

Two Sides of the Same Coin. Possible Interactions Between Text-written Law and Computer Code in the Near Future

Patryk Ciurak

1. Introduction

Even though it is already the third decade of the 21st century, digitalization has touched and transformed almost every part of our lives, and the Internet has become the equivalent of mankind's nervous system, the law has not changed its form. The words that form sentences written in a specific legal language still regulate the life and functioning of societies, and their interpretation allows different meanings to be attributed to seemingly identical expressions. Alongside this established order, which is based on the human understanding of justice, a completely different, parallel reality has grown up, where the prevailing language is computer code. With increasing computerization, the code has come to describe various aspects of human activity, regulating them more efficiently than the law due to its unambiguous expressions and speed at which it is executed. However, the same features pose a problem when reality changes. This is when the law prevails (if it has been well designed), providing sufficiently flexible and interpretable rules of conduct. According to M. Hildebrandt, this is the main advantage that law (written in natural language) has over regulations written in computer code.¹

The consequence of the dualism described above is the existence of countless computer systems regulating our reality and influencing our

1 'The rule of law, understood as an institution ensuring that nobody is above the law, while offering sufficient foreseeability as well as contestability, requires legal norms that build on the open texture of natural language, avoiding both the over- and under-inclusiveness of disambiguated computer code. For now, that means we should foster the adaptive nature of text-driven law before exchanging it for the code-driven nature of computational law. It also means that we should welcome computational technologies that contribute to challenging legalism, authoritarian rule by law and arbitrary rule by those in power.' Mireille Hildebrandt, 'The adaptive nature of text-driven law' (2021) 1(1) *Journal of Cross-Disciplinary Research in Computational Law* <<https://journalcrcl.org/crcl/article/view/2>> accessed 26 April 2021.

lives. Their creation was made possible, among others, by the translation of law into computer code. However, the rules of translation are not formalized anywhere, and it is not the legislator who decides on the final implementation of the law, but the programmers responsible for developing the code. Because of differences in design, systems are often incompatible with each other, even though they operate under the same law. Their maintenance also generates significant costs as there is no single source of truth that would provide an official version of code representing the current law. A possible solution to this problem may be the application of artificial intelligence that could apply the text-written law. However, whether one is thinking about contract analysis, similarity searches, or predictive analytics outcomes, at the heart of all these problems (and a few more) is the fact that computers do not understand the law written by humans and for humans to be interpreted and applied. To achieve a real fusion of the two realities - legal and IT - the law should evolve into two forms existing in parallel: rules written in natural language and computer code.

When we refer to the law in the form of computer code this can be attributed both to *machine-readable* legislation as well as legislation that is de facto a computer program ready to be executed and thus to produce legal or factual effects - *machine-consumable* legislation.² Both forms are already widely used, although only in the case of machine-readable legislation is this evident: various normative acts are being promulgated in form of .xml or .pdf files or presented in legal databases. Machine-consumable legislation, on the other hand, is a mapping of the applicable law in the code of the aforementioned applications and systems used to support the processes of law enforcement, e.g. by public administration. Importantly, while the promulgation of a normative act in electronic form may be the result of an official legal procedure (as it is in Poland)³ machine-con-

-
- 2 Accident Compensation Better Rules Discovery Team, 'Exploring Machine Consumable Accident Compensation Legislation. Lessons for a structural rewrite of the AC Act and opportunities to make it machine consumable' (The Service Innovation Lab, 1 July 2019) 18 <https://serviceinnovationlab.github.io/assets/Exploring_Machine_Consumable_Code_With_ACC.pdf> accessed 30 January 2021 ; James Mohun and Alex Roberts, 'Cracking the code. Rulemaking for humans and machines' (2020) 40 OECD Working Papers on Public Governance 5 <<https://doi.org/10.1787/3afe6ba5-en>> accessed 5 February 2021.
 - 3 The obligation to promulgate a legal act in electronic form established in Article 2a of the Act of 20 July 2000 on the promulgation of normative acts and certain other legal acts (Journal of Laws of The Republic of Poland of 2019, item 1461).

sumable legislation is, in general, created by private entities without any general rules regulating the process of law-making.

In the following paper, the law existing simultaneously in the form of text and computer code will be referred to as Rules as Code (hereinafter RaC)⁴. Following the systematic proposed by M. W. Wong, RaC referred to in this paper will correspond to solutions placed at level 3 or level 5⁵ which makes them similar (to a certain degree) to the concept of a smart contract.⁶ Furthermore, the possibility of immediate execution from the moment the RaC is created has to be considered as a constitutive feature, which is another point in common with the smart contract. Human intervention should be limited only to the introduction of certain values (if they cannot be obtained from other sources) or the final approval of the proposed ruling.

The aim of the paper is to consider the possible mutual influence that code and legal text may have in the near future. Without a doubt, the development and execution of code will be influenced by the rules of working with legal text, while the process of drafting the law in natural language can be improved by the application of selected code development practices. This paper explores the possibility to automate certain phases of legal interpretation with computer code as well as adopting software testing practices to ensure the quality of legal drafting. The final effect

4 The term Rules as Code has also been adopted in papers: Meng Weng Wong, 'Rules as code – Seven levels of digitisation.' (Research Collection School Of Law, April 2020) <https://ink.library.smu.edu.sg/sol_research/3093/> accessed 17 April 2021 and Mohun and Roberts (n 2).

