Computer-executable Legal Specifications

Computer readable, understandable, verifiable, and executable legal contracts, regulations, laws & statutes

Philip Treleaven, Danielle Denny, Rares Dolga, Jacob Moss, Martin Schoernig
University College London

ABSTRACT
This paper is a review of the major international activities to create computer-understandable and executable legal specifications. They cover rules for legal contracts, regulations and laws/statutes. This includes legal markup languages (e.g. legal XML); contract management software; computable (i.e. executable) contracts; blockchain Smart contracts; and what we refer to as 'digital twin'[1] legal (i.e. computer-executable plus human-readable) specifications.

Computer-executable legal specifications are fundamental for the digital automation of legal contracts and regulations, digital commerce and integrated infrastructures, such as Internet of Things interconnection of devices and Industry 4.0 for manufacturing automation (Alcácer and Cruz-Machado 2019) .

The paper reviews the different approaches to computer-understandable legal specifications, their potential impact on digital commerce (e.g. reducing errors, supporting automation), and important underlying AI technologies: machine learning, natural language processing (NLP) and sentiment analysis etc. We also present a 'digital twin' legal specifications system where each set of rules comprises a human readable specification (i.e. natural language) and a corresponding computer-executable specification (i.e. code).

1. Introduction

As an illustration of the potential impact of data science, in a well-publicized study, 20 US lawyers with extensive experience of corporate law competed with AI software to review five non-disclosure agreements. Interestingly, while the lawyers were able to achieve an accuracy rate of 85%, each taking around 1½ hour to complete a review; the legal AI software achieved an average rate of 95% and finished in 26 seconds (LawGeex 2018).

Central to automation are computer-understandable legal specifications that encompass legal rules a computer can read, understand, verify, and possibly execute. Three categories are a) specifications that are machine-*understandable* (e.g. markup languages); b) specifications that also are machine-*executable* (e.g. computable contracts. Smart contracts); c) 'digital twin' specifications that are human-readable and machine-executable (Digital Twin 2020).

We define *digital twin* legal specifications as legal rules a computer can read, understand, verify, and execute; with a corresponding human-readable specification. The term digital twin, in general, describes a computerized or digital version of an associated physical object. Digital twin legal specifications are pivotal in the 'digital revolution' allowing legal professionals to write natural language rules (e.g. contracts) that facilitate AI-based automation and digital marketplaces.

Numerous international consortia and research groups are investigating computer-understandable legal specifications, as we will detail later in the paper. Ideally, computer-executable legal specifications should embody the following attributes: human-readable, machine-readable, machine-translatable, formal verification and computer simulation (of the potential impact).

[1] Digital Twin – in general, a digital replica of physical assets; processes, people, systems and devices.

The major activities to develop computer-understandable legal specifications include: legal standards and markup languages, contract management software, computable contracts, smart contracts and digital twin legal specification, all of them will be explained on the following topics.

The driving force for these computer-understandable specifications are the underlying data science technologies. Broadly, the data science technologies divide into a) *algorithm technologies* – new forms of 'statistics', such as machine learning, computational statistics, and complex systems (e.g. deep neural networks, Monte Carlo simulation); b) *AI analytics technologies* – covering the application of the data technologies (e.g. natural language processing, sentiment analysis, Chatbots); and c) *infrastructure technologies* – providing the infrastructure for information management and automation (e.g. digital object identifiers, Internet of Things, blockchain).

Computer-executable legal specifications have a profound role in the automation of digital commerce. Figure 1 shows contributing technologies, such as Blockchain Distributed Ledger Technologies (UK Government Chief Scientific Adviser 2016), and Digital Object Identifiers (DOI 2020).

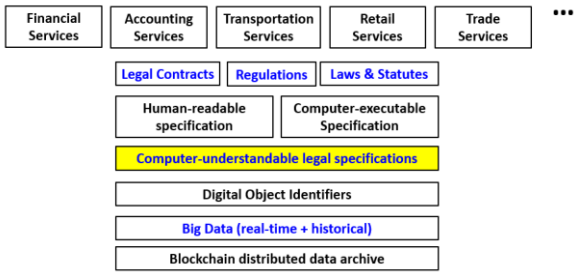


Figure 1: Digital automation, marketplaces & commerce

As an example, consider the automation of a Health Insurance contract. Each patient will have a blockchain record and a unique DOI; likewise each consultant recognised by the insurance company. When the patient needs to see the consultant, they apply online with their and the consultant's DOIs. The computer-executable contract verifies the consultation, the patient's medical condition, entitlements, consultant's expertise, and authorise the consultation. Once the contract verifies the consultation has occurred, it will initiate the payment and update the associated patient and consultant blockchain records.

2. International Activities and Research

Numerous international consortia and research groups are investigating computer-understandable legal specifications. These include international legal standards/markup consortia, such as the Oasis LegalXML group (Oasis LegalXML 2019), LawTech companies offering contract management software, and blockchain Smart contracts technology (Treleaven 2017).

Ideally, computer-understandable legal specifications should embody the following attributes:

- **Human-readable** – a legal specification drafted and readable by humans, especially legal professionals; from which the computer-understandable version is derived.
- **Machine-readable** – specifications in a data format understandable by a computer, such as XML[2], JSON[3] etc.
- **Machine-translatable** – supporting a 'data exchange' standard format; taking a legal specification under a source schema and transforming it into an equivalent specification under a target schema (cf. Fast Healthcare Interoperability Resources (FHIR, 2019)).
- **Generativity** – encompassing contracts, regulations and laws/statutes with the ability to create legal specifications and contractual formulations from existing templates.
- **Formal verification** - specifications where the correctness of the legal rules underlying a specification are verifiable with respect to formal methods of mathematics.

