Proceedings of the 23rd SMART-iSTEAMS Conference
in Collaboration with
The American University of Nigeria, Yola
& The IEEE ICN/IEEE Compter Society Nigeria Section
www.isteams.net/yola2020

Soft Computing Modelling for Blockchain Smart Contracts

[1]Salahudeen, R & [2]Fonkam, M.
School of IT & Computing
American University of Nigeria.
Yola, Adamawa State, Nigeria
E-mails: [1]ridwan.salahudeen@aun.edu.ng, [2]mfonkam@aun.edu.ng

ABSTRACT

Blockchain technology is radically changing how people carry out transaction in a more decentralized and trusted manner, by avoiding intermediaries and transparently distributing the entire ledger among all participant. The complexity of encoding the terms of agreement between transacting entities and the uncertainty in term of operation when third parties are employed to mediate between transacting parties are two challenges that one need to be traded-off for the other. In this paper, we propose a model thinking approach to leverage on the individual strength of two different programming paradigms to address this trade-off.

Keyword: Blockchain, Smart Contract, Modelling, Uncertainty, Complexity.

23rd iSTEAMS Conference Proceedings Reference Format
Salahudeen, R & Fonkam, M. (2020): Soft Computing Modelling for Blockchain Smart Contracts.
Proceedings of the 23rd iSTEAMS Conference, American University of Nigeria, Yola. April, 20. Pp 1-6 www.isteams.net/yola2020.
DOI - https://doi.org/ 10.22624/AIMS/iSTEAMS-2020/V23N1P1

1. INTRODUCTION

Transparent, trusted and verifiable digital transactions can be safely executed between unfamiliar participants today without the need of a trusted third-party institution (such as banks or governments) using blockchain technology. Blockchain is a radical technology that employs an immutable distributed ledger of transactions that maintains all records of transactions that have ever been stored on the network (Alharby & Moorsel, 2017). Transactional records are stored on the blockchain network as blocks, after verification for correctness and authenticity by special nodes in the network known as miners using a consensus protocol. Once a block is added to a blockchain network, the block is cascaded to the distributed ledgers of the entire participating nodes making it very difficult to alter blocks already added to the network. This not only ensures the integrity of transactions but also solves a critical problem for digital payments namely that of double-spending and fraud (Schwartz, Youngs, & Britto, 2014). Each block, which may contain multiple records of transactions, is identified by a cryptographically hashed key, and refer to the block immediately before it in the network, hence forming a chain of blocks from the very first block (the genesis block) in the chain to the one just added.

The innovation of blockchain technology has enabled the rise of a family of digital currency systems known as cryptocurrencies, which are gaining increasing traction globally. Most dominant amongst these cryptocurrencies are Bitcoin (that started the whole cryptocurrency revolution) and Ethereum (which introduced a generalized cryptocurrency model backed by programming language for smart contracts of all types and not just financial ones).

Bitcoin was introduced as a peer-to-peer (Nakamoto, 2008), purely electronic payment system based on cryptographic techniques that enable secure transactions between any two individuals without the involvement of a trusted intermediary such as a bank. This direct two-party digital payment system removed many overheads, resulting in reduced cost and faster processing with the removal of a third party or trusted middle man (Xiong, Zhang, Niyato, Wang, & Han, 2018). Some special nodes in the Bitcoin network known as miners have to solve a puzzle known as Proof of Work (PoW) to verify the correctness and authenticity of computations in any transaction to be added to the network and transformed into a block. The first miner to come up with the PoW will propagate its newly generated block into the network after which other nodes will validate and then build upon the newly added block. However, even with the great potentials of Bitcoin platform for blockchain technology, it is highly limited in creating complex distributed application (e.g., smart contract) on top of it due to its limited programming capabilities for complex transactions. This is where the Ethereum comes in. Ethereum is highly adaptable and hence more suitable for complex distributed applications (Wood, 2014). A major key feature of Ethereum is its support for smart contracts.

