Chibuzor Joseph Udokwu

A MODELLING APPROACH FOR BUILDING BLOCKCHAIN APPLICATIONS THAT ENABLES TRUSTABLE INTER-ORGANIZATIONAL COLLABORATIONS

ACTA UNIVERSITATIS LAPPEENRANTAENSIS 1037

Chibuzor Joseph Udokwu

A MODELLING APPROACH FOR BUILDING BLOCKCHAIN APPLICATIONS THAT ENABLES TRUSTABLE INTER-ORGANIZATIONAL COLLABORATIONS

Dissertation for the degree of Doctor of Science (Technology) to be presented with due permission for public examination and criticism in the Auditorium 1314 at Lappeenranta-Lahti University of Technology LUT, Lappeenranta, Finland on the 26th of September, 2022, at noon.

Acta Universitatis
Lappeenrantaensis 1037

Supervisors Assoc. Professor A.K.M. Najmul Islam
LUT School of Engineering Science
Lappeenranta-Lahti University of Technology LUT
Finland

Alex Norta (PhD)
Researcher, Tallinn University & CEO Dymaxion
Tallinn Estonia

Reviewers Professor Matti Rossi
Department of Information and Service Management
Aalto University
Finland

Assoc. Professor Josef Küeng
Institute for Application-Oriented Knowledge Processing
Johannes Kepler University Linz
Austria

Opponent Professor Matti Rossi
Department of Information and Service Management
Aalto University
Finland

ISBN 978-952-335-848-5
ISBN 978-952-335-849-2 (PDF)
ISSN-L 1456-4491 (Print)
ISSN 2814-5518 (Online)

Lappeenranta-Lahti University of Technology LUT
LUT University Press 2022

Abstract

Chibuzor Udokwu
A modelling approach for building blockchain applications that enables trustable inter-organizational collaborations
Lappeenranta 2022
106 pages
Acta Universitatis Lappeenrantaensis 1037
Diss. Lappeenranta-Lahti University of Technology LUT
ISBN 978-952-335-848-5, ISBN 978-952-335-849-2 (PDF), ISSN-L 1456-4491 (Print), ISSN 2814-5518 (Online)

Inter-organizational collaborations (IOCs) represent a significant aspect of any organization because organizations depend on each other to perform functions that are not part of their core competencies. There are trust issues due to lack of security, interoperability, and transparency in systems and processes that support inter-organizational collaborations. First, due to security issues, information exchange between collaborating parties can be modified and manipulated by the party responsible for managing the system that supports the collaboration. Second, due to the lack of interoperability, the real-time availability of data between collaborating parties cannot be guaranteed. Finally, lack of transparency implies that illegitimate manipulations performed by the centralized party that manages the collaboration cannot be easily detected.

Blockchain technology has shown potential in addressing the trust issues associated with inter -organizational collaborations. Blockchain is a peer-to-peer network that provides the capabilities for decentralized storage of data by replicating information across the nodes of the peers participating in a given network. This ensures immutability and real-time access to information stored on the blockchain. With decentralized consensus, blockchain provides a transparent governance system for adding new records or information to the network. With smart contracts, inter-organizational business processes and governance systems can be coded into computer-executable programs to ensure automated accountability of the parties involved in organizational collaborations. Computer programs that adopt and implement the properties of blockchain technology such as decentralized storage, consensus, and smart contracts are referred to as decentralized applications (DApps).

Nevertheless, despite the opportunities provided by blockchain DApps in addressing security, interoperability, and transparency issues in organizational collaborations, there exists a lack of a well-defined and structured software-engineering approach for designing and developing blockchain applications. This thesis proposes the development of a modelling approach for building blockchain DApps that specifically addresses problems in inter-organizational collaborations by applying agent-based modelling and UML concepts. The modelling approach developed in this thesis, referred to as Trustable DApp Modelling (T-DM), provides the possibility for outlining functions that enable the execution of complex interactions in organizational collaborations in a trustable manner. The development of the T-DM approach and corresponding support tool follows the design-science research method involving the rigorous creation and evaluation of artefacts. Evaluations are performed to understand the semantic correctness of the T-DM approach and the effectiveness of the support tool in creating blockchain applications design models.

Keywords: Blockchain applications, inter-organizational collaborations, trustable systems, software modelling

Acknowledgements

I hereby acknowledge Alex Norta for his guidance throughout the process of conducting research studies that resulted in this thesis. I appreciate the effort of Najmul Islam in the documentation of this thesis work. I also recognize the contributions of Raimundas Matulevičius and Patrick Brandtner towards the implementation of the support tool and evaluations carried out in this thesis. Lastly, I acknowledge the partial funding of the research conducted in this thesis received from the Erasmus+ BLOCKS project led by Roxana Voicu-Dorobantu. This research is partly funded from Estonian personal research funding "Team grant (PRG) project PRG1641".

Chibuzor Udokwu
September, 2022
Lappeenranta, Finland

This thesis is dedicated to my lovely wife Chidimma Prisca Udokwu and my beloved mother Elizabeth Udokwu.

Contents

List of publications

This dissertation is based on the following papers. The rights have been granted by publishers to include the papers in the dissertation.

I. Udokwu, C., Kormiltsyn, A., Thangalimodzi, K., and Norta, A. The state of the art for blockchain-enabled smart-contract applications in the organization. In 2018 Ivannikov Ispras Open Conference (ISPRAS) (pp. 137-144), (2018, November). IEEE.

II. Udokwu, C., Anyanka, H., and Norta, A. Evaluation of Approaches for Designing and Developing Decentralized Applications on Blockchain. In Proceedings of the 2020 4th International Conference on Algorithms, Computing and Systems (pp. 55-62), (2020, January).