5 In his paper *Rules as code - Seven levels of digitisation*, M.W. Wong discusses the levels of representation of law by code. Level 3 solutions consist of three layers: rules - natural language sentences, translation of rules into a form understandable by IT systems, and a rule engine which task is to perform the above. The solutions allow (supposedly) users without programming knowledge to edit the rules, while not forcing changes to the code on the engine side. Level 5 involves parallel creation of provisions in the form of natural language and code. However, the natural language text is created from the code (rather than the other way around, as might be expected) and, depending on the sophistication of the system, may require tweaking by the legislator. It should be emphasised that the code version of legislation can be considered as an authentic text on a par with the natural language version or can even fully replace it. *ibid.*

6 A detailed description of the smart contract is provided by Dariusz Szostek, *Blockchain and the Law* (1 ed., Nomos 2019) 110-35.

should present the same meaning whether it comes to text-written law or code, which is referred to as isomorphism.⁷

2. *Interpretation of Law and Interpretation of Code*

Ensuring isomorphism seems to be the biggest challenge standing in the way of widespread use of RaC. Text and code must present the same information, while the way they are decoded is significantly different. Pointing out the most obvious differences (and at the same time simplifying somewhat): a legal text is subject to interpretation, while a code is (in general) unambiguous. Interpretation of a text should aim at realizing a notion of justice that is difficult to define precisely, whereas a code aims at realizing some specific goal. The code is just executed, taking inputs, transforming them, and generating outputs without referring to external axiology (moral or other). Legal language deliberately uses vague or discretionary terms, primarily to ensure an appropriate level of flexibility in the law. Meanwhile, the code should be unambiguous (as mentioned above), functioning within a certain well-defined framework. Finally, the interpretation of the text and the application of the norms are separated by a certain time interval (more or less), while the code is executed and has legal effects in principle immediately; there is no so-called hermeneutic gap, which allows for the assessment of the adequacy of the norms and the possible suspending of their implementation.⁸

Is it therefore possible for the law to function in the form of a code? The analysis of the problem may begin with considering what kind of regulations may take the form of a computer code? J. Mohun and A. Roberts⁹ point to prescriptive rules, which are mostly unambiguous and thus do not require much interpretation. Moreover, they should be used frequently and by a large number of entities, and their functioning in the form of a code will bring specific, measurable benefits (e.g. reduction of the business costs).¹⁰ Moreover, authors repeatedly refer in their publication to services

7 Trevor J. Bench-Capon and Frans P. Coenen, 'Isomorphism and legal knowledge based systems.' (1992) 1 *Artificial Intelligence and Law* 65.

8 cf Laurence Diver, 'Digisprudence: the design of legitimate code.' (LawArXiv Papers, 14 July 2020) <<https://doi.org/10.31228/osf.io/nychu>> accessed 18 March 2021.

9 Mohun and Roberts (n 2) 92.

10 This last feature is not substantive and seems to have been added mainly to economically incentivise actors to implement RaC.

provided to i.e. citizens as the possible subject of regulation. Thus, RaC is supposed to improve the satisfaction of standard needs reported by selected categories of entities.

In the paper *Better Rules for Government Discovery Report*, authors working in The Service Innovation Lab in New Zealand suggest (based on their experience) the possibility of transforming into RaC provisions that:

- contain the formulas needed to carry out the calculations,
- confirm the existence of the entitlement, the constitutive features, or the opportunity to examine the application,
- regulate standardized, repeatable processes
- describe the steps in a certain process that must take place before a finding of legal compliance can be made (area of compliance)
- describe a process that can be implemented immediately in digital form.¹¹

The importance of public services has been recognized in the proposal for a regulation of the European Parliament and of The Council laying down harmonised rules on Artificial Intelligence (Artificial Intelligence Act) and amending certain Union legislative acts (hereinafter the proposal).¹² Recital 3 of the proposal points directly to public services (e.g. applying for or receiving public assistance benefits) and justice as areas of interest that can benefit from the use of AI systems. Moreover, AI systems that ‘(...) are used for determining whether such benefits and services should be denied, reduced, revoked or reclaimed by authorities (...)’ will be considered as high-risk as their use ‘(...) may have a significant impact on persons’ livelihood and may infringe their fundamental rights, such as the right to social protection, non-discrimination, human dignity or an effective remedy.’¹³ RaC could (and should) be used to provide clear and coherent regulations to implement in high-risk AI systems. Otherwise, leaving the translation of text-written law into computer code exclusively to high-risk AI systems providers will increase the risk of emerging gaps

11 The Service Innovation Lab (LabPlus), ‘*Better Rules for Government Discovery Report*’ (NZ Digital Government, March 2018) 27 <<https://www.digital.govt.nz/dmsdocument/95-better-rules-for-government-discovery-report/html#summary>> accessed 31 January 2021.

12 <<https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=COM:2021:206:FIN>> accessed 31 January 2021.

13 See recital 37, Article 6(2) and Annex III(5) of the proposal.

and inconsistencies between both versions¹⁴ as well as creating multiple translations. The consequence will be an incompatibility between systems, as described above, and a lower chance of meeting the requirements of accuracy and robustness as mentioned in the proposal.¹⁵

To sum up: the standardization and repeatability of the process, the mass usage, and the unambiguity of the regulations (or at least reduced need for interpretation) point to the possibility of implementing regulations in the form of RaC. Potential benefits in form of reducing operating costs or speeding up processes in an organization are an additional encouragement, important from an economic point of view. However, it should be remembered that RaC is at a very early stage of development. The aim of creating solutions according to the guidelines described above is to provide so much needed practical knowledge and to precede the preparation of more complex projects. It is necessary if RaC is ever to have a real application. Otherwise, it will remain one of the many ideas that have ended their existence at the proof-of-concept stage.