[2] eXtensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.
[3] JavaScript Object Notation (JSON) is a lightweight data-interchange format.

- **Computer simulation** – allowing the impact of legal specifications (cf. contracts, regulations, laws/statues) to be assessed prior to deployment.

A wealth of contributory but largely siloed legal specification research is underway:

- **Computational law** - is a branch of legal informatics concerned with the mechanization of rigorous legal reasoning (whether done by humans or by computers); it emphasizes explicit behavioural constraints and eschews implicit rules of conduct. Computational law is an approach to automated legal reasoning focusing on semantically rich laws, regulations, contract terms, and business rules in the context of electronically mediated actions" (Love and Genesereth 2005).
- **Legal informatics** - is the study of the structure and properties of information, as well as the application of technology for the organization, storage, retrieval, and dissemination of legal information (Salami 2017).
- **Ricardian contracts** - are digital documents that define the terms and conditions of an interaction between two or more parties, written in human readable text, which is then cryptographically signed and verified. The goal is easily readable contracts for both humans and programs, providing clarity to any situations that may arise.
- **Code-is-law** - "code is law" is a form of regulation where the developers who are private actors create rules and may as well embed their values into technology effectively constraining the possibilities of the users (Lessig 1999). This can bring benefits related to the ability to automate legal norms and to enforce rules and regulations (Hassan and De Filippi 2017).
- **Jurimetrics** - is the application of quantitative methods, and often especially statistics, to law.
- **Empirical legal analysis** - is the empirical analysis of legal decisions, and their relation to legislation. These efforts usually make use of citation analysis, which examines patterns in citations between works using, for example, graphs of legal precedent, called citation networks.

Next, we review the major international consortia developing computer-understandable legal specifications, especially legal XML markup languages, and Smart contracts. Two leading consortia are the Accord project (Accord project 2017) developing legal XML and Stanford's CodeX project developing computable contracts (Stanford University 2015).

3. Legal Specifications Approaches

As discussed, the major activities to develop computer-understandable specifications include:

- **Legal Standards & Markup Languages** – typically an eXtensible Markup Language (XML) markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable (e.g. <tag> <string> </tag>). An XML is typically domain specific (e.g. legal XML) with the set of <tag>s defining the domain (e.g. <contract> … </contract>).
- **Contract Management Software** – software manages the creation of a contract, and possibly negotiations, amendments, signatures and data analysis.
- **Computable Contracts** – a legal specification that a computer can read and importantly execute. Not to be confused with an "executory contract"; in law, one that has not yet been fully performed or fully executed.
- **Blockchain Smart Contracts** - are simply rules, possibly computer programs that attempt to codify transactions and contracts associated with a Blockchain or other distributed ledger technology.
- **Digital Twin Legal Specification** – a set of rules comprising a human readable specification (i.e. natural language) and a corresponding computer-executable specification (i.e. code). A

specification drafted and readable by humans from which the computer-understandable version is derived; or a human-readable version derived by AI from a computer-executable specification.

3.1 Legal Standards and Markup Languages

Two prominent markup languages projects are Oasis LegalXML and CEN MetaLex.

Oasis LegalXML

LegalXML (Oasis LegalXML 2019) is a member section within OASIS the not-for-profit, global consortium that brings legal and technical experts together to create standards for the electronic exchange of legal data. LegalXML produces standards for electronic court filing, court documents, legal citations, transcripts, criminal justice intelligence systems, and others.

OASIS members participating in LegalXML include lawyers, developers, application vendors, government agencies and members of academia.

CEN MetaLex

CEN MetaLex (Metalex 2006) standardizes the way in which sources of law and references to sources of law are specified in XML. CEN is the European Committee for Standardisation (CEN 2019). MetaLex was developed by the CEN workshops on an Open XML Interchange Format for Legal and Legislative Resources.

CEN MetaLex is an interchange format, a lowest common denominator for other standards, intended not to replace jurisdiction-specific standards and vendor-specific formats in the publications process but to impose a standardized view on legal documents for the purposes of information exchange and interoperability in the context of software development. To meet these requirements, MetaLex defines a mechanism for schema extension, adding metadata, cross-referencing, constructing compound documents and a basic naming convention.

3.2 Contract Management Software

This is drafting software supporting contract creation and includes libraries of contract templates, drafting tools frequently interfaced with Microsoft WORD, and verification tools for contracts. More generally, contract management software is the range of computer programmes, libraries and data used to support contract management, contract lifecycle management, and contractor management on projects (GetApp 2020). There is a vast list of companies in the marketplace; examples are LexisNexis Lexus-Draft, ClauseBase, Contract Express, Contract Advisor etc.

Lexus-Draft

Lexis Draft, as an illustration, is a Microsoft WORD toolbar that helps professionals draft contracts faster and efficiently (Lexis®Draft 2019). It checks for inconsistencies in documents, flags areas that need attention and links the contract directly to other LexisNexis services within MS WORD.

3.3 Computable Contracts

Prominent examples of *computable* (computer-executable) contracts projects are the Accord project and Stanford University's CodeX.

Accord Project

The Accord Project (Accord project 2017), comprising leading law firms, is developing technical specifications such as data schemas, models, templates, a Smart legal contract programming language and a contract execution engine that embody legal best practices and work towards a common, universal, implementation for Smart legal contracts. The Project has created the Open Source software packages described later.

Stanford CodeX Project
CodeX computable contracts (Stanford University 2015; LSP 2018) project works on developing a universal Contract Definition Language designed for expressing contracts, terms and conditions, and even laws in machine-understandable way so that automated tools can be used to work with them more efficiently. Its roots lie in declarative rules based logic programming. Automation includes: a) *Contract Formation* - checking for conflict between laws and/or contract terms; b) *Contract Analysis* - analysing the implications of laws and/or contract terms in a particular situation; c) *Contracts Implementation* - deciding what actions can/should be done in a particular situation for achieving a particular goal.