III. C. Udokwu, A. Norta. Deriving and Formalizing Requirements of Decentralized Applications for Inter-Organizational Collaborations on Blockchain, Arabian Journal for Science and Engineering, 46, (2021, March) Springer.

IV. Udokwu C., Norta A., Wenna C., "Designing a Collaborative Construction-Project Platform on Blockchain Technology for Transparancy, Traceablity and Information Symmetry." Service Sciences and software-engineering Conference. (2021), ACM.

V. Udokwu, C., Brandtner, P., Norta, A., Kormiltsyn, A., and Matulevicius, R. Evaluation of the DAOM framework and Tool Support for Designing Blockchain Decentralized Applications. International Journal of Information Technology, Springer 2021.

Author's contribution
The author's contributions to the published articles are outlined below for each article in the publication list.

I. I was the lead author, involved in the data collection and the analysis of the results, prepared the figures, and was also the lead writer of the article.

II. I was the lead author, interpreted the results, prepared the figures, and was involved in writing the article.

III. I, as the lead author, reviewed relevant articles, developed the model verification method, analyzed and interpreted the results, prepared the figures, and was also the lead writer of the article.

IV. I, as the lead author, developed the architecture of the proposed platform, prepared the figures and was also the lead writer of the article.

V. I was the lead author, I was in involved in the development of the support tool and the evaluation method. I was also involved in organizing the workshops, collecting evaluation results, and was also the lead writer of the article.

Abbreviations

AML	Agent Modelling Language
AOM	Agent Oriented Modelling
AAOM	Agile Agent Oriented Modelling
AOM4STS	Agent Oriented Modelling for Socio-Technical systems
AT	Acceptance Test
BM	Building Model
BO	Building Object
BPMN	Business Process Modelling Notation
CoPM	Collaborative Construction Project Management Platform
DApp	Decentralized Applications
DSML	Domain Specific Modelling Language
DSR	Design Science Research
DTD	Document Type Definition
FLG	First-Level Goals
FLR	First-Level Requirements
FMT	Framework, Method and Technique
MaSE	Multi-agent Systems Engineering
ID	Identity Document
IDE	Integrated Development Environment
IP	Intellectual Property
IOC	Inter-Organizational Collaboration
IOT	Internet of Things
IS	Information System
KYC	Know Your Customer
LTL	Linear Temporal Logic
NFR	Non-Functional Requirements
OEM	Original Equipment Manufacturer
OOP	Object Oriented Programming
PK	Public Key
PN	Petri-net
PoW	Proof of Work
PoS	Proof of Stake
RQ	Research Question

SCM	Supply Chain Management
S/W	Strengths and Weakness
STS	Socio-Technical System
UML	Universal Modelling Language
UT	Unit Test
XML	Extensible Markup Language

1 Introduction

1.1 Thesis Motivation

Inter-organizational collaborations (IOCs) constitute important aspects of business operations because organizations are usually not self-sufficient, and therefore, depend on each other for specific business functions where their expertise is limited [17, 30]. In Addition, services that are not part of an organization's core function are procured from third parties [71]. These types of multi-party collaborations to achieve specific organizational goals involve the exchange of data and shared processes between organizations. As a result, multi-dimension information exchange between different organizations, requires security and transparency of data exchanged. Current systems for managing inter-organizational collaborations are not trustable because they are usually centralized, lack data transparency, are not interoperable, and are insecure [1, 41, 49]. Due to insecurity, access control cannot be enforced on shared data, and due to centralization, activities of collaborating parties cannot be independently verified. In addition, a lack of interoperability leads to a manual, inefficient exchange of information among the collaborating parties. As a result, traditional information systems (IS) are not suitable for executing inter-organizational processes. The novel blockchain technology has shown great potential in addressing trust issues in IOCs due to decentralized- and tamper-proof data storage system [78].

Recent studies have identified and demonstrated trust issues in different forms of IOCs between organizations. The study [49] shows the collaboration between organizations involved in airline turnaround processes. One of the main information assets exchanged between these organizations are service requirement documents containing specifications for delivering a given service by the collaborating organizations. An analysis of associated security risks shows that information contained in the requirements can easily be manipulated. This is due to security issues associated with the systems that support the turnaround processes. As a result, additional security control is needed to ensure the trustability of the service requirement documents. Study [41] attributes the interoperability issues in the healthcare booking or reservation process as one of the major problems in healthcare processes, which results in the rising cost of healthcare. In addition, due to interoperability issues, healthcare providers cannot share information in real-time to allocate patients to different hospitals during peak times of healthcare demand. Trust issues also adversely impact IOCs in the delivery of public services. The study [1] shows that lack of transparency in managing collaborations in public sector supply chains, such as public procurement, is one of the leading causes of corruption in developing countries. This is because the information systems supporting the procurement processes are centralized and are comprised of mainly manual processes.

Several studies have been conducted to address trust issues (such as transparency, interoperability and security) in IOCs. The study [83] proposes a pragmatic conceptual model for trust management for IOCs. The findings show that establishing trust in certain IOC processes are impossible due to the inequality between collaborating parties and the possibility of manipulating collaboration rules by the most powerful party. Manipulation exists because the supporting collaboration information system is centralized and controlled by powerful parties. As a result, the entire collaboration process lacks transparency. The study [82] proposes collaborative governance that ensures that actors and organizations in IOCs are accountable for their activities. However, implementing such a decentralized governance method on traditional systems that support IOCs is nearly impossible.