Another area where RaC can certainly find application is the regulation of cyberspace. S. Shcherbak notes that '(...) everything that one sees on the Internet is delivered by means of code. Therefore, code is the architecture of cyberspace, and pieces of code are a construction material of this architecture.'¹⁶ This thought is developed by D. Szostek concerning the regulation of artificial intelligence:

The traditional way of promulgating, or enforcing, a regulation that includes orders or prohibitions that should be taken into account by AI is doomed to failure in advance, a point raised by experts who suggest that the regulation of AI is one of the more difficult challenges for modern lawyers. Assuming, however, that the regulation would be a computer code containing the orders and prohibitions necessary to be considered by AI, in technical terms it represents an organizational and logistical challenge, but not an impossible one. AI is a code that is part of cyberspace, so only by code can prohibitions and orders be imposed on AI, thereby regulating it.¹⁷

14 The identification of such gaps has been described by The Service Innovation Lab (LabPlus) (n 11) 10.

15 See recital 49 and Article 15 of the proposal.

16 Sergii Shcherbak, 'Integrating Computer Science into Legal Discipline: The Rise of Legal Programming,' 4 (14 September 2014) <<https://ssrn.com/abstract=2496094>> accessed 5 February 2021.

17 Dariusz Szostek, 'Sztuczna inteligencja a kody. Czy rozwiązaniem dla uregulowania sztucznej inteligencji jest smart contract i blockchain?' in Lai L. and

Research on the RaC should therefore seek to set standards for law-making that will allow the effective regulation of cyberspace (understood broadly), in particular the systems and algorithms on which AI is based.

As for now, it is unlikely that the European Artificial Intelligence Act (the Regulation) will take the form of RaC. Still, the proposal mentions a quality management system as a measure to provide compliance of high-risk AI systems with the Regulation. It is possible that some of its components may refer to other regulations as well and therefore could benefit from the implementation of RaC. As a result, some of the quality management processes could be automated (to a point), as the official version of the RaC could be made publicly available i.e. through application programming protocol (API).¹⁸ Making RaC available in this manner could also facilitate internal quality checks in the process of creating and maintaining a high-risk AI system.

The effectiveness of regulation, however, must not be equated with a complete lack of interpretation. The research should also cover less clear provisions that contain more discretionary elements or refer to immeasurable values, vague concepts, or general clauses. It should not be expected that the existence of law in the form of a computer code will eliminate the above-mentioned elements from legal acts.¹⁹ The RaC presupposes the coexistence of the legal text and the code, rather than striving to reduce the entire law to a syllogism that requires only the substitution of relevant values. This is at odds with the idea of the RaC and is simply not feasible.

Świerczyński M. (eds.), *Prawo sztucznej inteligencji* (C. H. Beck 2020) 21. In the text, the author refers to the regulation of artificial intelligence by means of a smart contract. Given that the constitutive feature of both RaC and smart contract is readiness for immediate execution, the quoted passage should be considered true in both cases.

18 cf Wong (n 4) 13.

19 For this would mean a return to prescribing casuistic, elaborate legislation. At the same time, it should be noted that it is not a problem for a computer to execute instructions hundreds or thousands lines of code long. Normative acts could finely regulate complex cases in an attempt to replicate reality as accurately as possible; they would not have to be simple, they just need to be consistent and suitable for writing in code (see The Service Innovation Lab (LabPlus) (n 341) 25). A negative consequence would be the difficulty in maintaining the consistency of the RaC. This can be avoided by introducing testing procedures that have been used for many years in software development. Moreover, assumption that RaC are more concise than the same rules written solely in natural language proven to be false (cf Accident Compensation Better Rules Discovery Team (n 2) 6, 30).

It is therefore necessary to identify as many common features as possible between the execution of a computer code and the interpretation (and applying) of legal rules, even if the task seems extremely difficult. Certain interpretative directives are executed based on parameters that are completely measurable and, as a result, lend themselves to a description through a code. Referring to M. Zieliński's conception of interpretation,²⁰ the interpretative moment²¹ may be determined based on the date of the event initiating proceedings, recorded in system logs. The same date will constitute the basis for selecting the correct temporal version of the provisions (both in text and code form) requiring interpretation. Combined with version management in a manner similar to the one used in code development, it would be possible to immediately select the correct version of the legal act for the chosen interpretative moment. Therefore, the ordering phase of interpretation²² would be mostly automated.²³

The situation becomes slightly more complicated when one moves to the reconstructive phase of interpretation. Following M. Zieliński it should be assumed that:

The directives of the reconstructive phase are aimed at obtaining from the provisions of different syntactic form a norm-shaped expression with a structure: A, O, n/z, Z,²⁴ which is to facilitate the determination of the sense of norms expressed in this provision.²⁵

The principle is further formulated, according to which:

The process of interpretation proceedings in the reconstruction phase should begin with the recognition of those syntactic features of the

20 This conception of interpretation has been presented by M. Zieliński in his book *Interpretation of law. Principles - rules - guidelines* first published in Poland in 2002. Maciej Zieliński, *Wykładnia prawa. Zasady – reguły – wskazówki* (7 ed., Wolters Kluwer 2017).

21 *ibid* 279.