3.4 Blockchain Smart Contracts
This topic will be further explained in item 7, but summing up, Smart contracts is a term used to describe computer code that automatically executes all or parts of an agreement stored on a blockchain-based platform. The Smart contract code is typically the sole manifestation of the agreement between the parties or might also have an associated traditional text-based contract The code executes certain provisions, such as transferring funds from Party A to Party B. Prominent Blockchain Smart contract projects include Ethereum Solidity (Ethereum 2016), Hyperledger Fabric (Hyperledger 2020) and JP Morgan's Quorium (JP Morgan 2017). These are discussed in section 7.

3.5 Digital Twin Legal Specification
We use the term 'digital twin' to emphasise a set of legal rules comprising a human readable specification (i.e. natural language) and a corresponding computer-executable specification (i.e. code).

The two starting points are:

- **Human-readable specification** – the legal contract, regulation or law is draft in natural language and the corresponding computer-executable 'code' is derived either 'manually' using templates associated with each rule, or 'automatically' by an integral AI-based NLP tool.
- **Computer-executable specification** – an existing legal specification, such as a contract or regulatory handbook, is analysed by an AI-based NLP tool, and a computer-executable specification derived. This is more challenging technically due to the syntactic and semantic ambiguity found in existing legal specifications.

Example 'digital twin' systems include Clause.io (Clause 2020), OpenLaw (OpenLaw 2019) and the UCL computer-executable legal specification system described in this paper.

Clause.io
Clause.io's uses smart clause templates to convert natural language terms into programmable clauses to automate business processes. It uses pre-built smart clause templates for common operations or build contracts. Inserts smart clause templates across different contract instances and contract types manually or programmatically via the Clause API.

OpenLaw
Is a solution to pull data from external sources into smart contracts, it can use pre-built clauses from its templates library and also enable the conversion of agreements using mark-up language. It helps the creation of decentralize applications that integrate legal frameworks and enables the transactions to be validated using Elliptic Curve Digital Signature Algorithm (OpenLaw 2019).

Next, as background and for completeness we describe the data science technologies associated with computer-understandable legal rules.

4. Computer Science Technologies
The key data science technologies underpinning computer-understandable legal specifications are:

- **Algorithms and Artificial Intelligence (AI)** – algorithms cover three broad domains: Computational Statistics (e.g. Monte Carlo methods), AI (e.g. Artificial Neural Networks), and Complex Systems (e.g. Agent-Based systems).
- **AI Analytics** – applications of AI, specifically machine learning. Examples include *Natural Language Processing (NLP)* – the analysis and synthesis of natural language and speech (Hovy 2005; NLP 2020); and *Sentiment Analysis* – using NLP, statistics, or machine learning methods to extract, identify, or characterize the sentiment content of text or speech (MonkeyLearn 2018).
- **Digital Object Identifiers (DOI)** – a DOI is an identifier or handle, potentially persistent, used to identify objects uniquely, standardized by an international body (DOI 2015; DOI 2020).
- **Data Standards** – this includes common data models for a sector, such as legal services, markup specifications for representing data in a machine-readable form, and data exchange formats for converting between a target format by another program (Data Exchange 2020).
- **Blockchain Technologies** – and other types of distributed ledger technology (DLT), distributed databases that secures, validates and processes transactional data; and Smart contracts, a self-executing contract with the terms of the agreement between buyer and seller directly written into lines of code (Treleaven 2017)

4.1 Algorithms and Artificial Intelligence

The terms algorithm, artificial intelligence (AI) and machine learning are used interchangeably. However, data science algorithms cover three broad domains: Computational Statistics (e.g. Monte Carlo methods), AI and ML (e.g. Artificial Neural Networks), and Complex Systems (e.g. Agent-Based systems) (Koshiyama 2020). See Figure 2.

- **Computational Statistics** - computationally intensive statistical methods.
- **AI Algorithms** - mimicking a new form of human learning, reasoning, knowledge, and decision-making
  - Knowledge or rule-based systems
  - Evolutionary algorithms
  - Machine learning
- **Complex Systems** - system featuring a large number of interacting components whose aggregate activity is nonlinear.

Figure 2: Algorithm taxonomy (Koshiyama 2020)

AI algorithms are a continuum of epistemological models spans three main communities:

- **Knowledge-based** or heuristic algorithms (e.g. rule-based) - where knowledge is explicitly represented as ontologies or IF-THEN rules rather than implicitly via code;
- **Evolutionary** or metaheuristics algorithms - a family of algorithms for global optimization inspired by biological evolution (e.g. Genetic Algorithms, Genetic Programming, etc.); and
- **Machine Learning** algorithms - a type of AI program with the ability to learn without explicit programming, and can change when exposed to new data; mainly comprising *Supervised*, *Unsupervised*, and *Reinforcement Learning*.

ML algorithms are broadly a combination of the *classic trio* of Supervised, Unsupervised and Reinforcement Learning, with the *disruptors*: Deep Learning, Adversarial Learning, Transfer and Meta Learning. This interaction constantly yields some models (e.g. Generative Adversarial Networks) and applications (e.g., Natural Language Processing, Object Recognition, Forecasting etc.).

Classic ML subdivides into:

- **Supervised learning** - learns or infers a pattern (function) from labelled training data consisting of a set of training examples.
- **Unsupervised learning** - learns or infers a pattern in a data set with no pre-existing labelled training data and with a minimum of human supervision.
- **Reinforcement learning** - enables an algorithm using a system of reward and punishment to learn by trial and error using feedback from its own actions and experiences.