Blockchain technology is a decentralized ledger that provides a tamper-proof method for storing data generated in the activities of a given network [32, 31]. Blockchain technology provides the potential to address the trust problems that affect IOCs. In [78], the authors show that the main purposes of applying blockchain in organizations are to improve transparency, security and interoperability of data. Blockchain technology provides a transparent method for verifying and validating activities of all the parties (including powerful parties) in an IOC [70]. Also, blockchain technology provides decentralized storage of information by replicating data across the nodes of participating parties in IOCs thereby addressing interoperability issues [78]. Several blockchain-based applications have been developed to address the security, transparency and interoperability issues in IOCs [90, 12, 24, 1, 87]

Notwithstanding the opportunities blockchain technology delivers in addressing inter-organizational collaborations problems, no aligned design framework exists for developing blockchain decentralized applications (DApps) for supporting IOCs [77]. Existing software-engineering-focused methods for building DApps do not address the transaction cost, scalability, and usability problems in the technology [47]. In addition, the process-reduction method for executing business processes on the blockchain does not capture all of the stages of software modelling [27]. The objective of this thesis is, therefore, to address these gaps by developing a modelling approach for designing and developing blockchain applications that enable IOC.

Blockchain has some inherent weaknesses. Blockchain technology has complexity and usability issues that have mitigated the adoption of blockchain applications in organizations [78]. Also, blockchain DApps have oracle problems which imply transparency issues in exchanging information between internal and external components of a DApp software system [3, 23]. The application of Agent-Based Modelling concepts and UML approaches can potentially address the inherent issues in blockchains while also producing a new modelling approach for building DApps. By applying modelling concepts defined in agent-based approaches [72, 85, 19], complexity problems associated with blockchain technology can potentially be addressed. Agent-based approaches provide simplified concepts for outlining the initial requirements of complex systems, such as the blockchain applications that enable inter-organizational collaborations.

The Agent-Based Modelling approach can also be applied in addressing the oracle problems in the blockchain by identifying and specifying the agent types, the functions they perform in a proposed system, and defining their properties based on the identified functions [72]. In establishing the software agents and outlining their properties and rules for interaction with external components in a designed DApp, trust issues associated with such interactions can be solved by specifying the conditions for an interaction. The UML diagram approaches, such as components and sequence diagrams, provide the possibility for identifying the components and interaction interfaces that exist in a designed system [9]. As a result, complex interactions that are contained in IOC applications and the information exchanged between several components in a designed blockchain application can easily be outlined using UML approaches.

Thus, a software engineering approach for building DApps for IOCs can be developed by combining the modelling methods in agent-based system modelling and UML diagram approaches. Existing agent-modelling and UML approaches cannot be used in their current forms for designing blockchain DApps. This is because, the traditional agent modelling and UML approaches are incapable of outlining the novel functional properties of the blockchain such as on-chain executions, tokenizations and on-chain events [13]. In blockchain systems certain functions when executed, result in a state change of the blockchain, thus generating transactions in the network. Such generated transactions can also involve the exchange of utility values between components in the system. The functions that result in a state change in the blockchain are referred to as on-chain functions, and the resulting transactions are referred to as on-chain events; the utility values exchanged are referred to as tokens. The main focus of this thesis is, therefore, to develop a modelling approach for building blockchain DApps for IOC by combining agent-based modelling and UML approaches and extending their modelling notations to capture the novel functions of the blockchain. Since the purpose of the modelling approach developed in this thesis is to create DApps that address the trust management issues existing in various IOCs, the developed approach is henceforth referred to as the trustable DApp modelling (T-DM) approach.

1.2 Development of Research Questions

The main research gap addressed in this thesis is the absence of a proper software-engineering focused approach for developing DApps that enable trustable IOCs. The research question helps to provide clarity on how the research gap identified in this thesis is addressed. The main research question addressed in this thesis is *How to develop a modelling approach that supports the design and development of DApps for trustable IOCs?* In order to properly address the main research question of this thesis,four research questions are derived:

1. RQ1: How to develop a model-driven (T-DM) approach that supports the design and development of blockchain applications for IOC?
2. RQ2: How to develop the ontology of the T-DM approach for building DApps?
3. RQ3: How to develop a blockchain application for trustable organizational collaborations using the T-DM approach?
4. RQ4: How to implement and evaluate the T-DM approach and support tool for designing blockchain applications?

1.2.1 Research question 1

The first research question (RQ1) enhances the understanding of the current approaches for building DApps and systematically develops the model-driven approach based on the findings. This is done by first identifying the use of blockchain applications in organizations and their limitations. Then, the current approaches and methods for developing blockchain applications are analyzed by examining their strengths and weakness. Lastly, a modelling approach for building blockchain applications that address the weaknesses of the current approaches and methods is developed.

A systematic literature review (SLR) of studies comprising of frameworks, methods, working prototypes and simulations that demonstrate the application of smart contracts in organizations is conducted to understand the challenges and problems mitigating the use of blockchain applications in organizations. The SLR is conducted in Publication I. To understand the current approaches and methods for developing blockchain applications, data is collected through expert interviews and related literature reviews examining the frameworks, methods and techniques used for building blockchain applications. The strengths and weaknesses of the identified methods are analyzed by examining how they address the important challenges and problems affecting the use of blockchain applications in organizations. Thus, by understanding the current methods used in building blockchain applications, and establishing their strengths and weaknesses, a new modelling approach is developed that addresses the weaknesses in the current methods. These are discused in Publication II.

