This is an author produced pre-print version of a conference paper later published in Erich Schweighofer et al. (Eds.), *Trends and Communities of Legal Informatics. Proceedings of the 20th International Legal Informatics Symposium IRIS 2017*. Österreichische Computer Gesellschaft / books@ocg.at, Wien 2017, pp. 425–432 (ISBN 978-3-903035-15-7), and in *Jusletter IT*, 23 February 2017, http://jusletter-it.weblaw.ch/issues/2017/IRIS.html (subscription required).

WISE CONTRACTS: SMART CONTRACTS THAT WORK FOR PEOPLE AND MACHINES

James Hazard / Helena Haapio

Founder, CommonAccord
40 Rue Lauriston, 75116 Paris, FR
james.g.hazard@gmail.com; http://www.CommonAccord.org

Associate Professor of Business Law, University of Vaasa/International Contract Counsel, Lexpert Ltd
Pohjoisranta 20, 00170 Helsinki, FI
Helena.Haapio@lexpert.com; http://www.lexpert.com

Keywords: ***Automation, codification, open source, prose objects, Ricardian contracts, smart contracts, smart contract templates, visualization***

Abstract: *Modern economies are held together by innumerable contracts. However, current contracts are neither machine-readable nor easily human-readable. The Ricardian Contract paradigm of parameters, prose and code posits a hybrid model of automation and conventional legal text. This paper connects recent work on design criteria for «Smart Contract Templates» with prose objects and prototype inheritance demonstrated at CommonAccord.org. Templates authored and shared as prose objects can become the basis for automation, codification, commentary, big data analysis and graphic presentations.*

1. Introduction[1]

Modern economies are held together by innumerable contracts, as recognized by the 2016 Nobel Prize in Economics.[2] Contracts are, however, difficult to understand and manage,[3] and inherently incomplete.[4] This paper argues that the problems of understanding, management and incompleteness can be greatly reduced by *codification* using the methods of open source collaboration, combining the benefits of «code is law»[5] with codified law and contract visualization, leading to what we have chosen to call *wise contracts:* smart contracts[6] that use and extend the wisdom of legal and other experts, iteratively learn from experience, and, echoing our title, work for people and machines.

The «Ricardian Contract» paradigm posits three parts of contracts necessary for full automation and legal enforceability – parameters, code and prose.[7] *Parameters* are the aspects that are specific to the particular

1 The authors wish to thank Christopher Clack and his co-authors for their valuable comments on an earlier version of this manuscript. Any errors are our own.

2 The Prize in Economic Sciences 2016.

3 *See, e.g.,* IACCM 2015.

4 The theory of incomplete contracts has been pioneered by Oliver Hart and his collaborators. *See, e.g.,* Hart & Moore 1988. The Nobel Prize in Economics 2016 was awarded to Oliver Hart and Bengt Holmström for their contribution to contract theory, including incomplete contracts.

5 *See, generally,* Lessig 1999 and 2000. *See also* De Filippi/Hassan 2016.

6 We reference the work and vocabulary as developed by Clack et al. 2016(a) and 2016(b). We argue – as those authors do – that it is important to ensure that the smart contract is also a legally enforceable contract, a *smart legal contract*. The authors define the term «smart contract» as an agreement whose execution is both automatable and enforceable (Clack et al. 2016(a)), i.e., to include both *smart legal contracts* and *smart contract code*. Our use of «wise contracts» is intended to refer to the addition of «wisdom» about contracts' business and legal objectives and how they can be reached. The term «wise contract» has a variety of other usages, including an application written to improve coordination among smart contracts (Sztorc 2017) and arbitration services relating to smart contracts (http://www.wisecontracts.com/).

