Faculty of Computer Science and Information Technology

An Extended Methodology Based on Agent-Oriented Modelling Approach for Blockchain Application Development

Michelle Ten LiBin

Master of Science
2023

An Extended Methodology Based on Agent-Oriented Modelling Approach for Blockchain Application Development

Michelle Ten LiBin

A thesis submitted

In fulfillment of the requirements for the degree of Master of Science

(Software Engineering)

Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK

2023

DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Malaysia Sarawak. Except where due acknowledgements have been made, the work is that of the author alone. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

……………………………

Signature

Name: Michelle Ten LiBin

Matric No.: 19020109

Faculty of Computer Science and Information Technology

Universiti Malaysia Sarawak

Date:1/5/2023

ACKNOWLEDGEMENT

Undertaking this master's research has been a truly life-changing experience for me, and it would not have been possible without the support and guidance that I received from many people.

First and foremost, I would like to express my special thanks to my supervisor Dr Cheah Wai Shiang.

I gratefully acknowledge the funding from the Zamalah Siswazah UNIMAS Scholarship to undertake my Master. Thanks to Associate Professor Dr Chiew Kang Leng to give me this opportunity to receive the scholarship.

I want to express my gratitude to the Centre for Graduate Studies and the administration of Universiti Malaysia Sarawak for guidance and support.

Lastly, my family deserves endless gratitude: my father, Ten Eng Le; my mother Cecilia Lee Phaik See; and my siblings, Melanie Ten Li Yen and Melisa Ten Li Lian, who have been supporting me along my master's journey.

ABSTRACT

Blockchain application development has received much attention nowadays. Blockchain handles sensitive data, which requires a layer of complexity such as cryptography, consensus mechanism, transaction immutability, and peer-to-peer network to manage decentralised data features, making blockchain development complex and challenging. Smart contracts are self-executing digital contracts that are stored on a blockchain. They enable parties to transact with each other without the need for intermediaries, making transactions faster, cheaper, and more secure. Through the synergy of blockchain and smart contracts, it is possible to create decentralised applications that can be used in a wide range of industries, bringing increased transparency, security, and efficiency to traditional business models. Various software methodologies have been introduced to support the development of blockchain applications systematically as well as reduce the development complexity. Despite the introduction of methodologies, further refinements are required to enhance their effectiveness for modelling. Moreover, all models are generic and do not consider the blockchain concept as first-class entities. Hence, a gap exists in extending and transforming the current blockchain modelling and practices when developing a blockchain application. This research introduces a new insight into blockchain-based application development through extended Agent-Oriented Modelling (AOM) to discover blockchain opportunities and commit to the growth of the technology. AOM is a methodology for complex socio-technical system development and can become an alternative methodology for blockchain application development. A walkthrough example of how extended AOM is used to model a blockchain-based lottery application is presented in this thesis. It reveals the issues and limitations of AOM, an extension of AOM, and the potential of extended AOM and turns it into a software methodology for blockchain-based application development. Furthermore, the extension of AOM is discussed to deal with the software challenges the novices faced. Three experiments were conducted with the novices. The novices (e.g. students) were required to construct the case study using Unified Modelling Language (UML), AOM, and extended AOM. The first experiment aimed to compare UML and AOM, which resulted in a mean mark of 15.636 for UML and 17.000 for AOM. The result revealed the potential usage of AOM to model blockchain applications apart from UML. The second experiment was targeted to specify the limitation of AOM in blockchain application development. From the findings, 78% of the students failed to model blockchain requirements through AOM because it needs to be more specific to identify blockchain needs. Hence, an extended AOM is needed. The third experiment intended to test the usability of the extended AOM and AOM. 44% of the students managed to capture more than 80% of blockchain requirements through extended AOM. On the other hand, 51% of the students scored between 20% to 60% on the blockchain requirement, and only 5% failed to model blockchain requirements through extended AOM. The results reveal the usage of the extended AOM to model blockchain-enabled applications comprehensively.

Keywords: Blockchain, smart contract, methodology, agent-oriented modelling, Ethereum

Metodologi Diperluaskan Berdasarkan Permodelan Berorientasikan Ejen untuk Pembangunan Aplikasi Blockchain