1.2.2 Research question 2

The second research question (RQ2) helps in specifying the ontology of the modelling approach developed in this thesis. The ontology shows the description of the elements and diagram types present in the modelling framework as well the relationship between the modelling elements and the diagram types. The ontology also shows the syntactic aspect of the modelling approach covering the graphical representations of the elements in the modelling framework, formal descriptions of the diagram types in the framework, and validation rules for verifying the correctness of the developed models.

In describing the ontology of the developed modelling approach, first, a review of related articles is conducted to identify useful requirement sets for blockchain applications in the context of organizational collaborations. Second, reference-architectural (requirement) sets are generated for graphical descriptions of the diagram types and elements in the developed modelling approach. Finally, a formal description of the modelling approach showing the diagram types, diagram elements, their relationships, and validation rules is specified using the XML document type definition (XML-DTD). A formal description of the modelling approach with XML-DTD ensures structural correctness and consistency of blockchain application models designed using the developed modelling approach. These are shown in Publication III.

1.2.3 Research question 3

The third research question (RQ3) shows the development of a blockchain application for trustable inter-organizational collaboration use-case by enabling transparent, traceable and symmetric exchange of information between collaborating parties. This is achieved by specifying the requirements and architectural heuristics of developed blockchain applications using the T-DM approach.

The requirement sets of the developed blockchain application show the functional, and non-functional requirements of the designed application. The functional requirements outline specific goals that have to be achieved in the given context of inter-organizational collaborations. The non-functional requirements specify the software properties showing how a particular function is executed. The architectural heuristics of developed blockchain application covers the static and dynamic concepts for modelling the behaviour of the designed application. The static architecture of the developed application shows the main components and interfaces for exchanging information between the components. The dynamic behaviour shows sequential events and transactions that are generated and stored on the blockchain during the execution of a particular use-case of the designed application. The development of blockchain applications using the T-DM approach is shown in Publication IV.

1.2.4 Research question 4

The fourth research question (RQ4) shows the implementation of the T-DM approach and the development of the support tool that enables the effective modelling of blockchain applications for IOCs using the developed modelling approach.

To implement the T-DM modelling approach, a meta-model is used to provide a single representation of all the modelling concepts defined in the T-DM approach and the connections used in realizing these concepts. To develop the T-DM support tool and create the custom modelling notations that define the diagram elements, UML elements notations are extended. The T-DM modelling approach is evaluated by checking the semantic properties of the modelling approach outlined in the T-DM meta-model and the practical usefulness of the modelling approach. The support tool is evaluated by checking the usability and efficiency of the tool in replicating existing blockchain application models generated using the T-DM approach.

1.3 Thesis Overview

1.3.1 Thesis contexts

As shown in Figure 1.1, three contexts provide the background for this thesis. The first is trust management in inter-organizational collaborations and their associated problems. The problems are outlined by transparency in verifying correctness of generated in collaborative processes, security issues of information exchanged in collaborations, and interoperability issues that affect real-time availability of information generated in organizational collaborations. The existing trust management frameworks are presented in detail in Section 2.1.

The second context of the thesis is about blockchain DApp development showing how the technologies that support blockchain can be used to address the trust issues in IOCs. Furthermore, this context on Blockchain DApp development outlines the current issues that affect the adoption of blockchain in organizations. Detailed background on blockchain technology and its applications in organizations are presented in Section 2.2.

The third and last context of this thesis, is the presentation of modelling approaches and concepts that can be adapted to develop a novel approach for building blockchain DApps. Agent-based modelling and UML provide the potential for addressing the inherent limitations in blockchain such as complexity, usability, and oracle problems. By combining these modelling concepts and extending their notations to capture the advanced functionalities of blockchain, a new modelling approach for building DApps can be created. The background on agent-based approaches and the selected approach for T-DM approach development is presented in Section 2.3. The background on UML diagrams and the selected diagrams for the T-DM approach development is presented in Section 2.4.

1.3.2 Research contributions

The objective of this thesis is to develop a comprehensive approach for building DApps that enable trustable inter-organizational collaborations. To achieve these objectives and answer the research questions developed for this thesis, various research studies are conducted generating several original contributions in this thesis. The original contributions resulting from these studies are outlined as follows.