7 Ricardian Contracts is a concept developed by Ian Grigg [Grigg n.d.].

Electronic copy available at: https://ssrn.com/abstract=2925871

contract, for instance deal points such as prices, dates and quantities. A deal point is a fluid notion since any aspect of a contract can become a critical negotiation issue in a particular transaction, but the point is that any aspect of a contract can fit into at least one of these three categories. Smart contracts fulfil the *code* part and are the subject of a strong movement including Hyperledger[8], Corda[9], Ethereum[10] and variants such as Interledger.[11] The *prose* part, however, has, until now, not been handled efficiently. A number of approaches have been developed, many of which can be classified as document assembly. A recent paper in two parts on «Smart Contract Templates» by Clack, Bakshi and Braine (the «SCT Paper»)[12] provides an excellent overview of the issues relating to the codification of prose.

We extend the SCT Paper by providing some suggested solutions to the issues. By handling the prose of legal contracts using the same infrastructure and methods that are used for software code, it is possible to rapidly and universally codify contract prose.[13] CommonAccord[14] demonstrates a simple data model for codified prose that allows easy migration of conventional contracts to standards-based prose objects ready for automation via smart contracts. Codified prose also supports the full set of legal and social methods such as setting standards, commenting and rating.[15] The codification can begin with a form proposed by a party, with a «standard» form or with modular materials from analytical or document assembly systems.[16]

The prose objects can optionally integrate presentation and visual elements, supporting the emerging demands for *truly human-readable* contracts.[17] In recent years, in response to the growing complexity of contracts, new approaches such as simplification, visualization and user-centered design have been introduced, even to the world of commercial contracting.[18] As uses accumulate, visualization can also provide guidance and feedback on uses.[19] The combination of code, codified prose, visual presentation and big data promises a revolution in how contracts are made, managed, and presented.

2. Codifying Prose

Picking up the conversation where the second part of the SCT Paper [Clack et al. 2016(b)] begins, there are five broad requirements for *smart contract templates.* We pair each requirement with a short word that evokes the idea:

1. Editing: Methods to create and edit smart legal agreements, including contract prose and parameters.
2. Transmission: Standard formats for storage, retrieval and transmission of smart legal contracts.
3. Stamping[20]: Protocols for legally executing smart legal agreements (with or without signatures).
4. Binding: Methods to bind the parameters or deal points of a contract and its corresponding smart contract code to create a *smart legal contract*, i.e., a legally-enforceable smart contract.

---

8 https://www.hyperledger.org (all Internet sources accessed on 9 January 2017).

9 https://www.corda.net.

10 https://ethereum.org.

11 https://interledger.org. *See also* Clack et al. 2016(a).

12 Clack et al. 2016(a) and (b). The discussion that follows is mostly based on the latter part, Clack et al. 2016(b).

13 The set of tools that we reference are very well known among software coders. They principally include plain text, HTML, key/values, file systems, git, and editing IDEs.

14 http://www.commonaccord.org/.

15 Standardization of terms is a long-standing and widely-practiced activity.

16 Analytical systems such as RAVN (https://www.ravn.co.uk), Seal Software (https://www.seal-software.com/), and KM Standards (https://kmstandards.com) generate modular materials from groups of conventional contracts. These modules can be formatted as prose objects.

17 *See, e.g.,* Haapio et al. 2017, Haapio 2013 and 2014.

18 *See, e.g.,* Passera 2015, Passera et al. 2016, Waller et al. 2016, Haapio/Barton 2017.

19 Systems such as RAVN (https://www.ravn.co.uk) have demonstrated the ability to graphically present aggregate information about groups of contracts.

20 «Stamping» is our term, the SCT Paper does not provide a one-word label for this principle.

Electronic copy available at: https://ssrn.com/abstract=2925871

5. Enforceability[21]: Methods to make *smart legal contracts* available in forms acceptable according to laws and regulations in the appropriate jurisdiction.

The SCT Paper carefully elaborates issues regarding these requirements. In the following, we suggest that there is a unified approach to achieving these requirements, based on tools and methods already well-established in software development.

