Formal Specification Technique in Smart Contract Verification

Seung-Min Lee
Dept. of Computer Engineering Graduate School
Dankook University
Yongin-si, Republic of Korea
Seungmin-Lee@dankook.ac.kr

Soojin Park
Sogang Institute Advanced Technology
Sogang University
Seoul, Republic of Korea
psjdream@sogang.ac.kr

Young B. Park*
Dept. Software Engineering
Dankook University
Yongin-si, Republic of Korea
ybpark@dankook.ac.kr

***Abstract*—The block chain technology is changing rapidly. The block chain guarantees the integrity of the book through a specific consensus of the participants. In the past, the block chain technology had a limited range of applications. However, the use of block chain technology is gradually expanding as smart contracts that can formulate general business logic are mentioned. Already studied the components of smart contracts in other studies and proposed the possibility of extending them on the basis of ontology. And research on securing traceability of smart contract based on ontology has been carried out. However, research on various transactions constituting smart contracts is lacking. In this paper, the constituent elements of smart contract are analyzed and expressed by ontology. And the process of negotiating the components is represented by each transaction. Finally, we construct the component represented by the ontology as XML by including the state information in the transaction. In this way, the smart contract is represented in a formal language that contains state information. It also laid the foundation for a smart contract that can be reused and verified.**

Keywords—Blockchain; Smart-Contract; Ontology; XML; Formal Language

I. INTRODUCTION

The block chain connects the data generated from the distributed nodes in a block unit. And all nodes share a chain of blocks generated through a specific agreement algorithm.

This is a technique to guarantee the integrity of the distributed database [1]. In the past, the scope of application of the block chain technology was limited, but the scope of application is gradually expanding as the smart contract is mentioned [2]. Smart contracts are to contract and implement various types of contracts by establishing rules [3]. Within a block-chain platform, smart contracts are written in programming language code, and smart contract configurations differ depending on the platform type [4, 5].

Various studies are underway to expand the scope and use of smart contract technology. D. Clack has studied the necessary requirements and design options for smart contract forms and has proposed future research directions [6, 7].

Alex Norta presented a lifecycle for smart contract execution. And, the proposed life cycle was verified by model-checking [8]. Patrick Dai analyzed the limits of smart contracts used in Etherium. We also proposed the advantages of smart contract used in Quantum and plans for future development [5].

In addition, ebXML-based data exchange has been conducted to exchange data between block-chain networks [9]. A self-adaptive system for dynamically changing the performance condition of smart contracts has also been studied [10].

Previous studies have focused on the process of implementing smart contracts and the advantages and disadvantages of smart contracts. These studies are the study of the overall flow and the pros and cons of smart contracts.

However, there is a lack of research to validate transactions that occur in smart contracts. To validate a smart contract, the components that make up the smart contract must be defined. Transactions occur in the process of mutually negotiating defined components. In other words, smart contract is the result of several transactions [Fig. 1].

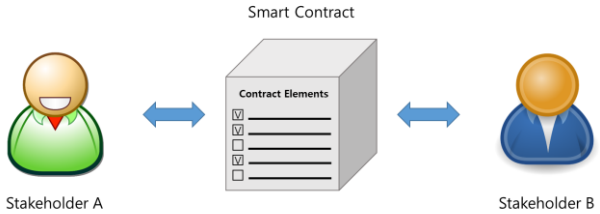

Fig. 1. Smart contract concept

In this paper, we analyze the smart contract components from the previous study for smart contract verification. Then, the analyzed contents are represented by ontology and converted into XML. In the process of transforming into XML, state information is added for analysis of transactions and composed of XML. This provides a basis for constructing smart contracts as state models.

The composition of this paper is as follows. Chapter 2 explains the essential requirements of block chains and smart contracts, ontologies, and ebXML-based data exchange studies. In Chapter 3, we analyze essential requirements of smart contract and express it based on ontology. And converts the content of the smart contract created in the ontology to XML.

978-1-7281-1288-6/19/$31.00 ©2019 IEEE

Section 4 analyzes the results of the conversion and discusses future research.

II. Related Research

A. Blockchain

Block chain is a kind of distributed computing that manages shared data according to agreed rules of participating nodes to ensure integrity. Information about all transactions written in the block chain can be shared by all nodes to maintain the integrity of the data.

Since the data is managed according to the agreement of the nodes, it is possible to secure the trust of the data without managing the data centrally [11].

Based on this, smart contracts are attracting attention. Smart contracts fulfill general contracts such as payment terms through computer transactions that enforce contract terms and minimize the need for a reliable intermediary [3].

B. Essential Requirements for Smart Contracts

In D. Clack's study, we defined five essential requirements to implement smart contracts and mentioned four discussion items to fulfill this requirement.