22 As described by M. Zieliński. *ibid* 281-258.

23 The implementation of RaC by state bodies would open the way to official assessment of the validity of legislation, similar to what is currently done in commercial legal databases (i.e. LEX or Legalis in Poland). The algorithms adopted by the editors of each system differ significantly and in consequence the same legal act may be indicated as binding in one database, while the other database will not mark it so. This solution is often a source of confusion and causes significant problems for less experienced lawyers.

24 Where A corresponds to the addressee of the norm, O to the circumstances, n/z to the order/prohibition, Z to the behaviour. Zieliński (n 20) 20.

25 *ibid* 286.

provision being interpreted, which determine the trends of further treatment of the provision to obtain from it a single norm or several norms.²⁶

The measures described above have been designed for the interpretation of text written in legal language. However, their applicability to computer code initially raises doubts. Nevertheless, to execute a given piece of code, all necessary elements and their relationships must also be identified. This occurs spontaneously when the code is executed, and deficiencies in this regard are reported by the program (in general) as errors. Thus, the activities of the reconstruction phase can, to some extent, be contained directly in the logic of the code itself. It must be coherent, non-contradictory, and, like a text in legal language, it should not contain redundant expressions.²⁷

Thus, a norm-shaped expression can be described, identified, and reconstructed using code. For example, in an object-oriented programming language such as Python,²⁸ a reconstructed norm-shaped expression could be an object created within a distinct class of objects. The attributes of this class would be the addressee of the norm (A) and the circumstances (O), while the methods in the class would be used to determine the pattern of ordered or forbidden (n/z) behaviour (Z). The modifiers described by M. Zieliński,²⁹ on the other hand, would be functions not directly connected with any class. A norm-shaped expression as an object would be reconstructed for the needs of a particular case and after its application, that is after the code is executed and legal effects are produced - it would be deleted. Also, it should not be assumed that all the mentioned elements must fit in a single provision. In the same way, as legal definitions or references are used in the text of a legal act, the Python code (and many others) refers to other fragments of code or uses so-called class inheritance, which allows creating new classes that are extensions of already existing ones.³⁰

26 *ibid.*

27 An additional benefit is that errors or loopholes can be detected by code testing if the two versions are kept equal. This issue is discussed in more detail later in the text.

28 More information about the Python language and its documentation are available at <https://docs.python.org>.

29 Zieliński (n 20) 111.

30 Please keep in mind that the above description is very much simplified and represents only an outline of the concept. Further research into the possible interactions between text and code is needed in order to develop optimal solutions.

However, to reconstruct the norm-shaped expression, a sufficiently broad set of input information necessary for the decision should be obtained at the earliest possible stage of the proceedings. Some of them will have to be provided by the entity initiating the proceedings (as it is now), others may come from the databases owned by the public administration (e.g. official registers from which data is retrieved through an appropriate interface). The scope and type of data collected would reflect the state of knowledge on a given subject (both general and specialized), the objectives of introducing a given regulation, and (if possible) other values that the legislator considered necessary to take into account at a given stage. As a result, the norm-shaped expression would be reconstructed from mostly objective information.

By far the most difficult part is to combine legal text interpretation and code execution at the final, perceptual stage.³¹ Following linguistic or systemic directives to determine the meaning of individual phrases of a norm-shaped expression is not that difficult; it may even be facilitated by the use of a code. However, the application of functional directives presents a significant difficulty and will in principle be impossible without human intervention.

The application of the linguistic directives of the perceptual phase can be regarded as similar to the process of code execution. Determining the meanings of the individual elements of a norm-shaped expression and their relationships to each other is akin to a computer checking³² that the code is complete and free of obvious errors; this stage precedes code execution. Specific values or attributes are established for specific expressions in the code, just as the meaning of individual words or clusters of words is determined when following linguistic directives. The use of references to other places in the code corresponds to the determination of the meaning of individual words by reference to legal definitions formulated earlier. These definitions, both content- and scope-related,³³ can also be mapped using the code, specifying the necessary input that indicates that the given expression (object) falls within the scope of the definition.

Despite the similarities between the application of linguistic interpretation directives and code enforcement described above, there still are some differences difficult to overcome. This is the case when in the course of in-

31 Zieliński (n 20) 290-302.

32 Specifically, by an interpreter, that is, by a computer program that executes the indicated programs.

33 Zieliński (n 20) 293.

terpretation one has to take into account someone else's binding decision (e.g. a Supreme Court ruling), or when a provision contains legal language phrases with an established meaning in the legal language (e.g. property) or factual phrases (e.g. cat).³⁴ Furthermore, in the case of application of the systemic directives, it should be pointed out that the norms-rules are often formulated with the use of general, unmeasurable terms, or even do not function as separate provisions, but are derived from the legal text through interpretation.

In the case of systemic directives,³⁵ vertical consistency can be enabled by the code architecture itself. Its hierarchical composition makes it easier to ensure the compliance of higher-order norms with lower-order ones. Possible exceptions to the rules can be introduced even through a basic if/then/else instruction. The difficulty arises with the need to ensure horizontal compliance with the norms-rules³⁶ of the legal system. This poses a serious problem at the moment and should be subject to further research.

By far the most problematic application is that of functional directives, in the form in which they currently operate.³⁷ Possible, and sometimes even necessary, references to sources outside the legal act or even completely outside the legal system make it impossible to ascribe meaning to individual elements of a normative expression using the code. Therefore, one of the aims of RaC research should be to determine the optimal way to map the functional directives of interpretation.