2.1. Abstract Specification of a Prose Object Model

The key to templating is what the authors of the SCT Paper refer to as an «abstract specification» for legal templates.[22] Such a system would ideally be expressible in a number of «concrete» formats, such as JSON, XML, markdown, etc.[23] CommonAccord concretely demonstrates such an «abstract» specification that is simple, extensible and unifying. Technically, the model can be described as records consisting of key/values, with links from one record to another that permit inheritance of other key/values based on «prototype» inheritance.[24] This is a technical mouthful. – Operationally, it means that legal templates can be reduced to their constituent parts and assembled into desired forms very simply, as building blocks.

The SCT Paper discusses various design options for storing the parameters – should they be in separate records, with the code or with the prose?[25] We suggest that they will ordinarily be best presented as separate records – term sheets for transactions – but that they can also be stored as part of the prose or the code, and that different uses will call for a mix of these approaches.[26] For instance, the prose forms can include default parameters or leave the parameters blank. In either case, an author of a contract can override it with a new statement of the parameter. Similarly, prose objects can reference or include code and vice-versa.[27] In negotiation, a party can propose changes to an existing prose form by listing the proposed changes. This derivative record can become «the» contract. In practice, key/values that represent parameters, prose or code can all be expressed where it is most convenient.

The abstract specification of key/values and prototype inheritance permits the development of both prose and code «objects» and «classes» that allow highly structured relationships and analysis. By being open-ended (the «prototype» part of inheritance), this model allows anyone – coders, non-coders, lawyers, non-lawyers, including the parties themselves – to rapidly add a preferred form or to adapt an existing one.

2.2. Editing Tools

Returning to the SCT Paper's principles, *editing* can be done with a variety of tools. In a practical system of transacting, most editing will consist of order-entry: filling in parameters or overriding specific spans of prose.

---

21 «Enforceability» is our term, the SCT Paper does not provide a one-word label for this principle, which could also be called «Legitimacy» or «Localization».

22 Clack et al. 2016(b), p. 3.

23 *Id.,* p. 2.

24 On prototype inheritance *generally, see* https://en.wikipedia.org/wiki/Prototype-based_programming. The CommonAccord variant is described most succinctly at https://github.com/CommonAccord/Cmacc-Org/issues/18. Code that precisely implements the abstract model in a concrete form was done by Primavera De Filippi (Harvard Berkman-Klein, and French CNRS) and is available at https://github.com/CommonAccord/Cmacc-Org/tree/master/vendor/library.

25 Clack et al. 2016(b), p. 7.

26 *E.g.,* parameters that complete a prose object – a YCombinator form of investment note. http://source.commonaccord.org/index.php?action=source&file=F/Demo/Note_YC/Cap_Discount.md. The source is available at https://github.com/CommonAccord/Cmacc-Source/blob/master/Doc/F/Demo/Note_YC/Cap_Discount.md.

27 *E.g.,* a form of bank check that incorporates both the prose elements and quasi-code via hash links http://www.commonaccord.org/index.php?action=doc&file=bqc/fr/bnpp/a5we/Account/Check/00001/06-Accept.md. Linking to code via hashes may facilitate the sharing of specific functions among actors. For instance, in the context of the European Payment Services Directive (Directive (EU) 2015/2366), banks will be required to expose payment APIs. They could collaborate on discrete, standardized functions used via hashed names.

This can be done in transaction systems, such as electronic payment systems and web interfaces. That does not preclude interfaces such as those mentioned by the authors of the SCT Paper, including MS Word and document assembly systems. For management of prose objects and extensive editing, we expect that the tools of software developers, such as IDEs,[28] will be preferred. That is what the current authors of CommonAccord use, including Emacs, Visual Studio and Yarn.

2.3. Presentation and Markup

After having been drafted mainly by lawyers for lawyers for years, contracts are now starting to communicate in new ways to new audiences.[29] A visual turn has taken place in many contexts, and contracts have not remained untouched: contracts, too, are shifting from *text-only contracts* (which may or may not contain elements of document design adding clarity) to *visualized contracts*: contracts with embedded images seeking to supplement text and enhance contract readability and usability. In the next few years, prose objects and guided interviews may become the easiest way to generate contracts, whether text-only contracts, visualized contracts, smart or intelligent contracts, or hybrids of these.