4.2 AI Analytics

AI analytics technologies, especially natural language processing (NLP) and sentiment analysis, play important roles in analysing legal specifications. Analytics is the application of computational statistics, AI and complex systems to analysis large and varied data sets to uncover hidden patterns to help make informed decisions. This includes natural language programming (NLP), sentiment analyse, plus behavioural and predictive analytics (Predictive Analytics 2020).

Natural Language Processing

NLP is the understanding of humans' natural language through text or speech. The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages. Natural Language processing is a difficult problem in computer science, due to the ambiguous nature of the human language. It requires understanding both the syntax and semantics; the words and how the concepts are connected to deliver the intended message. NLP can automate a Regulator's call-centre function. Providing a user interface to a regulatory rulebook, answering questions interacting in natural language text and speech (Hovy 2005; NLP 2020).

Sentiment Analysis

Sentiment analysis is the computational process of understanding the meaning and interpretation of words and how sentences are structured in a piece of text or speech. For example, identifying and categorizing opinions expressed to determine whether the meaning conveyed is positive, negative, or neutral (MonkeyLearn 2018).

4.3 Infrastructure Technologies

The technologies we highlight are digital object identifiers, data standards and DLT-Blockchain. We might have also included Chatbots and Internet of Things (IoT).

Digital Object Identifiers

DOIs are set to have a major impact on business and 'legal matters' for information management. As context, unique identifiers are central to the management of digital (e.g. ISBN, URLs) and physical objects (e.g. telephone numbers, barcodes, US SSN, IP addresses). Digital Object Identifiers (DOI) (DOI, 2020), pioneered by Bob Khan co-inventor of the Internet, unifies management of information and digital objects across the Internet, with the Handle system (Handle, 2019) resolving the physical location of an object. DOIs originated in the requirement to define uniquely publications, make them persistent, and accessible via the Internet. DOIs are unique identifiers used for a class of objects; persistent with a long-lasting reference to a digital object; and resolvable by identifying the location of a digital object. The DOI Handle System (CNRI 2019) provides secure name resolution over the Internet, designed to enable a broad set of communities to use the technology to identify digital content independent of location.

For legal specifications, DOIs offer a uniform schema for uniquely identifying documents, contracts, regulations and laws/statutes, together with associated code, individuals, firms and IoT devices.

Data Standards

Data standards support interoperability and collaboration, and typically are specialised to an industry sector (e.g. legal services). LegalXML is an example. Data standards cover three areas:

- **Data model** – organizes elements of data and standardizes how they relate to one another and to their properties; including ontologies and data tagging. Open source standard data models are emerging for major Services sectors (e.g. Open Banking).
- **Markup languages** – in general, a system for annotating a document in a way that is syntactically distinguishable from the text, meaning when the document is processed for display, and is only used to format the text (e.g. Web page HTML, XML).
- **Data Exchange formats** - data exchange or interchange formats are the process of taking data structured under a source schema and transforming it into data structured under a target schema, so that the target data is an accurate representation of the source data (e.g. FHIR).

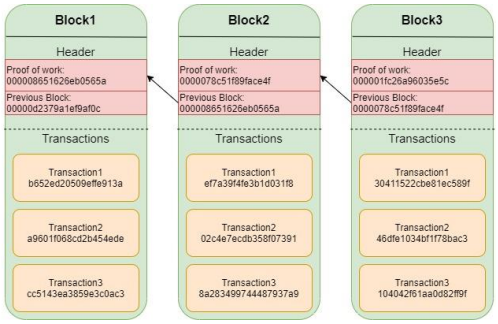


Figure 3: Blockchain Distributed Ledge

Blockchain technologies

Elements of blockchain technology, originally conceived for Bitcoin, have far-reaching potential in other areas (UK Government Chief Scientific Adviser 2016), especially for managing legal document and regulation. The core technologies are:

- **Distributed Ledger Technology (DLT)** – a decentralized database where transactions are kept in a shared, replicated, synchronized, distributed bookkeeping record, which is secured by cryptographic sealing. The key distinction between 'distributed ledgers' and 'distributed databases' is that nodes of the distributed ledger cannot/do not trust other nodes – and so must independently verify transactions before applying them. Blockchain is a type of DLT.
- **Smart Contracts** - are simply the rules, possibly computer programs that attempt to codify transactions and contracts with the intent that the records managed by the distributed ledger are authoritative with respect to the existence, status and evolution of the underlying legal agreements they represent. Smart contract technology has the potential to automate laws and statutes. (Provable 2015).

Next, we review in more detail the different approaches to formalising legal specifications, starting with markup languages.

5. Legal Standards & Markup Languages

Data standards for legal specification include markup languages and data exchange formats.

5.1 Markup Languages

Common machine-readable notations used for markup, include CSV (coma separated value), JSON (JavaScript Object Notation) (JSON 1999), XML (eXtensible Markup Language), and HTML (HyperText Markup Language) the most common. XML (XML 2019) employs user-defined tags <tag> text </tag> in the text to support machine-readability. As discussed, two prominent projects to develop a standard for legal markup are LegalXML and Metalex.