The existence of rules in two equal forms (text and code) does not mean the exclusion of human involvement. In the initial stages of the application of the RaC, human participation in the process and its supervision of the proceedings should be assumed. Solutions involving the application of the law, including dispute resolution, by artificial intelligence are only in the testing phase.³⁸ Until satisfactory results are obtained, algorithms should not be allowed to directly shape the legal situation of entities (due to low accuracy of decisions or ethical concerns). The role of the human supervisor would be threefold. First: to compare the results of the

34 *ibid* 294-295.

35 *ibid* 297-298.

36 More on norms-principles see *ibid* 34-36.

37 *ibid* 299-300.

38 Examples are Estonia and China. cf Maria Dymitruk, 'Sztuczna inteligencja w wymiarze sprawiedliwości?' in Luigi Lai and Marek Świerczyński (eds), *Prawo sztucznej inteligencji* (C. H. Beck 2020); Joshua Park, 'Your Honor, AI.' (*Harvard International Review*, 3 April 2020) <<https://hir.harvard.edu/your-honor-ai/>> accessed 5 February 2021.

linguistic and systemic interpretation obtained as a result of the execution of the code with the results of its functional interpretation and to make a final decision on the reconstructed norm-shaped expression. Then, to make discretionary decisions during the process, if required by the legislator,³⁹ and (finally) to come to a decision. Each of the described actions may in practice be divided into several lower-level decisions, which have their premises and adjudication. This solution may seem far from the idea of RaC, as it assumes the decisive participation of a human being in the process and resembles more the functioning of the so-called virtual assistant judges based on artificial intelligence. Nevertheless, the ordering and reconstruction phases will base on the execution of a code, just as for the determination of meanings using the directives of linguistic and systemic interpretation. These are activities at the level of data and information and (at least in part) knowledge. Describing them through code can result in significant time savings and less chance of error than if performed by a human. A supervisor will be able then to concentrate on the most human stage of the procedure - the application of functional interpretation directives requiring wisdom.⁴⁰ Moreover, the suggested solution aligns with the requirement of human oversight as provided by the proposal in recital 48 and Article 14. Assuming the proposal will come into force without significant changes in this matter, introducing the framework for the interpretation of RaC will be obligatory, at least for the providers of high-risk AI systems.

In parallel with subsequent decisions, a database of rulings will be created where the factual state, the legal state, the lower-level decisions described above, the final decision, and the relations between them are described in code. Such a database can be used for training machine-learning models and will provide high-quality data as mentioned in recital 44 and Article 10 of the provision. The emergence of artificial intelligence that can learn to apply law faster, more effectively, and accurately than today

-
- 39 Paradoxically, the degree of human interference in the process of applying the law can be a measure of the quality of provisions; see Accident Compensation Better Rules Discovery Team (n 2) 33. A law that is clear and logically coherent will require minimal human intervention, and in extreme cases it may not be required at all. By contrast, regulations that rely heavily on discretion, are internally contradictory or inconsistent with the rest of the legal system will not be able to function as a code.
- 40 cf Jennifer Rowley, 'The wisdom hierarchy: representations of the DIKW hierarchy' (2007) 33(2) *Journal of Information Science* 163. By wisdom one would also mean making ethical choices.

will be a matter of time. Whether this artificial intelligence will be allowed to apply the law to humans or will be only a more accurate and robust high-risk AI system remains a matter for debate.⁴¹ Nevertheless, translating the law into a language that can be understood by artificial intelligence seems like one of the biggest gains that RaC can offer.

The above considerations are only a sketch of the broader research problem of reconciling legal interpretation and its rich theoretical heritage - activities inherent in working with natural language text - with the development and execution of computer code. Without in-depth analysis, accompanied by the creation of prototypes, there is no chance of convincing a wider range of legal practitioners, let alone the legislator, of the feasibility of implementing RaC. The work should also cover other types of interpretation, in particular operative interpretation due to its role in the practice of law application.⁴² Some attempts have already been made to outline the standards that should be met by a code carrying legal norms, although they were motivated by determining the requirements for the legality of smart contracts (primarily decentralized autonomous organization, DAO).⁴³

3. Testing Code and Testing Law

RaC can be described simply as a set of rules of conduct. Undoubtedly, before implementing any rules it must be ensured that they will work properly and thus achieve the intended purpose. It is therefore necessary to carry out appropriate testing procedures.

41 As for now, see recital 40 of the proposal: '(...) In particular, to address the risks of potential biases, errors and opacity, it is appropriate to qualify as high-risk AI systems intended to assist judicial authorities in researching and interpreting facts and the law and in applying the law to a concrete set of facts. Such qualification should not extend, however, to AI systems intended for purely ancillary administrative activities that do not affect the actual administration of justice in individual cases, such as anonymisation or pseudonymisation of judicial decisions, documents or data, communication between personnel, administrative tasks or allocation of resources.'

42 The concept of operative interpretation is presented in detail in the articles by L. Leszczyński. cf Leszek Leszczyński, 'Wykładnia operatywna (podstawowe właściwości)' (2009) 6 Państwo i Prawo 11; Leszek Leszczyński, 'O wykładni prawa i jej wymiarze praktycznym. Kontekst sądowego stosowania prawa.' (2020) 2 Archiwum Filozofii Prawa i Filozofii Społecznej 66.