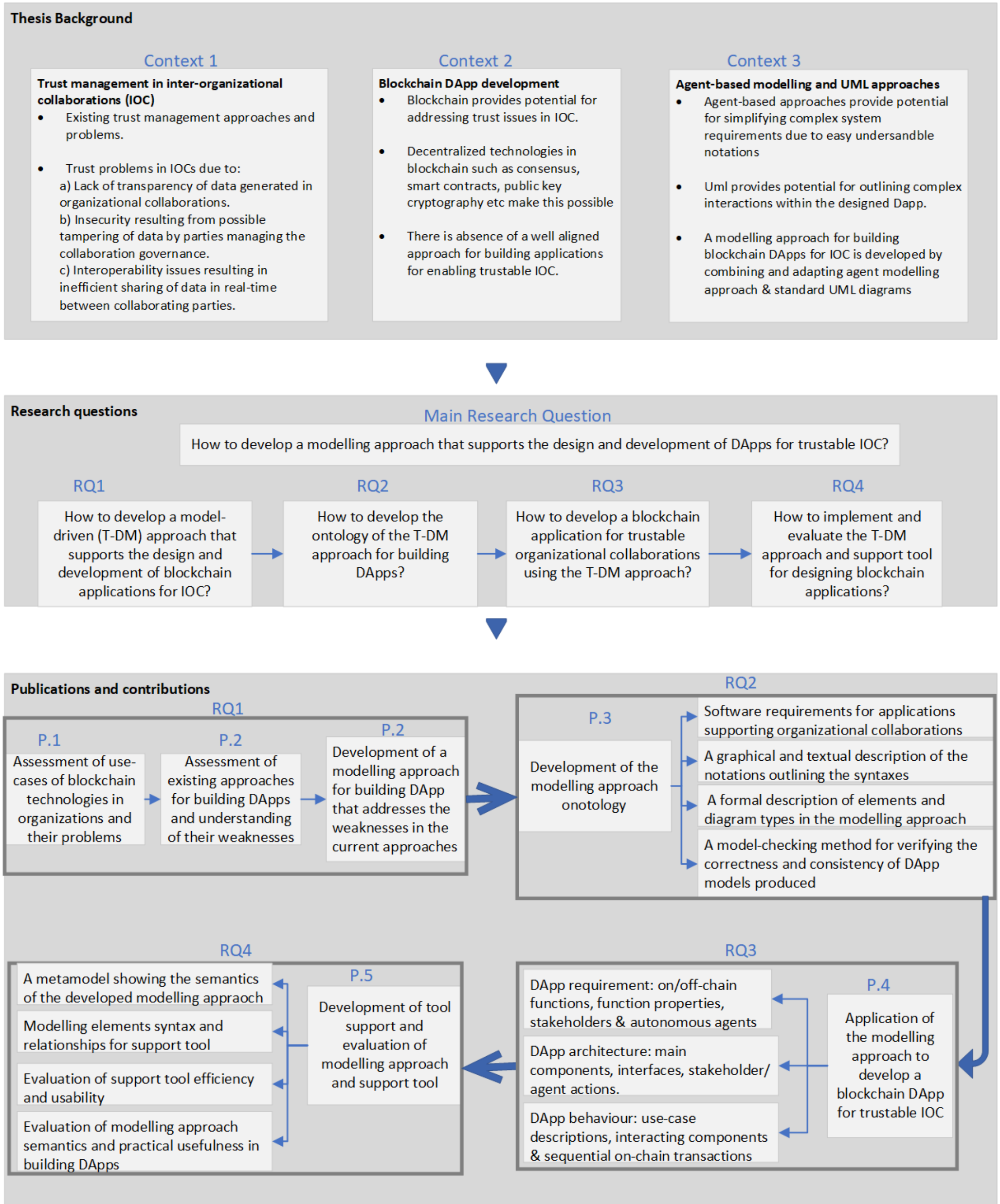


Figure 1.1: Thesis overview- contexts, research questions and contributions

1. **Identification of blockchain as a potential technology that addresses the trust issues that exist in inter-organizational collaborations**: A systematic literature review research conducted in [78] analyzing the application and use of blockchain technology in organizations shows that addressing transparency, interoperability and security issues are among the top reasons for adopting blockchain in organizations. This contribution is the basis of the motivation of this thesis in applying blockchain technology in addressing the trust issues in organizations. This contribution is summarized in Section 4.1 of this thesis.

2. **This thesis shows that the examined modelling approaches and common software development techniques are not suitable for building blockchain DApps for organizational use as they do not address the inherent limitations in block-chain technology**: Research conducted in [77] analyzed current approaches for building blockchain DApps by examining relevant literature and interviewing experts in the domain. The result shows that common approaches for building software such as agile, waterfall, business process modelling notation (BPMN), and freehand sketches, are not suitable for building blockchain applications. The result also shows that the modelling approaches that exist currently do not fully address the complexity, usability, and storage cost, scalability issues that exist within blockchain technology. This contribution is summarized in Section 4.2.

3. **Development of a modelling approach for building DApps that addresses weaknesses in current software development techniques by combining the modelling concepts in agent-based systems and standard UML models**: Research published in [77] shows the development of a new modelling approach adapts the agent-oriented modelling requirement model, UML component- and UML sequence diagrams into a modelling approach for building DApps to address the problems that are associated with software development in the blockchain domain. This contribution is summarized in Section 4.2 of the thesis.

4. **Development of heuristics-containing standardized modelling elements and their relationships for describing the ontology of the DApp modelling approach**: Research published in [79] shows the development of the conventional elements, element properties, and relationships, of the diagram types in the modelling approach for building DApps. The research

also outlines the standard syntax descriptions for the objects in the developed DApp modelling approach. This contribution is summarized in Section 4.3.

5. **Development of a model-checking method for verifying the syntax correctness for the developed modelling approach for building DApps**: An XML-based model-checking method is developed in [79] for verifying the correctness and consistency of the DApp-design diagrams that apply the DApp modelling approach developed in this thesis. This contribution is summarized in Section 4.3.

6. **Application of the DApp modelling approach in addressing problems that exist in inter-organizational collaborations**: the modelling approach developed in this thesis is applied in two different use-cases of specific IOC settings to show how DApps designed using the developed modelling approach address the trust issues that exist in such collaborations.

   (a) Development of the model requirements for an application that solves problems associated with identity document verification and validation for web-based financial service providers [79]. The developed software is referred to as the IdCredit system. The design of the IdCredit requirement is shown in Section 4.3 of this thesis.

   (b) Development of a blockchain DApp that enables symmetric information sharing in the multi-dimensional collaboration that exists in executing large building construction projects [80]. The designed software is referred to as the construction project management platform (CoPM). The design of the CoPM DApp is summarized in Section 4.4 of this thesis.

7. **Development of a T-DM metamodel representing the semantics of the modelling approach**: The T-DM metamodel shows the heuristics mapping used to realize the diagram types and elements in the T-DM approach from the base requirement model. This is summarized in Section 4.5.