The following image, adapted from the work of Haapio, Plewe and de Rooy,[30] shows contracts along the continuum of the ease (or difficulty) of human / machine readability. Research and practice tell us that many people find contract text intimidating and barely human-readable. Smart Contract Templates can provide a graphical user interface to text-only contracts as well as to smart and other code-based contracts, thereby helping the preparation of truly human-readable contracts.

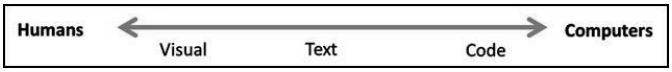

Figure 1. Ease of Human/Computer Readability: Visual-Text-Code Continuum[31]

Smart Contract Templates can also easily provide alternative displays and approaches such as *layering*.[32] This means presenting information so that it can be read first at a summary level that gives an overall understanding, with additional information available if needed.[33] Take the example of Creative Commons licenses.[34] They rely on a three-layer design: first, there are simple, recognizable icons that can be clicked on to reveal a plain-language version of the relevant text. If additional information is required, the full text is available just one click away. The structure includes the so-called Legal Code layer (the «lawyer-readable» version, the full license), the Commons Deed (the «human readable» version summarizing the most important terms), and then a «machine readable» version: a summary of the key terms written into a format that software systems, search engines, and other kinds of technology can understand.[35]

Returning to the issues posed in the SCT Paper, the common convention for presentation elements should, we think, be HTML. HTML is the nearly universal text and graphics language of the web; it is widely understood

[28] An IDE is a tool that developers use for doing their work. It can be compared to word processing and document management for legal work. There are many highly developed IDEs, many of them open source. They are very good at handling a large number of files at the same time, which is normally the case when using codified prose. *See, generally,* https://en.wikipedia.org/wiki/Integrated_development_environment.

[29] Haapio et al. 2017.

[30] *Id.*

[31] *Id.* Image used with the kind permission by the co-authors, Daniela Alina Plewe and Robert de Rooy.

[32] *See, e.g.*, Waller et al. 2016.

[33] *See, e.g.,* Haapio 2014 and Waller et al. 2016.

[34] *See* Creative Commons, http://creativecommons.org/licenses.

[35] «Taken together, these three layers of licenses ensure that the spectrum of rights isn't just a legal concept. It's something that the creators of works can understand, their users can understand, and even the Web itself can understand.» *See* Creative Commons.

among developers and designers and subject to standards processes.[36] HTML is fully capable of expressing legal documents and has a large set of more advanced graphics and presentation features. The inputs, for instance a list of choices, or a summary of the business terms, can be presented with graphic accompaniment and framing. The output – a full-text agreement – or list of transactions, or next steps – can also be presented with embedded icons and expressive formatting. A preference for HTML does not exclude markdown languages, but HTML is a common denominator.[37]

2.4. Transmission

Transmission of smart contracts will be done in a variety of methods. Blockchains, including Bitcoin[38], Ethereum and Hyperledger, have attracted attention and development efforts for their fraud-reduction aspects. Corda has been used in connection with the SCT Paper work. Interledger is another open standard. Current systems of payment also provide secure pathways for transmission of parameters and other key/values. We note that git, the cryptographically-supported version control system developed by the Linux community, provides a medium of transmission that is very useful for collections of materials and particularly for the «legal» part of legal codification.[39]

2.5. Stamping and Signing

Stamping is intended to refer to signature and its equivalents. Signature can be done by conventional methods, including by signing a full text or even a hash of the full text. It is, however, much more efficient and reliable to use the cryptographically-assured methods of Blockchains and payment systems. In these situations, prose objects depend on the code platform for signature.

