or justifications rest upon assumptions of transaction costs associated with mass-contracting based upon prevailing levels, and to the extent that technological advances allow for computing technology to reduce certain transaction costs broadly, the relative substantive scope of legal doctrines may alter as transaction cost levels change. Thus, to the extent that lawmakers wish to preserve substantive balances of rights in a context of changing transaction costs, the doctrine or statutory law cannot remain constant. To remain unchanged while the contextual framework of transaction costs upon whose scope is delineated changes is to effectively permit an alteration in substantive rights.

There are a few caveats to note. First, it is hard to predict to the extent to which these contracting technologies will gain adoption, and in which areas. As Part III noted, there are significant limitations in the extent to which the objects of contracts can be represented and automated. Moreover, there generally has to be an economic business case for these contracting technologies to gain widespread adoption. In the case of the financial domain, it was the extreme efficiencies

²²⁸ WILLIAM LANDES & RICHARD POSNER, *THE ECONOMIC STRUCTURE OF INTELLECTUAL PROPERTY LAW* 12-13 (2003) (“Reducing transaction costs is the very *raison d’être* of property rights.”).

²²⁹ For an argument that the scope of legal privacy protections is more dependent upon changing transaction cost levels than explicit changes in law, see Surden, *Structural Rights*, *supra* note 28, at 1605-09.

brought about by electronic trading that caused the impetus for data-oriented and, increasingly, computable contracting. However, as described above, due to changes in technology, these approaches are making their way into domains such as intellectual property.

In sum, the substantive scope of laws is often subtly dependent upon transaction cost levels. Embedded in laws are assumptions about prevailing levels of transaction costs and what activities are presumed to be possible or costly, given the understandings of lawmakers at the time the laws were crafted. When transaction costs levels change, the scope of a law can change even if the doctrine or text remains constant. Thus, any technology which broadly changes prevailing transaction cost levels — such as computable and data oriented contracting — can potentially change the substantive scope of even seemingly unrelated laws.

CONCLUSION

This Article introduced the concepts of data-oriented and computable contracts. Parties create “data-oriented” contracts when they express core parts of their contract in the form of highly-structured data. This data-oriented form of expressing contract information permits computers to reliably extract and identify core terms. Parties create data-oriented contracts to facilitate the use of computers as applied to their contractual obligations. This is mainly driven by the fact that contemporary computer technology is unable to reliably process written (or spoken) language — the form of expression in which commercial contracts have historically been expressed. Thus, parties have begun to reorient the form in which they express their contractual terms to make them more amenable to computer processing in domains — such as finance and e-commerce — where the efficiency benefits of computer processability are desirable.

Representing contractual information in computer-processable data allows for the application of computer abilities to contractual substance. In some instances, parties can design contractual terms or conditions to be computable. To make a contractual term computable, the parties have to design a computer-based system upon which a computer can make automated, *prima-facie* assessments as to conformance or non-conformance with certain contract terms. This process essentially involves the parties providing a translation of a particular contractual term or criteria into a comparable set of computer rules that effectuate their intended meanings. Similarly, to make the assessment partially or fully automated, the parties must also

provide the computer system with access to relevant information against which performance can be assessed. In this way, relatively straightforward comparisons between contract terms and party activities can be automated. In essence, by automating comparisons that may have been previously done manually, transaction costs related to monitoring or assessing compliance are reduced.

Not all, or even most, contractual arrangements or aspects of contracting are amenable to the data-oriented and computable paradigm. This Article further explored the limitations of this data-oriented and computable approach by linking to technology and legal theory. The computable contracting paradigm is consonant with legal assessments that look more like the determination of a “correct” *prima-facie* legal result. This paradigm, however, is poorly suited for legal contexts that require *ex-post* balancing of reasonable, but conflicting, rights that resemble policy-making or where flexibility to accommodate meritorious exceptions to general rules is desirable. To reflect the limitations of such automated assessments, this Article qualifies them as “prima-facie,” as they are automated, rules-based assessments based upon the information provided to the computer system. Such an automated result may differ, or prove inconsistent, with an ultimate determination by an authoritative legal decision-maker, such as a judge. Thus, the decision to create a computable contracting arrangement reflects a contractual judgment to deliberately forgo *ex-post* flexibility in favor of efficiency, or an election of contracting arrangements in which *prima-facie* judgments are relatively accurate proxies for ultimate judicial or authoritative legal determinations.

Finally, this Article noted that the data-oriented and computable contracting approaches have the effect of reducing particular transaction costs associated with contracting. In contexts where computable contracting-like approaches become common, this may affect the substance of existing doctrines which are justified upon assumptions of significant transaction costs associated with particular types of contracting.