8. **Development of a support tool that enables efficient creation of DApp models using the concepts defined in the developed modelling approach**: A support tool is developed for automating the process of creating DApps with the developed modelling approach. The evaluation of the tool shows there is an increase in efficiency in creating DApp models using the tool

when compared to a free-hand sketch. The evaluation of the developed DApp modelling approach and its support tool are summarized in Section 4.5.

1.3.3 Thesis structure

This thesis consists of six chapters. Chapter 1 provides the main motivations for the thesis and discusses the objectives and research questions addressed in this thesis. Chapter 2 provides the background by outlining the main contexts that form the theoretical foundation for the research conducted in this thesis. The background provided includes an overview of trust management in inter-organizational collaborations, an overview of blockchain applications and the supporting technologies, and an overview of agent-based modelling concepts. Also, the running cases of trust-related problems encountered in different settings of inter-organizational collaborations are presented in the latter part of the chapter. Chapter 3 shows the research methodology outlining the design science research (DSR) method adopted for this thesis. The main pillars of the DSR method are explained and the guidelines of the DSR method are presented showing how they are applied in this thesis. Chapter 4 summarizes the main results of publications used in this thesis and their relevancy towards the objectives and research questions addressed in this thesis. Chapter 5 provides discussions resulting from different research studies conducted in this thesis and their limitations. Chapter 6 presents the conclusion of the thesis by providing answers to the research questions and future work.

2 Theoretical Background and Related Works

This chapter presents the background of the contexts that form the foundation for the work performed in this thesis. Four main contexts that form the background of this thesis include trust management in IOCs, blockchain application development, Agent-Based modelling and UML approaches. The background on trust management between organisations is presented in Section 2.1. The main concepts of trust management between organizations are outlined, common approaches for trust management in IOCs are presented, and the problems that affect these traditional methods of trust management are discussed.

The background on blockchain applications is presented in Section 2.2. This is done by outlining the main concepts of blockchain and the supporting technologies. Background on Agent-Based modelling is presented in Section 2.3. The main concepts of agent modelling are shown together with common agent-based approaches, and the selection of a suitable agent-modelling approach that provides the foundation for the T-DM approach developed in this thesis is discussed. Lastly, background on the UML approaches is presented in Section 2.4 showing the UML models and diagrams.

In addition, two running cases outlining several problems that exist during IOCs are presented in Section 2.5. These running cases are applied in the development of a blockchain DApp using the T-DM approach to address the outlined IOC problems.

2.1 Trust Management in IOCs

This part of the thesis presents the main concepts in enabling trust between organizations, provides common approaches for trust management during IOCs, and outlines the problems that exist in these traditional trust management approaches.

2.1.1 Concepts of trust in IOCs

Connecting systems between organizations for the exchange of information results in several technical- and business-related trust problems [67, 25, 40, 49]. The citation [67] presents the main concepts in trust management between organizations by providing the trust information model, which outlines the factors affecting trust in decision making between organizations. These factors include asset context, the importance of the asset, risks on asset and the reputation of collaborating parties.

- **Asset context**: assets are valuable resources owned by organizations for achieving specific goals [61]. Different classes of assets exist in business organizations, such as physical assets and information assets. Several information assets are generated and exchanged during business collaborations between organizations.

- **Importance-level of an asset**: this factor expresses the business value of an activity or action for a given information asset. It measures the cost and rewards outcome of business activity in reference to a particular business asset.

- **Reputation of collaborating parties**: this factor measures the trustworthiness of parties in business collaborations based on their historical activities. The historical experiences with a particular collaborating party will help in predicting how that party will act in a current collaboration setup. Organizations are more likely to share information assets with credible and reputable organizations within a business collaboration. As a result, the higher the degree of importance an asset is to a given organization, the less likely they are to share it with a less-reputable organization.

- **Risks associated with assets**: Information assets contain important organizational information. As a result, information assets are prone to security risks such as confidentiality, integrity and availability [48]. To maintain trust within a business collaboration, information assets must be tamper-proof, be readily available to other parties within the collaboration, and be confidential so that they are not misused when shared with other parties in the collaboration. Thus, in order to achieve confidentiality, integrity, and availability of information assets in IOCs, the information systems (IS) that support such collaborations need to be *transparent*, *secure* and *interoperable*. Security of IS that enables inter-organizational collaboration ensures that information assets shared are tamper-proof and cannot be manipulated. Transparency ensures that misuse of confidential information assets shared during collaboration is negated by verifying the correctness of such information. Interoperability of IS that supports collaboration ensures that information assets are readily available to the collaborating parties in real-time.

2.1.2 Common approaches of trust management in IOCs

Some of the common approaches and frameworks for trust-management include TrustCoM framework [84], CONOISE-G project [59], trust-to-distrust (T2D) approach [69] and trust based on evidence (TubE) approach [68].

TrustCoM framework: The goal of the TrustCoM framework is a trust system for contract management between organizations designed to provide secure business collaboration and resource sharing within a dynamic environment. The framework involves the creation of virtual organizations with specific reputation scores assigned. These virtual organizations (VO) engage in service discovery and business negotiation with other organizations. Once a negotiation is successful, a service level agreement (SLA) is outlined for the instantiated collaboration.

Virtual organizations have a specific lifecycle that guides their creation and activities. The lifecycle phases of the created VO include the formation, operation, evolution, dissolution and termination phases. The first phase describes the creation and instantiation of a given VO. The second phase specifies the activities engaged by the VO such as bidding for contracts and performing tasks specified in an awarded contract. The third phase specifies the action taken by the VO manager in removing members that fail to perform a given contract correctly and replacing them with another VO. Failing to perform a given task correctly negatively affects the reputation score of the VO. The last phase involves the termination of the VO when a given contract has been completed successfully.
Trust in this type of collaboration set is based solely on the collaboration manager component that regulates the collaboration by rewarding and punishing parties according to the performance of an instantiated contract. The rewards and punishments are achieved by the addition and removal of trust tokens associated with each member of the collaboration.