Essential requirements include how to create and edit smart contracts, format support for storing, retrieving and transferring, protocols for smart contracts, traceability to smart contracts, and finally, non-lawful transaction methods.

And the four discussion items for these essential requirements are supported for contracts, including parameters and related metadata, standardized transmission format agreed, Ontology for supporting analysis and reasoning of smart contracts, and lastly, And to discuss the code [5].

C. Ontology

Ontology is a model that expresses knowledge and information constructed in a way that people can understand through mutual consensus in a conceptual and computer-friendly form.

By explicitly defining the type of the concept or the constraints of its use, it is possible to interact with the human being and the computer.

The ontology consists of triple sentences, and the triple sentence forms consist of purpose, relation, and object. Among them, relations can be composed of various concepts [10].

D. ebXML-based data exchange

[9] proposed a method for exchanging data between different block-chain networks.

[9] propose a process to collect the elements used in different block chain networks through ebXML and configure them to be able to communicate. However, there has been a problem that it is difficult to check the state change of data according to the process [Fig. 2].

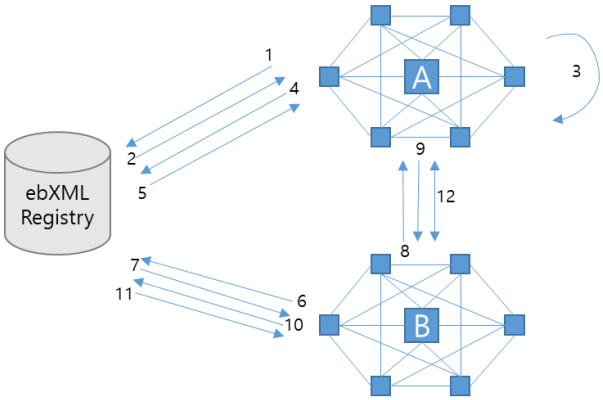

Fig. 2. ebXML-based data exchange process

III. Linking Formal Languages and Smart Contract Elements

A. Smart contract elements

The main content of the mandatory requirement is to check that the format of the document is a standard format that complies with the law when conducting the Smart Contract. The main content of the document is how the writing and editing process will proceed and how the smart contract process should proceed. In this paper, we analyze the essential requirements mentioned above and derive the main points of view [Table. 1].

Table. 1. Essential Requirements and Content Analysis

Essential Requirements	Focus
How to create and edit smart contracts	Create, Edit
Standard format for storing, retrieving, and transferring smart contracts	Standard format
Protocols for legitimately running smart contracts	Smart Contract process
How to combine legal contract with smart contract	Legal Contract
Smart contract forms that can accept laws and regulations	Legal based standard format

In this paper, we analyze the essential requirements mentioned above and derive the main points of view. In addition, a further discussion of this requirement in the D. Clack paper is given in [Table. 2].

Table. 2. Additional discussion and content analysis

Merit Further Discussion	Focus
Editing (What-You-Is-Is-What-You-Get, What-You-Is-What-You-Mean)	Smart Contract Format (Subject, Process, Object)
Transmission (For communicating data between different applications)	Standard Format
Ontology (Support for Standard Format)	Semantic Reasoning, Development of Querying Application
Binding smart legal agreement and code	Smart Contract Process, Legal Specification

The main content of further discussions refers to the forms, standards, processes, and extensions of smart contracts.

In this paper, we analyzed the issues that need to be discussed further, and found out the main points of view and summarized them as a focus.

B. Formal language, connection, conversion

The D. Clack's paper describes the use of ontologies to introduce smart contract extensions and query functionality. Based on this, we intend to express the contents of the focus on the elements that constitute the smart contract by ontology.

In summary, the parties agree to proceed with the Smart Contract. The agreements are related to the smart contract configuration and progress.

The Smart Contract configuration consists of a legal document format, a transport protocol, and a programming language for smart contracts based on it. The ontology configuration is as follows [Fig. 3].

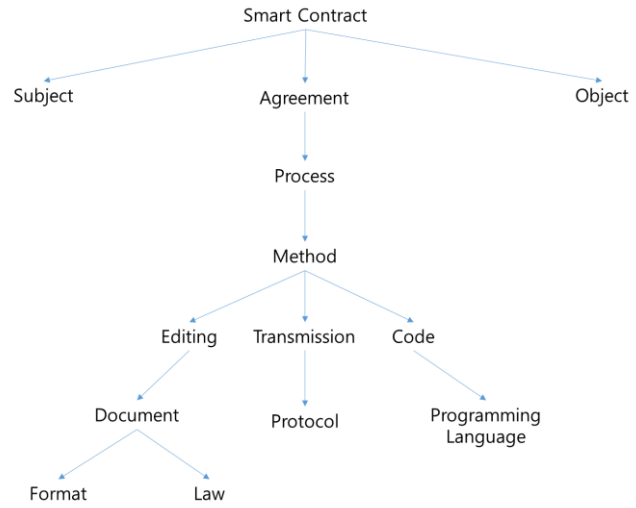

Fig. 3. The construction of a smart contract represented by an ontology

In order to use a smart contract consisting of an ontology on a computer, there must be information that several agreement processes have been completed.