2.6. Binding

Binding refers to connecting the parameters with the smart contract's prose and code. In a system based on templates, it will generally be possible for parties to adopt only the record that states the parameters, with a reference to the hash of the supporting prose and code. This is in fact very similar to the way that credit card transactions and many other forms of transacting are documented. They are a short statement of parameters that are understood in the context of a system of automation (code) and formal agreements (prose). There is, however, a difference in that a Ricardian model allows rapid iteration and generalization of the system. The record of the transaction will reference the object or objects that support it. If that reference is made using a hash of the referenced record (and all the links in the supporting object are also based on hashes) then there is a one-way inclusion of the full context. This permits rapid development and evolution of the context. It can also be privacy enhancing in case of a security breach; the record itself is reduced to merely the essentials and an opaque hash, while the full context can be proved by the parties, who each have a copy.

2.7. Ontologies, Enforceability and Localization – an Object Model of the Legal World

Ontologies are the essence of a templating system. In computer science, an «ontology» refers to the formally defined concepts and relationships that describe a domain. They include such things as classes, attributes, relationships and rules.[40] In law, these can be thought of as naming conventions and classifications.

The inventory of «things» in a legal conception of the world is of course enormous, as law touches on nearly everything that is important to people. However, by focusing on «documents» as the expression of legal matters, it is possible to make a robust, extensible templating system from only a few «classes» of things. For

36 https://en.wikipedia.org/wiki/HTML.

37 Markdown refers to such syntaxes as those used in Wikipedia and on GitHub, https://en.wikipedia.org/wiki/Markdown.

38 *See, e.g.,* https://en.wikipedia.org/wiki/Bitcoin.

39 Git was originated by Linus Torvalds and developed by the community. https://git-scm.com/.

40 For ontology as a computer science term, *see, e.g.*, https://en.wikipedia.org/wiki/Ontology_(information_science).

contracting, the central class is contract agreements. To allow for the fact that contracts are written in different languages and under different laws and customs, we posit a core «contract agreement» object. It has only a few attributes – an abstract header, identification of parties, signature and the possibility of annexes, for example.[41] This bare object can be extended by cladding it in text for a variety of jurisdictions and languages. The contract core can be configured for any number of parties. The provisions are organized in sections and subsections, which provide a ready-made taxonomy and structure, familiar to many parties and their counsel. Sections and subsections are composed of sentences, which consist of clauses, and may have dates, prices, quantities, maximums and other deal points. In addition to the provisions themselves, the *defined terms* may be parameterized, as may cross-references. By parameterizing all of these elements as key/values and organizing them into records, it is possible to be extremely systematic in document organization and text reuse, while still allowing full flexibility to the parties.[42] Legal codes, like good software code, tend to state each matter once and reference it rather than restate or rework it when needed again. Fixed formulations allow the aggregation of knowledge, including precedents.

There are a few additional document-related classes in contracts. These may include subdocuments such as Statements of Work or Exhibits. These become part of contract agreements, but may also be independent documents. In the other direction, groups of documents may be necessary for a single operation – for instance a *closing binder* of documents for an acquisition in some jurisdictions. In those situations, a single set of parameters (deal points) informs the group of documents. These parameters resemble a term sheet or summary page of a closing binder.

Of course, documents are merely one part of the picture. Legal documents connect and increment the relationships of the parties. They often reference properties and places. They are subject to jurisdictions. As with documents, there can be a core object for «persons.» This includes the notion of name, address, applicable pronouns and the like. Persons can be extended for natural persons (humans) and their varieties, including female and male, juvenile or adult, and for legal persons, in their vast variety. Each of these of course can be expressed in the range of languages, and jurisdictions.

The same is true for places, where, for instance, a particular street address depends on the street, which is associated with one or more municipalities, counties, states, provinces, departments, etc., and nations and supra-national groupings.[43] Properties are similar – a vehicle or patent right has identifying features, may be registered under a jurisdiction, etc. This ontology of prose objects can model the legal world, as expressed in legal documents.