CONOISE-G: The focus of the CONOISE project is to develop a resilient infrastructure that supports the creation and operations of virtual organizations in a trustable manner. The framework applies grid computing involving a network of virtual computers that participate in the collaboration. Quality of service (QoS) levels are used to assess the behaviour of collaborating VOs. To set up collaboration, service providers advertise the services needed, a requester agent analyses the service requests contained in the advertised service and locates relevant service providers. Bids are submitted and assessed. The bids are assessed by a quality agent and determined based on the trust and QoS level of the service providers.

The operations of VOs that engage in collaborations are managed by a single component that monitors and assesses services based on service requirements specified in the service level agreement. An existing collaboration is destroyed if the rendered service falls below the specified agreement and a new collaboration is initiated by the collaboration manager.

T2D approach: This approach defines a distributed trust calculation that evaluates the trusted relationship between collaborating entities using maximum thresholds, and each party is free to define their threshold values. The Trust to Distrust (2TD) approach presents a three-phase model for setting up trusted business collaborations between organizations.

The first phase involves the initialization of trust, and assignment of trust values to the individual nodes, such that each party performs trust-based evaluations of the other. The ratings are based on each node's experience with respect to the other members. The second phase involves the classification of nodes into their various level of disposition to trust. The third phase involves generating quantitative trust values by collectively assessing information on the trustworthiness of each member based on their ratings by other members in the network.

TuBE approach: This approach seeks to provide a general architecture for managing trust management by implementing a middleware system for reputation management, monitoring of collaboration setups, and reacting to anomalies that may occur within organizational collaborations. Two clear objectives are provided within the TuBE approach: providing useful information for participants to make trust-related decisions and managing the reputations of the organizations.

To achieve the first objective, middleware is used to assess the risks associated with the activities performed by the collaborating parties. The middleware components intercept a trust-related request, and analyse the risks associated with the request. These risks are a set of cost-reward probability scores for information assets affected by the potential activity. For reputation management, the aggregated historical activities of the parties within the collaboration are used to estimate their reputation level. This provides input for a risk assessment of any potential activity performed in the collaboration.

2.1.3 Problems of trust management approaches in IOCs

The methods of trust management for IOCs that are presented have some common similarities. Some of the methods only apply to virtual organizations [84, 59] while others can apply to both virtual organization and real enterprises [69, 68]. They provide techniques for specifying service requirements and a system for managing the reputations of members within the collaboration network. All of the described approaches outline and emphasize a component that manages the collaboration.

The main weakness of the approaches is, therefore, the central component that manages the collaboration. Although these approaches are presented as peer-to-peer (p2p) systems, the collaboration manager component can only be executed from a single node. Thus, business logic for enforcing the collaboration rule is centralized on the information system of the organization it is executed from. Due to this centralization problem, other parties within the collaboration cannot independently verify the business logic executed by the collaboration manager. As a result, the IOCs enabled by these traditional trust management frameworks are insecure and prone to manipulation. This is in line with the study [83] which shows that establishing completely trustable collaborations are nearly impossible due to inequalities between the collaborating parties and the possibility of manipulating collaboration rules by the party that manages the governance of such collaborations.

Furthermore, existing trust management approaches do not fully address the risks of confidentiality of assets generated, and shared, in business collaborations. Since the use (and misuse) of assets in business collaborations cannot be independently verified, such collaboration setups are not transparent. Also, the risk of information asset availability within business collaboration has not been addressed in these trust management approaches. The presented trust management approaches do not in any way enable real-time availability of data between collaborating parties. This results in data interoperability issues. Therefore, the traditional approaches and infrastructure for trust management are insecure, lack transparency and are not interoperable.

2.2 Blockchain Technologies

The main technologies that support the blockchain include, decentralized consensus and storage, public-key cryptography and asymmetric encryption, and smart-contracts. These blockchain-supporting technologies provide the basis for executing trustable business collaborations between organizations [75, 13].

Decentralized consensus: To ensure the correctness and consistency of information added to a blockchain network, blockchain consensus defines the method of updating records in a blockchain network [75]. For an IOC that adopts DApp concepts, a blockchain-consensus system ensures that only correct information validated by the collaborating parties is added to the network. This increases transparency in IOC processes. Proof-of-work (PoW) is the most popular consensus method used by Bitcoin and Ethereum. Bitcoin is the largest blockchain network, while Ethereum is the largest blockchain network for executing smart-contracts [7]. In the PoW consensus method, a difficult mathematical puzzle is presented and the first participant that solves the puzzle is selected to add the next record on a blockchain ledger. Because of scalability issues and resource consumption in PoW, researchers have proposed a proof-of-stake (PoS) consensus method as a viable alternative to PoW. In PoS, participants are selected to add the next record on the ledger based on the amount of stake deposited on the network [36].

Public-key cryptography (PKC): The PKC provides a system for unique identification of participants in a decentralized network using the public- and private-key pair [91]. With the public key, participants in an IOC are identified, and with their private keys, transactions can be signed. The PKC, thereby, provides a tamper-proof system of source verification for any activity in an IOC. With the asymmetric encryption in the PKC, access control can be implemented in IOC processes, ensuring that specific IOC functions can only be executed by certain parties on the network [62]. As a result, for an IOC that adopts DApp concepts, the PKC in blockchains strongly address the security issues currently experienced in traditional systems for executing IOCs.

