A Legally Relevant Socio-Technical Language Development for Smart Contracts

Vimal Dwivedi
Department of Software Science
Tallinn University of Technology (of Aff.)
Tallinn, Estonia
vimal.dwivedi@ttu.ee

Supervisor: Prof. Alex Norta
Department of Software Science
Tallinn University of Technology
Tallinn, Estonia
alex.norta@ttu.ee

***Abstract*—Smart contracts play an advent role in automated business participation by rendering collaboration processes more time efficient, cost-effective and establishing more transparency. Smart contracts facilitate trust-less systems, without the need for intervention from third-party intermediaries. Existing smart-contract languages mainly focus on technical utility and do not take into consideration social and legally relevant issues, e.g., lack of semantics, ontological completeness, and so on. In this research, we address the gap by developing with rigorous means a smart contract's language that aims to be legally relevant, and that comprises socio-technical utility for cross-organizational business collaboration. The proposed language seeks to retain the strengths of the already existing languages of different generations while eluding their limitations. We aim to identify and implement abstract grammar patterns for a smart- contract language that has the expected application utility and verifiability. We evaluate the developed language based on automating industry-collaboration cases with our novel smart-contract language to test the suitability, utility and expressiveness.**

***Index Terms*—Ontological completeness, Suitability, Expressiveness, Socio-technical, ANTLR, Blockchain, Smart contracts**

I. Motivation

The blockchain is an incorruptible distributed ledger, a trust-less system that is duplicated across multiple networks [4]. Therefore, an application that could run previously through centralized medium only, now it can make possible without the trusted medium such as smart property, e-healthcare. With the emergence of blockchain technology, smart contracts have become popular because it can now operate in a decentralized fashion without the requirement of the trusted third-party. The smart contract is self-enforceable, self-executable, written in a program code, runs on a blockchain network [3]. Smart contract code executed itself when a set of pre-determined conditions met during the life-cycle of contract execution.

With respect to existing smart-contract languages, the most notable version such as Solidity and Rohlang are more recent adoption by industry. However, not only solidity but another existing smart-contract language does not have the social and legal semantics of business collaboration [7]. The social utility comprises the mechanism to maintain the transparency to all relevant stakeholders of business collaboration if conflicts arise [1]. The legal utility comprises the semantics of efficient breach of smart contracts as per law and economic [6].

Recently an experiment of the smart contract has been accomplished with the crowd-funding project named decentralized autonomous organization (DAO). It was hacked because of security flaws in smart-contract language, resulting in losses of money. This incident shows it is not sufficient to develop a contract with Turing-complete language such as solidity. Instead, it is essential to study suitability and expressiveness of the language [9]. Suitability means that smart-contract languages comprise the concept and properties to allow formulation of real-world contracts in the legally relevant way. Expressiveness implies the libraries of smart-contract languages have mathematical clarity that ensures the uniform enactment by several business process engines.

In [12], the proposed methodology uses solidity programming language to demonstrate the feasibility of untrusted business-process monitoring and execution in smart contract. However, the solidity does not take into account the ontological suitability and expressiveness because of Turing-complete nature. The core ontological concepts answer the conceptual questions of Who, Where and What about contracting parties. Without such an suitability and expressiveness, if a contract is valid and free from security issues, then verification of parties is not possible before the enactment. Therefore, the goal of this work is to develop a smart-contract programming language that incorporates socio-technical suitability and expressiveness for guiding business collaboration in a legally relevant way.

II. Objectives

This paper fills the gap in the current state of the art by posing the main research question how to develop a smart-contract language that has concept and properties for guiding business collaboration in a legally relevant way. From there we deduce several sub-questions. How to establish legal relevance in smart contracts that have socio-technical utility in smart contracts? How to design a language that combines the strengths of established languages of various generations while avoiding the weaknesses? How to identify and implement abstract grammar patterns for a smart-contract language that has the expected application utility and verifiability? From the first sub-question, we achieve the concept and properties

978-1-5386-5175-9/18/$31.00 ©2018 IEEE
DOI 10.1109/FAS-W.2018.00016

of business collaboration in legal relevance by developing an ontological structure of real-world contracts.

In order to aid the automation of contracting in business collaborations, it is required that working smart contracts could be produced by using the choreography language eSML [10] as a foundation. However, at the moment of writing, no solutions exist for mapping high-level choreography languages to smart-contract languages, such as the smart contracting lingua franca Solidity, while maintaining legal recognisability. Therefore, we need to deduce first sub-questions into several sub-questions. What are the requirement sets for an option to intent-fully not execute a contractual obligation, for the formation of contracts, and for party identification in smart contracts? What is the ontological structure of an option from legally enforceable requirements sets? What is the dynamic processing of ontological attributes?