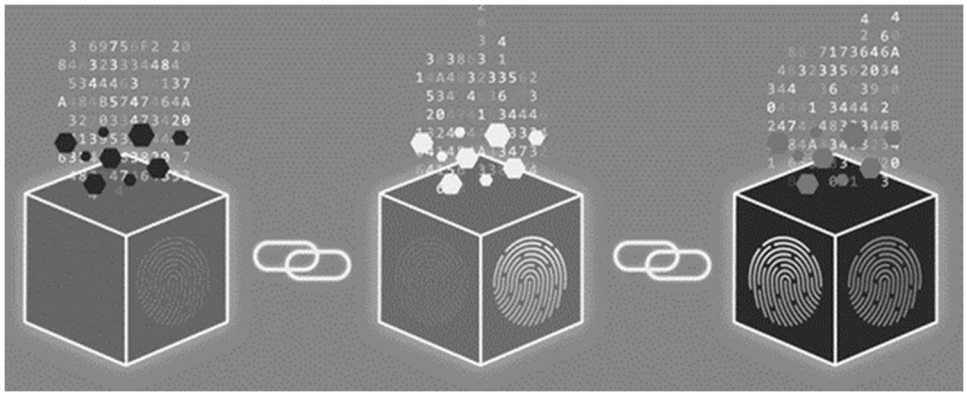

Figure 1: Depiction of a Blockchain

Smart contracts are computer executable codes for mutual agreements between the participating parties in a contractual relationship and run on a blockchain (particularly Ethereum) to coordinate the terms of an agreement (Cox, 2017). At the core, they facilitate three functions: stores rules, verify rules and self-execute rules. A smart contract is an idea proposed by Szabo in 1994 to enhance the efficiency in the enforcement of terms agreed upon by participating parties in an automated manner without the need of a trusted third party to enforce or execute the terms of agreement. However, the terms of agreement used in creating a smart contract are human natural language that are prone to vagueness and imprecision. Hence, the idea conceived in this research is to model how uncertainties could be handled in these smart contracts due to the inherent vagueness and imprecision in the natural languages used to express the terms of agreement.

The paper is organised in sections, from sections 1 – 5. While section 1 already introducing the background of the study. Section 2 will give a basic overview of blockchain technology and smart contract. Section 3 will discuss the problem statement of this research paper. Section 4 will explain the proposed model for addressing the identified problem. Finally, section 5 will conclude the paper.

2. OVERVIEW OF BLOCKCHAIN AND SMART CONTRACT

Blockchain is a decentralised database of transactions that is replicated among all participating nodes of a network. Each participant maintains the same copy of the database and when some special nodes of the network called the miners make an update, the update is propagated into the decentralised copies of each node of the network in a peer-to-peer manner. The miners are also normal participating nodes of the network but have additional resources like huge processing power to verify the recent transactions that needed to be added to a blockchain network and propagate their Proof of Work (PoW) to other nodes. Because there is incentive attached to the process of mining a block to be added to a blockchain network, miners are often in a race to complete the PoW puzzle. The PoW propagated by the first miner along with the block created by the miner will be propagated to all other nodes in the network in a peer-to-peer fashion. The PoW must be consensually agreed upon by other miners then another race for the next block to be added will begin. Hence, the records of transactions are ordered to form blocks, then blocks are back-linked together to form chains of blocks, giving the genesis of the name Blockchain. Figure 1 illustrates how series of transactions are synchronized to form blocks and several blocks are chained together to form chain of blocks.

The ultimate goal of blockchain technology is to ensure transparency, trust and secured transaction between unfamiliar parties without the involvement of trusted middle authorities such as lawyers, agents, banks or any form of centralized management. As such, blockchain is universally agreed to be a tool for radically reshaping the society and the economy to a more decentralised world. Smart contracts are terms of agreement between transacting parties that are automatically executed by computer codes giving rise to cheaper and transparent transactions. The smartness of this contract is due to its automatic and transparent way of facilitating and enforcing terms in a contract once the specified conditions in the contact are met [7]. The idea of smart contract was first conceived by Szabo in 1997 but was not adopted until the advent of blockchain technology. With smart contract, cheaper and more transparent transaction can be achieved.