Figure 4 illustrates a simple (Oasis-Open) LegalXML specification.

```
- <employees>
  - <person id="1392">
      <name>John Smith</name>
      <dob>1974-07-25</dob>
      <start-date>2004-08-01</start-date>
      <salary currency="USD">35000</salary>
    </person>
  - <person id="1395">
      <name>Clara Tennison</name>
      <dob>1968-03-15</dob>
      <start-date>2003-05-16</start-date>
      <salary currency="USD">27000</salary>
    </person>
  </employees>
```

Figure 4: simple LegalXML specification

5.2 Data Exchange Formats

A data interchange format allows marked-up data structured under a source schema to be transformed into data structured under a target schema, so that the target data is an accurate representation of the source data. A leading example is Fast Healthcare Interoperability Resources (FHIR) a data standard for exchanging electronic health records (FHIR 2019).

```
public class Example08_ValidateResource {

    public static void main(String[] args) {

        // Create an encounter with an invalid status and no class
        Encounter enc = new Encounter();
        enc.getStatus().setValueAsString("invalid_status");

        // Create a new validator
        FhirContext ctx = new FhirContext();
        FhirValidator validator = ctx.newValidator();

        // Did we succeed?
        ValidationResult result = validator.validateWithResult(enc);
        System.out.println("Success: " + result.isSuccessful());

        // What was the result
        OperationOutcome outcome = result.getOperationOutcome();
        IParser parser = ctx.newXmlParser().setPrettyPrint(true);
        System.out.println(parser.encodeResourceToString(outcome));

    }
```

Figure 5: Data Exchange Format FHIR

Another example is Industrial Foundation Classes which is an international standard managed by BuildingSMART (BuildSmart 2020). It is an open international standard for BIM data that creates exchangeable and sharable information among various software applications that can be used by various participants in the construction or facility management industry sector.

6. Contract Management Software

As an illustration of the LawTech activity, see Figure 6 created by Chin & Co (Chin 2018).

Figure 6: LawTech activity

Legal tools comprise:

- **Contract templates libraries** – templates are used by legal professionals and industry associations (e.g. finance, construction) to encourage the use of general conditions in contracts and reduce subsequent disputes. Examples include the International Federation of Consulting Engineers (FIDIC 2019).
- **Contract drafting tools** – computer software tools that support the drafting and checking of a legal contract. These tools frequently interface with Microsoft WORD. A number of drafting tools use the lightweight markup language Markdown (Markdown 2004).
- **Contract verification tools** - verification tools for checking cross-references, syntax, semantics and meaning of contracts.

For a review of tools, see the list of Legal Document Management Software (Capterra 2020).

6.1 Contract Template Libraries

As discussed, numerous law firms and industry associations, especially in construction, are developing and providing (online) contract templates with general conditions. An example construction association is the International Federation of Consulting Engineers (FIDIC 2019). Contract templates are pre-formatted documents that, typically, are used to sign a contract or agreement between two parties, and act as a legal document, recognized by a court. Since the requirement is different for individual users, various types of contract templates are available, either open source (Template.net 2020) or commercial from companies such as RocketLawyer or LawDepot.

6.2 Contract Drafting Tools

Closely related to template libraries are legal drafting tools. An example is Practical Law (Thomson Reuters 2020) that supplies a set of Web-based tools to help lawyers created practice notes, document

templates, standard clauses, deal checklists, and tools that lay out the basics of deal making for junior associates.

6.3 Contract Verification Tools

These legal proofreading, analysis and review solutions typically work within Microsoft WORD, scanning legal documents as they are being created, alerting the legal professional to errors, inconsistencies, and missing information.

7. Computable contracts

Computable contracts are legal specifications that are both machine-understandable and machine-executable. Leading consortia include the Accord project and Stanford University's CodeX project.

Accord Project

The Accord Project (Accord project 2017), comprising leading law firms, is developing technical specifications such as data schemas, models, templates, a Smart legal contract programming language and a contract execution engine that embody legal best practices and work towards a common, universal, implementation for smart legal contracts.

The Project has created the following Open Source software packages:

- **Cicero** - a specification and set of libraries for parameterized natural language, data models and computable logic. Cicero templates can be used to create reusable smart clauses, allowing legal clauses to interact with data from the outside world, and to perform automated computation.
- **Concerto** - a lightweight and web-friendly data model specification used to capture formally domain specific data models.
- **Ergo** - a domain specific language to express the executable logic of legal templates.
- **Model Repository** - a repository of open source data models for use in Cicero templates and with Ergo logic.
- **Template Library** - a library of open source Cicero templates.
- **Verbatim** - a Microsoft Word add-in to help define templates.

Stanford CodeX Project

CodeX computable contracts (Stanford University 2015) project works on developing a universal Contract Definition Language designed for expressing contracts, terms and conditions, and even laws in machine-understandable way so that automated tools can be used to work with them more efficiently. Its roots lie in declarative rules based logic programming. Automation includes:

- **Contract Formation** - checking for conflict between laws and/or contract terms.
- **Contract Analysis** - analysing the implications of laws and/or contract terms in a particular situation.
- **Contracts Implementation** - deciding what actions can/should be done in a particular situation for achieving a particular goal.

8. Blockchain Smart contracts

When smart contracts first appeared, lawyers often disparaged them as "not smart and not contracts" (Black 2019). Nick Szabo called the new automatic contracts "smart", because they can be more functional than their paper-based agreements; it is a "set of promises, specified in digital form, including protocols within which the parties perform on the other promises" (Szabo 1996). The automation has to maintain some basic characteristics of the contracts, specially: "verifiability; observability; rights and obligations, and enforceability" (Szabo 1996).

Upon execution, if the set of pre-defined rules are met, the Smart contract executes itself to produce the output. The characteristics of smart contracts are: a) they are self-verifying due to automated possibilities; b) they are self-enforcing when the rules are met at all stages; and c) they are reasonably tamper-proof, as its difficult to change what's been programmed

Companies are increasingly launching smart contracts as prototypes but they do not necessarily inherit the trustiness and informational integrity of the distributed ledger technology because they often have to rely on off-chain information, provided by specialized intermediaries (Lianos 2019).

8.1 Smart Contract Languages

A number of groups are developing proprietary smart contract languages (SCL) used either to write a (blockchain) smart contract directly or is compiled to it. Leading examples are Ethereum Solidity, Accord Project's 'Ergo'. Scilla and Vyper used with Ethereum.