From the first sub-question, we generate ontological structure and dynamic processing of business process agreements that work as input for designing smart-contract language. Therefore, we focus on second sub-question how to design a language that combines the strength of establishing various generation language while eluding their limitations. From the third sub-question, we achieve mathmatical clarity of the smart-contract language. Therefore, we focus on the semantics of context-free grammar to ensures uniform enactment by different process engines [8].

III. Methodology

The choice for research methodology for this thesis is the approach of the design science research methodology [2]. The authors of [2] described, ”the design-science paradigm seeks to extend the boundaries of human and organisational capabilities by creating new and innovative artifacts.” Hevner et al. [2] propose research principles, to conduct design science research (DSR), to create new and innovative artifacts, and for understanding, executing and evaluating information systems (IS) research. We focus on the following design-science research principles to generate results.

- **Design as an Artifact:** Our contribution is to produce artifacts in the form of constructs as a smart-contract language.
- **Problem Relevance:** The objective of this research is to develop smart-contract language that has socio-technical suitability and expressiveness for business collaboration.
- **Design Evaluation:** The utility, quality, and efficacy of a smart-contract language rigorously demonstrated via ANTLR tool.
- **Research Contributions:** Our first contribution is to develop an ontological structure of socio-technical and legal utilities of the business collaboration. Then follows the designing of language that combines of strengths of established languages of various generations while avoiding the weakness. Finally, we focus on the implementation of abstract grammar pattern for smart-contract language.
- **Research Rigor:** Existing tools from Truffle, will be used to emulate calls to a blockchain where these smart contracts are deployed, and write automated tests to ensure they work as expected.
- **Design as a Search Process:** Comparison of the results obtained from the artifact as a smart-contract language with existing languages, to reach desired goals.
- **Communication of Research:** We will present our contribution effectively both to technology-oriented as well as management-oriented audiences.

A potential solution to the identified research goal is to use the eSML schema previously defined by Norta [10], to develop a context-free grammar. Therefore, We focus on existing tools such as ANTLR to help reach closer to the solution of our research goal. ANTLR has a consistent syntax for specifying lexers, parsers, and tree parsers [11]. We will evaluate the language based on automating industry-collaboration cases with our novel smart contract language to test suitability, utility and expressiveness.

IV. Research Plan

We plan the time-line of our research activities as per the design-science guidelines in the table 1. Till the moment, we have accomplished the activities of problem relevance, research questions and finalized the methodology.

Table 1: Time line for Ph.D. Research

Activity	2018 Jan-Dec	2019 Jan-Dec	2020 Jan-Dec	2021 Jan-Dec
Problem Relevance: Identifying, analyzing and categorizing the challenges of smart contracts languages	✓			
Research Contribution 1: Design the ontological structure of suitable semantics of business collaboration	✓	✓		
Research Contribution 2: Identify the suitable libraries for designing ontological based smart contract language		✓	✓	
Research Contribution 3: Identify and implement abstract grammar pattern for a smart contract language			✓	
Design as an artifact /Design evaluation / Design as a search process: Composition of the thesis and preliminary defense			✓	✓
Communication of Research: The defense of the thesis				✓

References

[1] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. A survey of attacks on ethereum smart contracts (sok). In *International Conference on Principles of Security and Trust*, pages 164–186. Springer, 2017.

[2] Martin Bichler. Design science in information systems research. *Wirtschaftsinformatik*, 48(2):133–135, 2006.

[3] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 2014.

[4] Konstantinos Christidis and Michael Devetsikiotis. Blockchains and smart contracts for the internet of things. *IEEE Access*, 4:2292–2303, 2016.

[5] ConsenSys. *trufflesuite/truffle*, 2015 (accessed March 3, 2018). https://github.com/trufflesuite/truffle.

[6] Patrick Dahm. *The Efficient Breach of Smart Contracts*. http://learn.asialawnetwork.com/2018/02/22/efficient-breach-smart-contracts/.

[7] Mark Giancaspro. Is a smart contractreally a smart idea? insights from a legal perspective. *Computer Law & Security Review*, 33(6):825–835, 2017.

[8] Donald E Knuth. Semantics of context-free languages. *Mathematical systems theory*, 2(2):127–145, 1968.

[9] Alex Norta, Lixin Ma, Yucong Duan, Addi Rull, Merit Kõlvart, and Kuldar Taveter. econtractual choreography-language properties towards cross-organizational business collaboration. *Journal of Internet Services and Applications*, 6(1):8, 2015.

[10] Alexander Horst Norta. Exploring dynamic inter-organizational business process collaboration. *Dissertation Abstracts International*, 68(04), 2007.

[11] Terence Parr. *The definitive ANTLR 4 reference*. Pragmatic Bookshelf, 2013.

[12] Ingo Weber, Xiwei Xu, Régis Riveret, Guido Governatori, Alexander Ponomarev, and Jan Mendling. Untrusted business process monitoring and execution using blockchain. In *International Conference on Business Process Management*, pages 329–347. Springer, 2016.