3. PROBLEM STATEMENT: THE UNCERTAINTY VS COMPLEXITY TRADE-OF

One of the major achievement of blockchain is the elimination of third-party institutions whom trust is relied upon when two or more unfamiliar parties are transacting. In the traditional based contract systems where a third-party mediates between the participating entities, the operational details of this middle party are mostly abstracted from their client. This abstraction creates a huge room for uncertainties on how fair deals between the participating clients are being reached by this middle party. On the other hand, blockchain's elimination of third-party institutions has added additional complexity to the technology since the network is now responsible for handling the middle-party operation.

This complexity in implementing the logic to mediate fairly between the transacting entities is a trade-off for the uncertainty involved when dealing with third-party institutions. To give a simple illustration of this trade-off, let us assume you are a software company that deals with online services such as e-commerce or airline ticket booking and your customers make payment through payment gateway processors such as Paypal or Remita. Your customer recently made complaints about the enormous flat rate charges the gateway processor deducts regardless of the might of their transaction. Based on this, it is either you are ready to shoulder the complexities of these payment processing with its other complicated details -such as the credit risks when dealing with reminding customers of due date for their payments.

In this way, you are more certain about all the abstractions in the payment and billing process. That is, trading off uncertainty for complexity. Alternatively, you could allow the intermediary payment gateway to continue to deal with those payment complexities while you are abstracted from the complicated details, here, you are trading off complexity for uncertainty. This analogy is the exact implications of blockchain technology especially with the smart contract. All the nitty gritty in the terms of agreement among transacting parties needs to be coded logically, holistically and comprehensively without the involvement of third-party mediation. Hence, a comprehensive model for handling these uncertainties that may arise in a smart contract encoding is what this research paper seeks to address.