ABSTRAK

Pembangunan aplikasi blockchain (blok rantai) makin jadi perhatian pada masa kini. Blockchain menyimpan data sensitif yang dilakukan degan mata wang kripto, algoritma konsensus, transaksi, peer to peer dan terdesentralisasi. Kontrak pintar adalah kontrak digital yang dapat melaksanakan diri sendiri dan disimpan pada blockchain. Mereka membolehkan pihak-pihak bertransaksi dengan satu sama lain tanpa memerlukan pihak tengah, menjadikan transaksi lebih cepat, lebih murah, dan lebih selamat. Dengan gabungan penggunaan teknologi blockchain dan kontrak pintar, ia dapat mencipta aplikasi terdesentralisasi yang boleh digunakan dalam pelbagai industri, untuk menaikan taraf keselamatan, dan kecekapan kepada model perniagaan tradisional. Pelbagai metodologi perisian telah diperkenalkan untuk menyokong pembangunan aplikasi blockchain secara sistematik dan mengurangkan kerumitan pembangunan. Walaupun metodologi telah diperkenalkan, taraf kesan modelling masih perlu dipertingkatkan. Selain itu, semua model adalah generik dan tidak menganggap konsep blockchain sebagai entiti kelas pertama. Oleh itu, jurang wujud dalam memanjangkan dan mengubah permodelan dan amalan blockchain semasa semasa membangunkan aplikasi blockchain. Penyelidikan ini memperkenalkan pandangan baharu ke dalam pembangunan aplikasi berasaskan rantaian blok melalui Permodelan Berorientasikan Ejen (AOM) lanjutan untuk menemui peluang rantaian blok dan komited terhadap pertumbuhan teknologi. AOM ialah metodologi untuk pembangunan sistem sosial-teknikal yang kompleks dan boleh menjadi metodologi alternatif untuk pembangunan aplikasi blockchain. Contoh panduan tentang cara lanjutan AOM digunakan untuk memodelkan aplikasi loteri berasaskan blokchain dibentangkan dalam tesis ini. Ia *mendedahkan isu dan had AOM, lanjutan AOM, dan potensi AOM lanjutan dan mengubahnya menjadi metodologi perisian untuk pembangunan aplikasi berasaskan blokchain. Tambahan pula, lanjutan AOM dibincangkan untuk menangani cabaran perisian yang dihadapi oleh pemula. Tiga eksperimen telah dijalankan dengan orang baru. Orang baru (contohnya pelajar) dikehendaki membina kajian kes menggunakan Bahasa Pemodelan Bersepadu (UML), AOM dan AOM lanjutan. Eksperimen pertama bertujuan untuk membandingkan UML dan AOM, yang menghasilkan markah min 15.636 untuk UML dan 17.000 untuk AOM. Hasil eksperimen mendedahkan potensi penggunaan AOM untuk memodelkan aplikasi blockchain selain daripada UML. Eksperimen kedua disasarkan untuk menentukan had AOM dalam pembangunan aplikasi blockchain. Daripada penemuan, 78% daripada jumlah pelajar gagal memodelkan keperluan blockchain melalui AOM kerana ia perlu lebih khusus untuk mengenal pasti keperluan blockchain. Oleh itu, AOM Lanjutan diperlukan. Eksperimen ketiga bertujuan untuk menguji kebolehgunaan AOM dan AOM Lanjutan. 44% daripada jumlah pelajar berjaya menangkap lebih daripada 80% keperluan blockchain melalui AOM Lanjutan. Sebaliknya, 51% daripada jumlah pelajar mendapat markah antara 20% hingga 60% pada keperluan blockchain, dan hanya 5% gagal memodelkan keperluan blockchain melalui AOM Lanjutan. Hasilnya mendedahkan penggunaan AOM Lanjutan untuk memodelkan aplikasi yang didayakan blockchain secara menyeluruh.*

Kata kunci: *Blok Rantai, kontrak pintar, metodologi, permodelan berorientasikan ejen, Ethereum*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABCDE	Agile Block Chain Development Engineering
AOM	Agent Oriented Modelling
CIM	Computation Independent Modelling
DAPP	Decentralised Application
FCSIT	Faculty of Computer Science and Information Technology
HOMER	Human-Oriented Method for Eliciting Requirement
MDE	Model Driven Engineering
PIM	Platform-Independent Modelling
PSM	Platform-Specific Modelling
UML	Unified Modelling Language

CHAPTER 1

INTRODUCTION

1.1 Introduction

With the rapid growth of innovative technology, blockchain technology has received much attention.

Blockchain has introduced a new decentralised concept to solve synchronised problems in a traditional database (Lin & Liao, 2017). Blockchain technology uses a consensus mechanism to ensure all blockchain nodes agree on the same message, ensures that the newest block has been appropriately added to the chain, guarantees that the message stored by each node is the same, and protects against malicious attacks. A node is created when a user joins the blockchain network. The primary function of a blockchain node is to verify each block of network transactions. A unique identification distinguishes each node from the others. The data will not be lost easily, even if one of the nodes fails or encounters a physical disaster.