In a global economy, contracts frequently cross boundaries between different legal systems. Even the key legal elements necessary to create a contract vary. A critical part of the object model is localization for the jurisdiction. The contract's choice of law will influence its substance, style and language. In the CommonAccord demonstration materials, jurisdiction has been chosen as the basis to organize contracts and other documents.[44]

2.8. Incremental Codification

Prose objects can be easily and directly adapted from current document practice by the broad community of contracts and legal experts and practitioners. They provide a platform for conventional legal codification, and can advance at the speed of software development. They can be driven by first-movers and go viral, they do not depend on committees to bring everyone to agreement on everything. Codification can be extremely granular

41 http://source.commonaccord.org/index.php?action=doc&file=F/00/Agt/Base/Outline/0.md.

42 For instance, this presents an attempt to organize a complete system of legal documents for a variety of jurisdictions, based on a common core. http://source.commonaccord.org/index.php?action=list&file=F/Demo/.

43 A demonstration object model for geographic locations – http://source.commonaccord.org/index.php?action=list&file=U/at/.

44 http://source.commonaccord.org/index.php?action=list&file=F/Demo/.

– issue by issue, phrase by phrase – and multi-threaded – various solutions can be developed and adopted simultaneously. This is the dynamic by which open source software has come to codify most of computing.

2.9. Intelligence – Expert, Artificial and Natural

The system of prose objects is not «intelligent.» Prose objects are static; they do not reason; they are nouns, not verbs. But these unintelligent prose objects provide anchors for many forms of intelligence. First is human intelligence, such as commercial or legal expertise. Prose objects are consistent with conventional codification efforts, model document projects, commentaries, and trade group efforts. The prose objects also fit with software code in the Ricardian paradigm, so coders can use smart contracts to automate interactions, without having to automate everything about a contract. Since the great majority of contract interactions seen from a legal point of view fall into definable patterns, most of contract interactions can be automated with code, while relying on the legal system to deal with the unusual, unexpected events that reflect, for example, the incompleteness of contracts. This is how most automated systems work, and the model can be generalized in the Ricardian paradigm.[45]

With respect to AI and big data intelligence, the prose objects provide structured expressions of the relationships. The patterns in contracting behaviour can be deduced from the relationships among the prose objects, their frequency and variations. By analyzing this data, some intelligent systems will come to offer the equivalent of self-driving bureaucracies. They will identify the pathways to achieving the goals of constituents, the costs and tradeoffs. This is already part of the goals of many AI systems. Extensive use of prose objects may allow the reasoning to be more easily understood and controlled by humans.

3. Conclusion

This paper illustrates how prose objects can serve as the connection between automated systems and human systems. They can provide anchors for a number of forms of explanation and clarification, including layering, graphic representations, layout, style sheets, voice recognition, ratings, commentary and the like – all depending on the needs and expectations of the audience.

The vision combines conventional contract drafting processes, web development and business process automation, transforming contracts into codified text and images, appropriate for the needs of different audiences. The goal is a public, open source collaboration for codified contracts that both people and machines can easily understand, improve, and act upon.

4. References

Clack, Christopher D./Bakshi, Vikram A./Braine, Lee, Smart Contract Templates: foundations, design landscape and research directions. Version v2, 3 August 2016(a). https://arxiv.org/abs/1608.00771.

Clack, Christopher D./Bakshi, Vikram A./Braine, Lee, Smart Contract Templates: essential requirements and design options. Version v2, 15 December 2016(b). https://arxiv.org/abs/1612.04496.

Creative Commons, About The Licenses – Creative Commons, http://creativecommons.org/licenses.

De Filippi, Primavera/Hassan, Samer, Blockchain technology as a regulatory technology: From code is law to law is code, First Monday Volume 21, Number 12, 5 December 2016, http://firstmonday.org/ojs/index.php/fm/article/view/7113.