43 Diver (n 338).

At present, there is no single, coherent methodology for quality assurance of draft regulations in the Polish legal system. Certain activities aimed at testing regulations take place at different stages of the process and different levels of detail. For example, for legislation initiated by the government, the quality of drafted regulations is to be ensured by the impact assessment process, which consists of regulatory impact assessment and ex-post regulatory impact assessment. These activities are complemented by public consultations.⁴⁴ Even though their assumptions are described in detail,⁴⁵ the convergence with existing guidelines and the quality of the activities undertaken is often insufficient, as confirmed by the Supreme Chamber of Control audit.⁴⁶ Therefore, it can be concluded that at least part of the high-level procedures for ensuring the quality of law is defective. Another precaution is the law commission appointed during the legislative work carried out on the government side. The commission is responsible, among other things, for the quality check of the legislation. Nevertheless, the possibility to exempt a draft from the commission's consideration⁴⁷ may be thought of as a breach in the procedure for assuring the proper quality of draft legislation.

At the same time, lawyers not connected to the government (legal counsels, attorneys, or legislators) and working on drafts of normative acts (both generally and internally binding) conduct on their own various simulations and thought experiments. These may take place both in close cooperation with the entities commissioned to draft the project and without their participation. Sometimes an iterative approach is used (assuming the existence of several rounds of tests) as well as an incremental one (consisting of gradual coverage of the draft regulations with tests); tests may

44 Departament Oceny Skutków Regulacji, 'Ocena wpływu w rządowym procesie legislacyjnym.' (Gov.pl, 13 November 2020) <<https://www.gov.pl/web/premier/ocena-wplywu-w-rzadowym-procesie-legislacyjnym>> accessed 18 April 2021.

45 Ministerstwo Gospodarki and Kancelaria Prezesa Rady Ministrów, 'Wytyczne do przeprowadzania oceny wpływu oraz konsultacji publicznych w ramach rządowego procesu legislacyjnego.' (Rządowe Centrum Legislacji) <<http://www1.rcl.gov.pl/?q=book/wytyczne>> accessed 5 February 2021.

46 Najwyższa Izba Kontroli, 'Dokonywanie oceny wpływu w ramach rządowego procesu legislacyjnego.' (Najwyższa Izba Kontroli, 5 March 2018) <<https://www.nik.gov.pl/plik/id,16190,vp,18712.pdf>> accessed 5 February 2021.

47 cf Maciej Berek, 'Rządowa procedura prawodawcza i jej znaczenie dla jakości stanowiącego prawa.' in Federczyk W. and Peszkowski S. (eds.), *Doskonalenie i standaryzacja procesu legislacyjnego – dobre praktyki opracowane w ramach projektu LEGIS (Krajowa Szkoła Administracji Publicznej im. Prezydenta Rzeczypospolitej Polskiej Lecha Kaczyńskiego 2019)*.

also be carried out gradually, as subsequent parts of the draft regulation emerge. Again, however, there is no methodology for conducting quality control; the measures taken are the know-how of individuals or entities, and testing is not obligatory.

When a gap in the law is discovered, its proper application is possible (in general) because of appropriate interpretation. It plays a role of a last line of defence, an ex-post measure taken to mitigate potential damage, and may be a premise for amending the act. Although it is a natural process, it also consists of actions that only take place when a problem arises. As a result, they may negatively affect the confidence in the state and the law it creates, and in a more tangible aspect - generate higher costs than thorough testing of the proposed regulations before they enter into force.

The introduction of RaC creates a valuable opportunity to adapt the quality assurance practices known from the development of computer code to the legislative practice. This applies both to tests carried out during code development and the subsequent stage of acceptance by end-users. At this point, it should be noted that although the scope of topics related to code quality assurance far exceeds the volume of this paper, the author would like to briefly discuss the most important issues.

Assuming that the law is drafted in parallel in two forms: code and natural language, the need for code testing would facilitate simultaneous testing of the language form. Thus, at a very early stage of work, it would be possible to detect errors, potential gaps and determine whether the proposed legislation achieves the intended purpose. As a result, the number of amendments issued to improve the law could significantly decrease, as could the overall cost of legislative activities. A positive side effect would be an increase in stability and certainty of the law in force.

Testing RaC, like any computer code, can increase consistency and reduce conflicts with previous regulations by performing so-called integration and regression tests.⁴⁸ Similar actions are currently undertaken during

48 Integration testing can be divided into component integration testing which '(...) focuses on the interactions and interfaces between integrated components' and system integration testing which '(...) focuses on the interactions and interfaces between systems, packages, and microservices. System integration testing can also cover interactions with, and interfaces provided by, external organizations (e.g., web services).' Meanwhile '(...) automated component regression tests play a key role in building confidence that changes have not broken existing components.' See International Software Testing Qualifications Board®, 'Certified Tester Foundation Level Syllabus.' (International Software Testing Qualifications Board, 11

impact assessment but, as mentioned above, they are not carried out according to a specific methodology. At this point, it should be stressed that integration and regression tests could initially cover only a small group of provisions created as RaC in a given branch of law. Only as the adoption of RaC becomes widespread would the integration and regression tests begin to cover an increasing number of provisions. Thus, the real benefit of conducting the tests would be postponed in time. Nor should integration tests be expected to reveal inconsistencies with legislation that did not originate as RaC. However, even a small improvement in the quality of legislation will be a significant benefit.