The main advantages of blockchain technology are a decentralised ledger, immutable blocks, and strong security (Kim & Laskowski, 2018; Nakamoto, 2008). Blockchain replicates the data over the nodes that do not require any governing authority. Since there is no central governance point for a blockchain network, the blockchain users verify and validate the data as a node. The replication and synchronisation of data across networks help multiple organisations trace and track data.

Imagine if the blockchain network has one million nodes (users). It means there are millions of records in the blockchain network. For security purposes, the malicious actor is required to alter 51% of the nodes to make changes simultaneously (Lin & Liao, 2017). Hence, such strong security protection provided by blockchain can enhance the prevention of fraud and unauthorised activities.

Blockchain is used in healthcare systems to manage electronic health records (EHRs), validate and track patients' medical history, and trace research methods on fabricating drugs (Kombe, Ally, & Sam, 2018). The patient's data is securely held on blockchain and is accessible and traceable by another hospital/clinic through the healthcare system. When an accident occurs, and a patient is sent to the nearest hospital, the medical staff can easily access the patient's information accurately and instantly. At the same time, blockchain technology will ensure the security of the patient records by tracing who accessed the information and controlling the authorisation to amend the records (Azogu, Norta, Papper, Longo, & Draheim, 2019).

In the financial industry, blockchain has removed the central authority and enhanced trust (Sukheja, Indira, Sharma, & Chirgaiya, 2019). The blockchain is treated as a global ledger to record financial transactions, asset management, insurance claims, and global transactions (Sukheja et al., 2019). With blockchain, the financial industries can trace transactions and supporting documentation effectively and transparently. Meanwhile, the clearing and settlement of transactions can be handled transparently, at low risk and effectively (Knezevic, 2018). Using blockchain will increase trust in any transaction in a secure manner.

The blockchain's decentralised, transparent, and synchronised behaviour in supply chain management leads to transparent and accurate end-to-end tracking. Blockchain does benefit the supply chain management business when two or more organisations (manufacturer, supplier, seller, delivery, and buyer) cooperate. Organisations can digitise physical assets and generate a decentralised, immutable record of all transactions, allowing for asset tracking from manufacturing to delivery or end-user use. Meanwhile, the status of each physical asset can update and record on the blockchain network in nearly real-time. As a result, traceability and transparency can improve organisational interactions by improving productivity, lowering risks, and being cost-effective. (Aslam, Saleem, Khan, & Kim, 2021; Bumblauskas, Mann, Dugan, & Rittmer, 2019).

Smart contract was introduced in 1994 by Nick Szabo. It is a digital agreement between two people written in computer code (Lallai, Pinna, Marchesi, & Tonelli, 2020). A smart contract is a digital self-enforcing contract with rules and agreements encapsulated in lines of codes in object-oriented programming such as Solidity, Golang, JavaScript, C++, Java, etc. According to the founder, a vending machine is the most appropriate example of a smart contract. Consumers buy and get the item at a vending machine by paying the amount of money without the existence of the middleman (the contract enforcer). In essence, a smart contract allows us to perform a transaction without formal legal documents and the existence of an intermediary (lawyers and judges). Compared to the traditional way of performing a transaction, the smart contract would help achieve the same purpose autonomously, and the transaction is traceable and immutable. The idea behind this smart contract is to increase the level of trust of the users and live forever on the blockchain network.

A smart contract is an agreement in a code that will automatically enforce and make a transaction when the condition is met. A smart contract is invented to eliminate the existence of a middleman (Poels, Kaya, Verdonck, & Gordijn, 2019). Using smart contracts will save documentation, time, and cost. Furthermore, it could solve the trust issue when trading online involving multiple parties like supply chain management. Imagine when a buyer buys an asset or liability. A smart contract replaces a lawyer and automatically examines the agreements made in the contract. A transaction is performed when the agreement is achieved.

On the other hand, a smart contract can reduce costs and improve efficiency. However, the smart contract does more than replace the contract in the real world. Still, they strengthen themselves in the software world by adding decentralisation, autonomy, and immutability.

Ethereum is a blockchain-based platform founded by Vitalik Buterin in 2013 and released in 2015. It is the most common platform for creating a smart contract in the Solidity programming language. The smart contract can be developed using open-source code tools such as Remix straight from the browser. In addition, Remix supports debugging, testing, and deploying to ease development. The code compilation will return the Application Binary Interface (ABI) in JSON format and bytecode. ABI is a list of functions and arguments in a JSON format to access binary data in the contract. Bytecode is binary data converted from a smart contract that will run and be stored on Ethereum Virtual Machine (EVM) under a contract address. EVM is a runtime environment that runs on every node to execute smart contract bytecode.