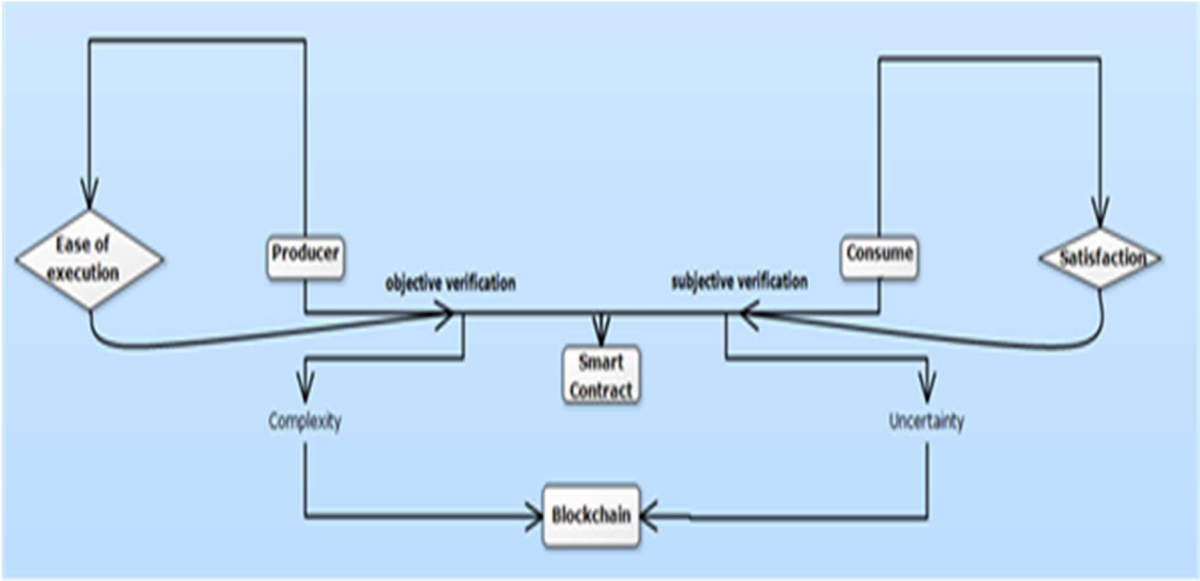

Figure 2: Smart Contract Modelling

4. PROPOSED MODEL FOR SMART CONTRACT UNCERTAINTIES

The proposed model is named Smart Contract Modelling and illustrated in Figure 2. A producer rectangle in the diagram represent the goods and/or services rendered by one of the transacting entities while the other rectangle labelled customer represents the other transacting partner who consumes the goods and/or services provided by the first entity. The other rectangles in the diagram are smart contracts depicting the legal terms of agreement between the two transacting entities and blockchain depicting the technology of automating transactions without the involvement of third-party entity. While the producer can verify the consumer's compliance to the terms of agreement objectively, these could include terms such as payment value, renewal periods, penalty etc., the consumer's verification of producer's compliance are rather subjective, these may include issues like, delivery satisfaction, standard of service/product. Hence, the verification of the producer is objective verification while that of the customer is subjective verification.

The ease of execution variable emanating from the producer measures the simplicity of encoding the dynamic terms of agreement and it equally serves as a reinforcement loop to the objective verification. The more tedious the objective verification gets the more complex the smart contract becomes. Similarly, the satisfaction variable from the consumer measures level of consumer's satisfaction with the transaction, it also reinforces the subjective verification, which in turn reinforces the uncertainty level of the smart contract. Finally, after a careful observation of the proposed model, we could address this trade-off still on the blockchain technology by clearly delineating the appropriate programming specification model that is best suited to either of the trade-offs.

When the complexity outweighs the uncertainty then a declarative programming spec should be used, because in declarative languages, contractual terms and their enforcements can be made declaratively without specifying the details of how each party must accomplish their terms of agreement. Correspondingly, when uncertainty outweighs complexity, then a procedural programming paradigm will be more suitable. In procedural languages, systematic details of how to perform a task is encoded, hence the ambiguity or doubts of any of the transacting parties would be detailed out explicitly for a smart contract to execute automatically. An algorithm to interpret the model is given below.

Algorithm: SmartContractUncertainty()
Input:
    $X_p \leftarrow$ Producer term
    $X_c \leftarrow$ Consumer term
    $\Delta_c \leftarrow$ Complexity threshold
    $\Delta_u \leftarrow$ Uncertainty threshold
Output:
    $P_b \leftarrow$ Blockchain programming paradigm

foreach term (t_i) in X_p
    if $t_i > \Delta_c$ then
        P_b should implement declarative paradigm
    else
        P_b should implement procedural paradigm
end foreach

foreach term (t_j) in X_c
    if $t_i > \Delta_u$ then
        P_b should implement procedural paradigm
    else
        P_b should implement declarative paradigm
end foreach

5. CONCLUSION

Blockchain is an innovative technology providing the possibility of trustful and transparent transaction between strangers without the involvement of third-party institutions. However, there was a concern about the technology's smart contract on its efficient way of addressing the complexity and uncertainty trade-off (Liu, et al., 2019). This research work was able to show the feasibility of addressing the trade-off through a model thinking approach and came up with a dynamic model of this problem. Future works can be carried on this proposed model to implement the developed algorithm into a real-time automated system.

REFERENCES

1. Alharby, M., & Moorsel, A. (2017). Blockchain-based Smart Contracts: A Systematic Mapping Study. *Computer Science & Information Technology*, 125– 140.
2. Cox, T. (2017). Eos. io technical white paper. *GitHub repository*.
3. Liu, Z., Luong, N. C., Wang, W., Niyato, D., Wang, P., Liang, Y., & Kim, D. I. (2019). A Survey on Applications of Game Theory in Blockchain. *IEEE*.
4. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Selfpublished paper*. Retrieved from https://bitcoin.org/bitcoin.pdf
5. Schwartz, D., Youngs, N., & Britto, A. (2014). The ripple protocol consensus algorithm. *Ripple Labs Inc White Paper, 5*.
6. Szabo, N. (1997, September 1). *Formalizing and securing relationships on public networks.* Retrieved March 28, 2020, from https://firstmonday.org/ojs/index.php/fm/article/view/548/469
7. Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper, 151*, 1-32.
8. Xiong, Z., Zhang, Y., Niyato, D., Wang, P., & Han, Z. (2018). When mobile blockchain meets edge computing. *IEEE Communications Magazine, 56*(8), 33–39.