Along with the framework for the RaC quality assurance procedure, sets of test cases should be created like input data (factual states) which, when subsumed into the proposed regulations, would produce the expected results (output data). The test cases would gradually be extended with real-life situations. Over time, a comprehensive set of tests would emerge, which (if automated) could be carried out on a scale and at a speed unattainable by humans.⁴⁹ The scope of time and subject matter of the tests could be freely chosen, depending on established priorities. On the other hand, the task of curating the collection of test cases would demand a dedicated team of people to conduct reviews and updates periodically. Undoubtedly, some selection of test cases would also be necessary, as the need to maintain too rich a collection would generate significant costs.⁵⁰ Another disadvantage would be to limit the tests only to cases that are unambiguous or do not require complex interpretation - due to the previously described need for human involvement in the application of functional directives of the perceptual phase of interpretation.

The release of an official version of the RaC with a collection of test cases would allow certifying the software developed by private entities as compliant with the law in force. Such certification could be performed in

November 2019) 31-32 <<https://www.istqb.org/downloads/send/2-foundation-level-documents/281-istqb-ctfl-syllabus-2018-v3-1.html>> accessed 18 April 2021.

49 Accident Compensation Better Rules Discovery Team (n 2) 21.

50 Costs will result primarily from the need to periodically review and modify test cases. 'Continuously repeating the same tests leads to a situation where they stop detecting new defects at some point. To be able to detect new defects, it may be necessary to modify existing tests and test data, as well as to write new tests. Unmodified tests lose their ability to detect defects over time, just as pesticides are incapable of eliminating pests after a period of time. In some cases - such as automated regression testing - the pesticide paradox can be beneficial because it allows you to confirm that the number of defects associated with regression is small. International Software Testing Qualifications Board® (n 378) 17.

a highly automated way (by passing specific sets of test cases), and the revenue from it would contribute to the state budget. Such a solution would be profitable also for software developers; appropriately calculated costs of certification would be more beneficial than the necessity to transcribe regulations into code by oneself and then test the created solutions.

Software testing is divided into different levels and types. Levels⁵¹ group the tests according to the complexity of the code and include tests:

- modular - checking the operation of individual components in isolation from the overall code; in the case of RaC, this would mean testing the operation of the lowest existing editorial unit,
- integration - already mentioned above, involves the interaction of a component (e.g. editorial unit) with other components, but may also concern the interaction of the whole system with other systems; in the case of RaC it could mean checking the integration of higher-level units (chapters, sections, etc.) or even entire normative acts with other provisions in the form of RaC,
- system - concerning the behaviour and capabilities of the system as a whole, in terms of functional and non-functional aspects (e.g. reliability); in the case of RaC this means testing interactions within or among individual normative acts,
- acceptance - involving a level close to the system level but carried out by the target user or system operator, e.g. the addressees of standards or law enforcement bodies.⁵²

The division of tests into types is based on separating groups of tests that check specific characteristics of the code.⁵³ RaC test types would not differ significantly from the quality assurance of standard code and would include functional tests (whether the code can perform the desired actions), non-functional tests (whether the code is efficient, safe, etc.), and so-called white-box tests checking to what extent the code is covered by tests. However, tests related to change would be of particular importance for RaC. In addition to the previously mentioned regression tests, this includes tests confirming the removal of a previously detected defect, e.g. a logical error (in the case of RaC this could be a loophole). In their most

51 *ibidem* 30.

52 It should be noted that at this stage tests are also carried out on the compliance of the product with the contract for its creation or the applicable legislation. cf *ibidem* 37.

53 *ibidem* 39-41.

basic version, they consist of re-executing the test that previously returned an error.⁵⁴

RaC quality assurance can be automated to some extent, just like for any computer code. Software test creation has long been seen in the IT industry as a tedious and low-value activity that distracts professionals from their main objectives.⁵⁵ Systems such as Jenkins are widely used to facilitate, among other things, test execution but particular attention should be paid to the use of artificial intelligence-based solutions - due to their ability to learn quickly and be more efficient than previous applications. Some of them conduct static code analysis in real-time, which enables to detect, identify, and correct errors in code while still at the stage of development (similar to how text editor checks the correctness of spelling and grammar).⁵⁶ Other applications automatically create unit (module) tests for the analysed code.⁵⁷ This could create an interesting situation when one code (algorithm) controls another code (RaC), which in some cases would define the rules for other algorithms.

RaC quality assurance is also associated with regulatory sandboxes.⁵⁸ Testing of RaC may take place in special test environments, i.e. isolated programs or groups of programs simulating real-world operations. In the next step, the designed regulations should undergo a test on a selected group of addressees, who will apply them in practice and provide feedback.

The idea of regulatory sandboxes is not new. Dedicated pilot programs have been established in many countries, mostly in the FinTech sector,⁵⁹

54 The importance of frequent and early testing of code found its particular expression in a separate software development methodology Test-driven-development, which was described by Kent Beck. Kent Beck, *Test Driven Development: By Example* (1 ed., Addison-Wesley Professional 2002).

55 Matthew Lodge, 'Software Testing Is Tedious. AI Can Help.' (Harvard Business Review Home, 22 February 2021) <<https://hbr.org/2021/02/software-testing-is-tedious-ai-can-help#>> accessed 18 April 2021.

56 An example of this is the DeepCode tool:<https://www.deepcode.ai/>.

57 A model example is the Diffblue Cover application: <https://www.diffblue.com/>.

58 'Regulatory sandboxes enable a direct testing environment for innovative products, services or business models, pursuant to a specific testing plan, which usually includes some degree of regulatory lenience combined with certain safeguards.' Radostina Parenti, 'Regulatory Sandboxes and Innovation Hubs for FinTech' (European Parliament Think Tank, 30 September 2020) 9 <[https://www.europarl.europa.eu/RegData/etudes/STUD/2020/652752/IPOL_STU\(2020\)652752_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/STUD/2020/652752/IPOL_STU(2020)652752_EN.pdf)> accessed 5 February 2021.