In order to record the agreement information in the process of converting to the formal language, the state information is added and converted into the formal language.

The ontology is expressed using XML which is a commonly used markup language among various formal languages. The result is as follows [Fig. 4].

By constructing XML information, including state information, the smart contract process can be configured to follow state-based flows.

In other words, since the smart contract process can be represented as a state model, it can be expressed by State Diagram, which can be used as a basis for verification through various system model verification tools.

```
<Smart Contract>
    <StatusInfo>
        <Subject = "StakeHolder1">
            <State>
            </State>
            <Date>
            </Date>
        </Subject>
        <Object = "StakeHolder2">
            <State>
            </State>
            <Date>
            </Date>
        </Object>
    </StatusInfo>
    <Agreement>
        <Process>
            <Method>
                <Editing>
                    <Document>
                        <Format>
                            <Content>
                                <Subject>
                                </Subject>
                                <Object>
                                </Object>
                                <SourceItem>
                                </SourceItem>
                                <TargetItem>
                                </TargetItem>
                                <Deadline>
                                </Deadline>
                            </Content>
                        </Format>
                    </Document>
                </Editing>
            </Medhot>
        </Process>
    </Agreement>
</Smart Contract>
```

Fig. 4. Smart contract structure converted to XML format (Docment-format element)

IV. Conclusions and Future Works

In this paper, we analyze smart contract components and express them in ontology to verify smart contracts. It is confirmed that various conclusions are needed in constructing a smart contract through ontology expression.

And it is expressed in XML including state information in order to make it recognizable on the computer. This provides a foundation for creating State Diagrams and provides a foundation for linking with various tools for analyzing State Diagrams. In addition, by expressing it as an ontology, we were able to visually confirm the configuration of the smart contract and laid the ground for reuse.

However, there are various types of block-chain networks, and the configuration of smart contracts used for each block-chain network differs. This may lead to data exchange between different block-chain networks. Future studies will verify the smart contract composed of XML through the state model validation tool. We also plan to study how to exchange data between different block-chain networks

Acknowledgment

This research was supported by The Leading Human Resource Training Program of Regional Neo industry through the National Research Foundation of Korea(NRF) funded by the **Ministry of Science and ICT**(No.NRF-2016H1D5A1909989)

This research was supported by the MSIT(Ministry of Science, ICT), Korea, under the ITRC(Information Technology

Research Center) support program (IITP-2018-2017-0-01628) supervised by the IITP(Institute for Information & communications Technology Promotion)

REFERENCES

[1] Nofer, Michael, et al. "Blockchain. Business & Information Systems Engineering 59, 3, 183–187." DOI: http://dx. doi. org/10.1007/s12599-017-0467-3 (2017).

[2] Buterin, Vitalik. "A next-generation smart contract and decentralized application platform." white paper (2014).

[3] Szabo, Nick. "Smart contracts." Unpublished manuscript manuscript (1994).

[4] Jung Hyun An, Young B. Park. “Context recognition of self-0adaptive system for dynamic smart contract confition conflict resolution”. Korea Society of Block Conference. May 2018

[5] Clack, Christopher D., Vikram A. Bakshi, and Lee Braine. "Smart Contract Templates: essential requirements and design options." arXiv preprint arXiv:1612.04496 (2016)

[6] Clack, Christopher D., Vikram A. Bakshi, and Lee Braine. "Smart contract templates: foundations, design landscape and research directions." arXiv preprint arXiv:1608.00771 (2016).

[7] Norta, Alex. "Creation of smart-contracting collaborations for decentralized autonomous organizations." International Conference on Business Informatics Research. Springer, Cham, 2015.

[8] Dai, Patrick, et al. "Smart-contract value-transfer protocols on a distributed mobile application platform." URL: https://qtum. org/uploads/files/cf6d69348ca50dd985b60425ccf282f3. pdf (2017).

[9] Jung Hyun An, Young B. Park. “mutual consultation method for data exchange between blockchain networks”. Korea Society of Block Conference. May 2018

[10] An, JungHyen, and Young B. Park. "Methodology for Automatic Ontology Generation Using Database Schema Information." Mobile Information Systems 2018 (2018).

[11] Yli-Huumo, Jesse, et al. "Where is current research on blockchain technology?—a systematic review." PloS one 11.10 (2016): e0163477.