Smart contract languages are typically declarative and deterministic. They describe a "problem" but they usually do not say how the problem should be solved. Traditional declarative languages are Prolog, XSLT, LISP and SQL.

```
pragma solidity >=0.4.0;

contract SimpleStorage {
   uint storedData;

   function set(uint x) public {
      storedData = x;
   }
   function get() public view returns (uint) {
      return storedData;
   }
}
```

Figure 7: Smart Contract notation in Solidity

8.2 Case study: Ethereum Solidity

Solidity is known as a contract-based, high-level programming language, made to enhance the Ethereum Virtual Machine. Solidity is statically typed scripting language, which does the process of verifying and enforcing the constraints at compile-time as opposed to run-time.

Ethereum Solidity

Ethereum is an open source, public, blockchain-based distributed computing platform and operating system supporting distributed ledger technology and Smart contract scripting using the Solidity programming language. Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs, which govern the behaviour of accounts within the Ethereum state. Solidity was influenced by C++, Python and JavaScript and is designed to target the Ethereum Virtual Machine (EVM).

JP Morgan Quorum

Quorum is one of the first major steps towards common adoption of blockchain and Smart contracts among financial industries, for applications such as settlement. Quorum is an enterprise-focused, permissioned blockchain infrastructure specifically designed for financial use cases built on Go Ethereum. Quorum targets applications requiring high speed and high throughput processing of private transactions within a permissioned group of known participants. Quorum addresses specific challenges to blockchain technology adoption within the financial industry, and beyond.

9. Digital Twin Legal Specifications

As discussed, the notable 'digital twin' feature is that a set of legal rules comprises a human readable specification (i.e. natural language) and a corresponding computer-executable specification (i.e. code).

The coded version of the legal specifications will have its existence alongside the natural language text as a digital twin for computation. This enables downstream tasks such as automation for legal services, regulation, governance etc. Furthermore, such specifications could underpin commerce in future digital markets and control of IoT devices (e.g. smart contracts) (Treleaven 2020).
The two principal components of a computer-executable legal specification system are: a) Drafting tool – Microsoft WORD is clearly the industry-standard for drafting contracts, regulations and statutes;

b) Mark-up tool – for translating legal specifications into machine-readable notations, such as eXtensible Markup Language (XML) and (JSON).

9.1 Drafting Tools

A growing number of contract drafting tools (e.g. Contract Express) are available commercially. The software typically allows professionals to automate templates inside Microsoft Word by using markup; with the contracts generated by filling out web-based forms or questionnaires. Figure 8a illustrates the style. The contract drafting tools automate the work of a lawyer and paralegal, without addressing the need for ELRs.

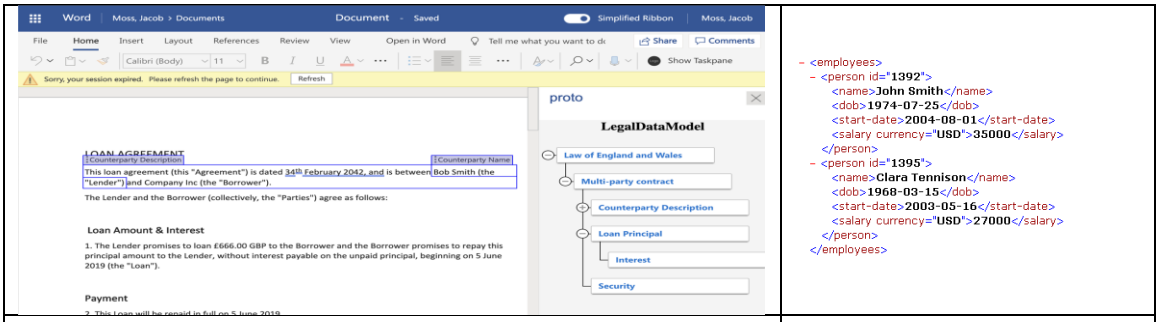
9.2 Markup Tools

There are essentially three approaches to creating the computer-executable specification:

- **Markup templates** – in the drafting tool, associate explicit templates in the natural language specification for creating the computer-executable specifications.
- **Automated markup** – in the drafting tool, using an AI-based system to generate automatically generate the computer-executable specification from the natural language specification.
- **Natural language translation** – using an AI-based system to translate a previously created natural language specification (e.g. regulatory handbook) into a corresponding computer-executable specification.

9.3 Case study: LegalExec

This shows the UCL 'Digital Twin' computable contract tool.



a) *Drafting tool* – MS WORD | b) *Markup* - XML document

Figure 8: Digital Twin Contract drafting and markup tool

Because digital twin refers to the pairing of natural language contracts and computer-executable counterpart and the majority of the existing contracts are specified in English (or any natural language), to obtain the computer-executable counterpart, a human needs to understand the specifications and requirements of the original contract and then transform it into code. This first step can be costly, however, it is possible to create automatic tools for it, but then the barrier is to make an algorithm to understand and extract meaning from the natural language contract (Dolga 2020).

Deep learning models from the Natural Language Processing (NLP) can be used to extract the main elements of a contract, which are its terms and conditions and subjects/parties involved (Dolga 2020). The task of recognizing the entities involved can be reduced to Named Entity Recognition and the extraction of terms and condition can be reduced to a classification problem involving detection of tokens that form the structure of the contract's terms, in conditions and consequences that build up a logical rule from the natural text. "Legal services digitalisation is one aspect in the wider context of Trade, Digital Markets & IoT, Accounting, Audit & Tax, Regulation & Compliance, Laws & Legal Services" (Treleaven 2020).

For legal specifications, the relationship between NL and code is bi-directional. On one hand contract drafting (code -> NL) saves companies money, reduces manual effort of checking over contracts, modifies previous contracts allow for mistakes to creep in. On the other hand contract markup (NL -> code) establishes a prerequisite for future digital markets; reduces inconsistent/malformed legal documents, and reduces legal exposure (Treleaven 2020).