Grigg, Ian, The Ricardian Contract, no date, http://iang.org/papers/ricardian_contract.html.

Haapio, Helena, Designing Readable Contracts: Goodbye to Legal Writing – Welcome to Information Design and Visualization. In: Schweighofer, Erich/Kummer, Franz/Hötzendorfer, Walter (Eds.), Abstraction and Application. Proceedings

[45] Merchant websites are largely automated, but have terms of use and complaint procedures. Stockmarkets are similarly configured. The system works well most of the time and occasionally a problem arises that is dealt with in conference rooms and courts. The parties learn and incrementally so does the system. The benefit of an open source, generalized approach, such as is offered by smart contracts and by prose objects, is that the learning can be widely and symmetrically shared.

of the 16th International Legal Informatics Symposium IRIS 2013, Österreichische Computer Gesellschaft OCG, Wien 2013, p. 445–452.

Haapio, Helena, Lawyers as Designers, Engineers and Innovators: Better Legal Documents through Information Design and Visualization. In: Schweighofer, Erich/Kummer, Franz/Hötzendorfer, Walter (Eds.), Transparency. Proceedings of the 17th International Legal Informatics Symposium IRIS 2014, Österreichische Computer Gesellschaft OCG, Wien 2014, p. 451–458.

Haapio, Helena/Barton, Thomas D., Business-Friendly Contracting: How Simplification and Visualization Can Help Bring It to Practice. In: Jacob, Kai/Schindler, Dierk/Strathausen, Roger (Eds.), Liquid Legal – Transforming Legal into a Business Savvy, Information Enabled and Performance Driven Industry, Springer International Publishing 2017, p. 371–396. DOI: 10.1007/978-3-319-45868-7_24.

Haapio, Helena/Plewe, Daniela Alina/de Rooy, Robert, Contract Continuum: From Text to Images, Comics and Code. In: Proceedings of IRIS 2017.

Hart, Oliver/Moore, John, Incomplete Contracts and Renegotiation, Econometrica, Vol. 56, No. 4, 1988, p. 755–785.

IACCM, Commercial Excellence: Ten Pitfalls To Avoid In Contracting. [Booklet] International Association for Contract and Commercial Management 2015, https://www2.iaccm.com/resources/?id=8451.

Lessig, Lawrence, Code and Other Laws of Cyberspace, Basic Books, Inc., New York, NY, 1999.

Lessig, Lawrence, Code Is Law: On Liberty in Cyberspace, Harvard Magazine, 1 January 2000, http://harvardmagazine.com/2000/01/code-is-law-html.

Passera, Stefania, Beyond the Wall of Text: How Information Design Can Make Contracts User-Friendly. In: Marcus, Aaron (Ed.), Design, User Experience, and Usability: Users and Interactions, Springer International Publishing 2015, p. 341–352. DOI: 10.1007/978-3-319-20898-5_33.

Passera, Stefania/Smedlund, Anssi/Liinasuo, Marja, Exploring contract visualization: Clarification and framing strategies to shape collaborative business relationships, Journal of Strategic Contracting and Negotiation, Vol. 2, Issue 1–2, March/June 2016, p. 69–100. DOI: 10.1177/2055563616669739.

The Prize in Economic Sciences 2016. Nobelprize.org. Nobel Media AB 2014. http://www.nobelprize.org/nobel_prizes/economic-sciences/laureates/2016/.

Sztorc, Paul, Upgrading «Smart Contracts» to «Wise Contracts», 15 January 2017, http://bravenewcoin.com/news/upgrading-smart-contracts-to-wise-contracts/.

Waller, Robert/Waller, Jenny/Haapio, Helena/Crag, Gary/Morrisseau, Sandi, Cooperation Through Clarity: Designing Simplified Contracts, Journal of Strategic Contracting and Negotiation, Vol. 2, Issue 1–2, March/June 2016, p. 48–68. DOI: 10.1177/2055563616668893.