Gas is required as an execution fee to perform a transaction and execute a smart contract. Gas is a unit measurement for computational performance to reward the miners; any operation executed on Ethereum requires some gas. The unit for gas in Ethereum is known as Ether. A crypto wallet that holds the Ether can interact and transfer gas to an account to account or account for a smart contract. The most common crypto wallet is MetaMask, Mist, and MyEtherWallet. Ethereum accounts are classified into two types which are Externally Owned Accounts and Contract Accounts. An externally Owned Account is the wallet address controlled by the private key and holds an ether balance to perform a transaction. Contract Account can only be activated by Externally Owned Account. The contract account has an associative code and contract balance. It is activated when there is a transaction. The contract address can be retrieved when the contract owner deploys the contract to the Ethereum network. Gas is required to deploy a contract; the gas fee is to pay EVM for the executive power; the higher the gas fee, the faster the block can be mined.

Software engineering methodologies have been introduced to aid blockchain application development. However, more study is needed to address standard tools and methodologies for implementing blockchain applications, especially software engineering methodology and framework for designing blockchain applications (Jiang et al., 2022). It has been reported that blockchain developers should focus more on software engineering practices to have a specific and systematic way to analyse and ensure the software quality in the blockchain development process (Chakraborty et al., 2018). In addition, proper software engineering practices could provide crucial actions, guidelines, checklists, design principles, and heuristics for mitigating, tracing, and correcting the source of vulnerabilities and errors (Fahmideh et al., 2022). These practices are essential to guarantee high-quality software that achieves the desired results while staying within budget and time restrictions. Meanwhile, using software engineering practices in a blockchain application to increase sustainability is crucial (Lund et al., 2019).

Although the methodologies have been introduced (Fridgen et al., 2018; Marchesi, Marchesi, & Tonelli, 2018; Ibba, Pinna, & Pani, 2017; Frantz & Nowostawski, 2016), there are still problems pertaining to unsuitable software models in demonstrating the internal architecture of the blockchain, such as gas transaction and decentralisation of data.

Moreover, all the models are generic and do not take into blockchain concepts as first-class entities. As blockchain is a new emerging technology, it can be challenging to understand the technical concepts and handle unforeseen constraints and uncertainties (Almeida, Albuquerque, & Silva, 2019).

Hence, a gap exists in using, extending, and transforming the current blockchain modelling and practices when developing a blockchain application. This research introduces insight into blockchain application development through extended Agent-Oriented Modelling (AOM). AOM is a methodology for complex socio-technical system development and can become an alternative methodology for blockchain application development. As AOM has not been explored in blockchain application development, to what extent it is useful for blockchain application development is yet to be explored.

This research introduces an extended AOM to provide a systematic approach to developing a blockchain application. With extended AOM, the blockchain application developer can identify the problems, determine the needs of the blockchain, and design the blockchain application.

1.2 Problem Statement

Blockchain technology is a new emerging technology and complicated. Blockchain developers' skills and experience are still immature. Hence, blockchain developers should follow a standard Software Engineering practice throughout development. Bosu, et al. (2019) addressed that current blockchain development lacks modelling tools, and metrics could not better represent blockchain applications. Traditional use case diagrams, activity diagrams and state diagrams are unsuitable for representing blockchain applications.

It still needs clarity on how the application should be performed (Deshpande, Stewart, Lepetit, & Gunashekar, 2017). The immaturity of blockchain technology commits a need for more understanding for developers and stakeholders to implement blockchain on potential use cases. Fridgen, et al. (2018) mentioned the importance of systematic methodology to help us to understand and create blockchain use cases. Gordijn, Wieringa, Ionita, and Kaya (2019) mentioned that a sustainable blockchain use case is crucial to the business ecosystem. Having a proper methodology is like a project management technique to lead us to plan, design, develop, and analyse a project. Clarity on implementing smart contracts through blockchain may restrict the functionality of the smart contract. Many people still misunderstand smart contracts as electronic contracts converted from traditional ones (Deshpande et al.,2017).

To support blockchain application development, software development methodology is vital to serving as a guide and blueprint. While software methodologies are introduced, there still needs to be more work to elicit blockchain requirements, perform analysis, design, and implement blockchain applications. Blockchain can be challenging to understand technical concepts and handle unforeseen constraints and uncertainties, especially for novices. Hence, there is a gap in developing blockchain applications that lie in understanding how to adapt and apply software methodology to the unique challenges of blockchain technology and developing new methodologies and practices tailored to the requirements of blockchain development to create a sustainable use case.

1.3 Research Objectives

The objectives of this research are:

i. To compare the suitability of Agent-Oriented Modelling (AOM) and Unified Modelling Language (UML) notations for modelling blockchain applications to identify the notation tailored to the requirements of blockchain development to create a sustainable use case.