The mark-up of NL legal contracts centres on establishing a mapping between the NL contract and a suitable data model. The latter we call the Legal Data Model (LDM). This means to decompose the NL contract into smaller building blocks which itself have a legal meaning (Treleaven et al. 2020).
To obtain robust results, powerful models such as Bidirectional Transformers can be employed, but they require a lot of training data. Since the task of translating from natural language contracts to computer-executable rules is a newly explored topic, there are not many available data sets. However, using pre-trained models such as BERT (Google BERT 2020) and transfer learning techniques, this problem can be solved (Dolga 2020).

10. Digital Automation, Marketplaces and Commerce

In Figure 1 we illustrated the technology stack for digital automation and marketplace. At the base is blockchain providing a secure distributed database; next the Big data layer; then unique digital identifiers, and above federated learning.

10.1 Digital Automation

As an illustration of computer-executable legal contracts, consider the shipment of products from a factory in China to a retailer in Germany requiring a series of contracts and payments. A transport company takes the products to the port; arrival might trigger payment to the manufacturer and transport company. Another contract covers the shipping of the product. Its arrival in Germany might trigger payment to the shipper. The products are stored temporarily in a warehouse at the docks, prior to transportation to the company warehouse, and so on. Each stage is likely to require a separate contract and staged payments.

Conceptually, a single computer-executable contract can automate the whole process paying each party as they complete their part.

10.2 Digital Marketplace Maersk

TradeLens (IBM and Maersk 2018) is an interconnected ecosystem of supply chain partners — cargo owners, ocean and inland carriers, freight forwarders and logistics providers, ports and terminals, customs authorities, and more. It uses the IBM Blockchain Platform that is based on open-source permissioned distributed ledger technology Hyperledger Fabric.

10.3 Case study: LegalExec and RegNet

In Figure 1 we presented a technology stack for computer-executable legal specifications. This combines two UCL projects: a) *LegalExec* to create a 'digital twin' legal specification platform, and b) *RegNet* a privacy-preserving data platform. We have already outlined the LegalExec platform in section 8.3.

RegNet is an infrastructural platform enabling data access through a number of privacy preservation techniques, including federated learning.

RegNet makes use of UOIs as a means to manage and locate digital objects across its network of peers and channels. What is unique about the RegNet is the way in which it implements the DOA Handle system (DOI 2015). Rather than implement the handle system as part of the blockchain's core protocol (that manages aspects such as blockchain consensus), the Handle system is a layer on top of the blockchain- on the computer-executable contract Layer. This abstraction maintains the integrity of the blockchain and the network, from the use of the network.

11. Future Legal Specification Software

An interesting role model for future 'digital twin' legal software is the Adobe web programming environment Dreamweaver (Adobe 2019). Dreamweaver is a web-authoring program that uses a WYSIWYG "what you see is what you get" interface. The program displays both the web page graphics, alongside the HTML code. (In our terms, a natural language contract together with the associated executable code.) This allows the professional (cf. lawyer or techie) to author a page either using the graphics or underlying HTML, with the corresponding notation generated automatically. The original Dreamweaver 'digital twin' system provided elementary graphics and corresponding HTML, but now provides a rich set of features.

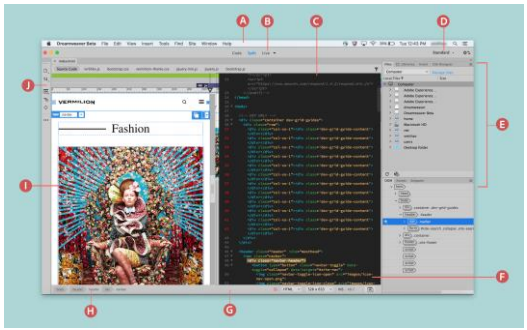


Figure 9: Dreamweaver 'digital twin' screen

Dreamweaver comprises a 'digital twin' developer workspace with real-time preview. It also includes a code editor that supports syntax highlighting, code completion, real-time syntax checking, and code introspection for generating code hints to assist the user in writing code. In addition, it provides support for Web technologies such as Cascading Style Sheets (CSS), the programming language JavaScript, and various Web scripting languages and frameworks, such as PHP. External stylesheets are stored in CSS files. Using Dreamweaver, Figure 10 shows analogous features of a legal integrated development environment. Here *style sheets* equate to the format of contracts for a specific jurisdiction (e.g. English Law); *CSS files* to a library of contract templates; *scripting language* to a programming (e.g. Solidity) for legal specifications.

	Web tool	Legal tool
Digital twin	WYSIWYG	WYSIWYG
Graphics	Web page	Natural Language doc
Markup	HTML	legal XML
Style sheet	CSS	Legal (jurisdiction) templates
Library	CSS files	Template library
Scripting Lang.	JavaScript	Ethereum Solidity

Figure 10: Dreamweaver as Role model

In conclusion, the contribution of this paper was to describe some of the technologies that may pave the way for the full automation of some legal services and regulation, such as computer readable, understandable, verifiable, and executable legal contracts, regulations, laws and statutes. Although there are many obstacles to be overcome technically, ethically, and legally an increasing integration of technology into Law is viable and necessary.