59 Examples of existing solutions for FinTech are mentioned by J. G. Jiménez and M. Hagan as well as R. Parenti. Jorge Gabriel Jiménez and Margaret Hagan, 'A

although similar programs have also been created for other sectors of the economy, e.g. transport or energy.⁶⁰ So far, the main purpose of sandboxes has not been to test new regulations, but to enable the implementation of new business ventures that are difficult to classify and to make it easier for supervisory authorities to understand how they operate.⁶¹ The conditions for running a venture within a sandbox are relaxed compared with reality. In return, the participating entities are obliged to cooperate closely with state authorities and comply with the established rules. New regulations are developed empirically, as a result of the experience of sandbox participants.

The model described above can, and should, be applied to RaC. Existing sandboxes could successfully serve to introduce and test regulations in the form of RaC. There is a chance to create a positive feedback loop: innovative ventures of sandbox participants would justify the creation of new regulations in the form of RaC, which during testing would inspire the creation of further innovations based also on the RaC code, which would cause further expansion and improvement of RaC. Separate regulatory sandboxes may also be established due to the specific needs of a particular area of law (e.g. tax law) or even a particular law (e.g. public procurement law). Thanks to them, interested entities would be able to test the proposed regulations and, at the same time, work on new business models or ways of providing certain services.⁶² There are thus two possible starting points: from the venture or the draft regulation. The effect will be similar: new regulations will be based on empirical data.⁶³

Recently a legal framework for creating AI regulatory sandboxes has been provided in Articles 53 to 55 of the proposal. This is particularly

regulatory sandbox for the industry of law.’ (Legal Executive Institute, 2019) 2 <<http://www.legalexecutiveinstitute.com/wp-content/uploads/2019/03/Regulatory-Sandbox-for-the-Industry-of-Law.pdf>> accessed 5 February 2021; *ibid* 9. It should be noted that the Polish Financial Supervision Authority has established a regulatory sandbox in 2018.

60 cf Deloitte Center for Government Insights, ‘Future of Regulation. Case studies.’ (Deloitte Center for Government Insights, 2018) <<https://www2.deloitte.com/content/dam/Deloitte/us/Documents/public-sector/us-fed-future-of-regulation.pdf>> accessed 5 February 2021.

61 cf Parenti (n 58) 9.

62 cf Jiménez and Hagan (n 59) 3.

63 Such an approach directly supports the idea of *evidence-informed policy making* supported by the OECD. cf OECD, ‘Building Capacity for Evidence-Informed Policy-Making: Lessons from Country Experiences’ (2020) OECD Public Governance Reviews <<https://doi.org/10.1787/86331250-en>> accessed 5 February 2021.

important to developing RaC as one of the objectives of creating regulatory sandboxes for high-risk AI systems is to conduct research on the effective way to regulate AI. The suggested approach is based on recital 71 of the proposal according to which ‘Artificial intelligence is a rapidly developing family of technologies that requires novel forms of regulatory oversight and a safe space for experimentation, while ensuring responsible innovation and integration of appropriate safeguards and risk mitigation measures.’ The statement above should be interpreted as referring not only to the meaning of the law but also to its form. Using text-written law as a sole form of regulating high-risk AI systems will have a negative impact on innovation for the reasons discussed above.

The detailed principles of regulatory sandboxes for RaC require an in-depth analysis, based on the experience for FinTech and AI sandboxes. A law tested in this way would be better adapted to the reality and expectations of the addressees and more stable compared with laws created only using consultation or impact assessment. Additionally, the way RaC is used in regulatory sandboxes can be analysed continuously, improving the identification of bottlenecks in the process, and accelerating the design of necessary improvements.⁶⁴

4. *The Beginning of the Road*

The topics discussed above are only a small part of the RaC issue. An issue that is complex, difficult but at the same time fascinating and closely related to the everyday problems of a vast number of people.

The computer code can support the interpretation of legal text despite the ambiguity of the latter. In turn, the need for interpretation will influence the development of RaC; the necessary elements of norm-shaped expressions will have to be represented in the computer code. However, it is unlikely that human involvement will be eliminated from the process, at least in the near future. Near-complete automation of law may be introduced to only some of the most repetitive, standardized services and only in the first instance of proceedings. In the remaining cases, the application of RaC ought to be supervised by a human, mainly because of technical and ethical issues concerning artificial intelligence.

64 Similar comments on the creation of regulatory smart contracts are raised by Dariusz Szostek. cf Szostek (n 17).

At the same time, the practice of developing code, in particular ensuring its quality, can offer proven solutions to enhance the drafting of normative acts, eliminate potential contradictions and prevent emerging gaps in the legal system. This can be conducted by humans or, to a certain extent, by artificial intelligence algorithms. Thus, errors detected in computer code will also be corrected in text written in natural language.

There is no doubt that the research on RaC must be continued and should also concern (apart from the aspects described above) the necessary changes in the principles of legislative technique, the creation of law focused on the end-user, the principles of promulgation of normative acts and finally the version management (e.g. using the Git version control system).⁶⁵ Regardless of which issue attracts more attention from researchers or businesses, the simultaneous existence of law in form of a text and code seems to be essential. Without RaC we accept the existence of a gap between human law and computer code, both of which define the rules for the functioning of our reality and are simply two sides of the same coin.

65 More information about Git is available on its official website: <http://git-scm.com>.