12. References

Accord project (2017) Accord Project. https://www.accordproject.org/

Adobe (2019) Adobe Dreamweaver, https://www.adobe.com/en/products/dreamweaver.html

Alcácer V, Cruz-Machado V (2019) Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems. Engineering Science and Technology, https://www.sciencedirect.com/science/article/pii/S2215098618317750

Black DB (2019) Blockchain Smart Contracts Aren't Smart And Aren't Contracts. In: Forbes. https://www.forbes.com/sites/davidblack/2019/02/04/blockchain-smart-contracts-arent-smart-and-arent-contracts/

BuildSmart (2020) BuildSmart Industry Foundation Classes, www.buildingsmart.org/standards/

Capterra (2020) Best Legal Document Management Software | 2020 Reviews of the Most Popular Tools & Systems. https://www.capterra.com/legal-document-management-software/

CEN (2019) European Committee for Standardization - Home. https://www.cen.eu/Pages/default.aspx

Chin E (2018) Exploring the global LegalTech ecosystem. In: services@disruption. https://disruptservices.io/global-legaltech-ecosystem/

Clause (2020) Clause Connect your contracts to APIs and systems. https://clause.io/

CNRI (2019) DOI Handle.Net Registry. http://www.handle.net/

Data Exchange (2020) Data exchange, Wikipedia, https://en.wikipedia.org/wiki/Data_exchange

Digital Twin (2020) Digital twin, Wikipedia, https://en.wikipedia.org/wiki/Digital_twin

DOI (2020) Digital object identifier. Wikipedia, https://en.wikipedia.org/wiki/Digital_object_identifier

DOI (2015) The DOI Handbook. https://www.doi.org/doi_handbook/1_Introduction.html

Dolga R (2020) Executable Legal Contracts with Deep Learning, Masters Thesis, University College London

Ethereum (2016) Solidity — Solidity 0.6.6 documentation. https://solidity.readthedocs.io/en/v0.6.6/

FHIR (2019) Overview - FHIR v4.0.1. https://www.hl7.org/fhir/overview.html

FIDIC (2019) International Federation of Consulting Engineers | The Global Voice of Consulting Engineers. https://fidic.org/

GetApp (2020) Contract Management Software Reviews & Ranking for 2020. In: GetApp. https://www.getapp.com/operations-management-software/contract-management/reviews/

Google BERT (2020) Bidirectional Encoder Representations from Transformers, Wikipedia, https://en.wikipedia.org/wiki/BERT_(language_model)

Hassan S, De Filippi P (2017) The Expansion of Algorithmic Governance: From Code is Law to Law is Code. Field Actions Science Reports The journal of field actions 88–90

Hovy E (2005) Natural Language Processing: A Brief Review. https://www.isi.edu/natural-language/teaching/cs544/spring12/current/notes/setup1-NLP-ov.pdf

Hyperledger (2020) Hyperledger Fabric, https://opensource.com/article/19/9/introduction-hyperledger-fabric

IBM and Maersk GI (2018) TradeLens - Digitizing global supply chains. https://www.tradelens.com/

JP Morgan (2017) Interbank Information Network. https://www.jpmorgan.com/country/US/en/detail/1320562088910

JSON JON (1999) Introducing JSON (JavaScript Object Notation). https://www.json.org/json-en.html

Koshiyama A, Firoozye N, Treleaven P (2020) Algorithms in Future Capital Markets. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3527511

LawGeex (2018) 20 top lawyers were beaten by legal AI. Here are their surprising responses. https://blog.lawgeex.com/20-top-lawyers-were-beaten-by-legal-ai-here-are-their-surprising-responses/

Lessig L (1999) Code: And Other Laws Of Cyberspace, 1st U.S. Edition, 3rd Printing edition. Basic Books, New York

Lexis®Draft (2019) Legal Drafting Software. https://www.lexisnexis.co.uk/products/lexis-draft.html

Lianos I, Dimitropoulos G, Eich S, Hacker P (2019) Regulating Blockchain: Techno-Social and Legal Challenges. Oxford University Press

Love N, Genesereth M (2005) Computational law. In: Proceedings of the 10th international conference on Artificial intelligence and law. Association for Computing Machinery, Bologna, Italy, pp 205–209

LSP WG (2018) Developing a Legal Specification Protocol: Technological Considerations and Requirements - Legal Specification Protocol project (LSP)

Markdown (2004) Markdown Guide. https://www.markdownguide.org/.

Metalex (2006) CEN MetaLex - Open XML Interchange Format for Legal and Legislative Resources. http://www.metalex.eu/

MonkeyLearn (2018) Sentiment Analysis. https://monkeylearn.com/sentiment-analysis.

NLP (2020) Natural language processing. Wikipedia, https://en.wikipedia.org/wiki/Natural_language_processing

Oasis LegalXML (2019) OASIS LegalXML Electronic Court Filing TC | OASIS. https://www.oasis-open.org/committees/tc_home.php

OpenLaw (2019) OpenLaw. https://www.openlaw.io/

Predictive Analytics (2020) Predictive analytics. Wikipedia, https://en.wikipedia.org/wiki/Predictive_analytics

Provable (2015) Provable - blockchain oracle service, enabling data-rich smart contracts. https://provable.xyz/

Salami E (2017) A Brief Overview of Legal Informatics, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2966201

Stanford University (2015) Computable Contracts. http://compk.stanford.edu/

Szabo N (1996) Smart Contracts Building Blocks for Digital Markets. https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html

Template.net (2020) Contract Templates - Word, Docs, Pages | Free & Premium Templates. https://www.template.net/business/simple-contract-templates/

Thomson Reuters (2020) Practical Law - UK Home.
https://uk.practicallaw.thomsonreuters.com/Browse/Home/PracticalLaw/

Treleaven P, Gendal Brown R, Yang D (2017) IEEE Special Issue: Blockchain Technology in Finance. Computer 50:14–17.
https://doi.org/10.1109/MC.2017.3571047

UK Government Chief Scientific Adviser (2016) Distributed Ledger Technology: beyond blockchain.
https://www.gov.uk/government/news/distributed-ledger-technology-beyond-block-chain

XML (2019) XML, Wikipedia, https://en.wikipedia.org/wiki/XML