Decentralized storage: Records stored on a blockchain network are replicated across all the participating nodes [62]. The existing blockchain storage can be extended further using decentralized databases that provide increased repositories for blockchain systems [16].This therefore, provides a way for preserving large information assets that are exchanged in IOC-executed blockchain networks. As a result, for an IOC that adopts the DApps concepts, decentralized storage ensures that data resulting from the execution of IOC functions are accessible in real-time to all the participants. The real-time access to data solves interoperability challenges in the current systems that supports IOC. The interplanetary file system (IPFS) is a Common examples of extended storage systems for blockchain applications[8].

Smart-contracts: These are digitally verifiable, and enforceable, computer programs that run on a blockchain network [18]. Inter-organizational collaborative processes can be reconstructed into smart-contract workflows and executed on blockchain networks [44]. Business logic- and -rules can be coded into a smart contract, thus ensuring that IOCs are executed in a trusted manner without relying on a centralized authority. Blockchain-enabled smart-contracts do not interact with external data; they only rely on information provided within the blockchain system itself for executing business logic. External data gathering components, usually referred to as decentralized oracles, provide the possibility for real-world data inputs to the smart-contracts transparently without relying on centralized parties [28]. This ensures that smart-contracts use trusted data inputs for executing business logic in IOCs.

2.3 Background on Agent-Based Modelling

2.3.1 Concepts in agents-based modelling

Agents are composed of multiple entities interacting with other computer elements and objects within a specified environment [86]. Agents have different types of relationships with other entities and objects. In any given environment in which agents operate, they perform specific tasks or functions for the achievement of common goals. Using the conical view of agents presented in [37], agent-based systems comprised of agents that perform specific tasks in a particular environment, interact with other agents and objects, and establish different types of relationships with the other agents and objects they interact with.

- **Agent environment**: the environment is represented as a distributed system where agents can exist in different organizational backgrounds. Agents perform specified tasks within their organizational domain for the achievement of a common goal of the multi-agent system. Thus, an agent environment is considered as a complex system where various entities interact, exchange information, and perform their specified tasks. An agent environment, therefore, comprises the space of interactions between multiple agents [57].

- **Agent interaction**: agents interact and exchange information with other entities in the multi-agent system. These entities can be physical objects like sensors, computer systems or other similar agents. Agents can have a single relationship with only one entity or multiple relationships with several entities. Agents possess different sets of knowledge about their environment, and therefore, are capable of exchanging useful information with other agents [34].

- **Agent actions**: these are functions or tasks executed by agents in a given environment. The tasks are specific problems that are to be addressed by an agent in a multi-agent system. The functions performed by the agents can be decomposed or refined into several sub-functions that are entirely independent of each other. The level of refinement determines how detailed the modelled system is expected to be [72].

2.3.2 Common agent-based modelling approaches

Agent-based modelling approaches provide techniques for describing complex systems comprising several interacting entities that perform different functions for the achievement of common goals [26]. Several methods exist for agent modelling in software development. Some of the notable approaches include the i* framework [88], Tropos [29], Service-Oriented Goal models [14], Business-Model Alignment with Goal models [5], the Agent-Oriented modelling [73], Prometheus methodology [85], the MaSE methodology [19], the ROADMAP method [39] and INGENIAS [60]. These agent-based approaches are presented as follows.

i* framework: The framework for modelling strategic relationships commonly referred to as the i* provides a method for analyzing relationships and dependencies that exist between actors (agents) performing given tasks (goals) and dependencies between the performed tasks [88].

Tropos: focuses on the requirement analyses in the initial phase of software development and little emphasis is placed on role assignment and on properties of an agent executing a given goal [29].

Service-Oriented Goal modelling:The agent-modelling approach in [14] focuses on separating the agent interactions from the business logic executed by the agents thereby creating two abstraction layers, one for the agent and the other for business process.

Business-Model Alignment Goal modelling: The goal modelling approach described in the citation [5] provides a method for identifying organizational functions that can be realized as an e-service to provide a point of interaction between different organizational systems.

Prometheus methodology: This methodology seeks to provide detailed support for the complete stages of agent-based systems development [85]. Three main stages of software development captured in the method include system specification, architectural design and a detailed design phase. The first phase identifies the interfaces in the designed system and functions performed through the interface. The second phase captures the agents interactions and the environment in which the interaction occurs. The last phase provides detailed internal processes that result in the performance of a given function by the agents.

MaSE methodology: This method of agent-based system development focuses on analyzing the requirements of a proposed system and providing support for the system design. As a result, two stages of software development are used: the analyses phase and the design phase. In the analyses phase, the goals of the system are identified and structured hierarchically. The use-cases which apply to the goals are developed and the roles that perform given goal tasks are outlined. The design phase shows class diagrams that describe the agents in the system, a conversation diagram shows the interaction between agents, agent architecture and deployment diagrams.

The ROADMAP methodology: This method focuses on modelling of complex open systems by eliciting the initial requirements and hierarchical organization of roles of the modelled system. The method also provides a formal model for describing the environment and knowledge of the agents in the modelled system.

Agent-Oriented modelling: The AOM approach developed from the ROADMAP methodology is specifically designed for modelling complex Socio-Technical Systems (STS) comprising several software and non-software agents across different organizational spaces [73]. Software agents are computer programs that perform specific function(s) in a given system. The AOM approach systematically captures in detail functions (goals), executed in a given complex system, refinement of the goals, assignment of actors (agents) to roles, and describes the software property of how a given goal is to be achieved.

INGENIAS: This approach combines the agents-based method with the messaging system outlined in the UML to show the activities and estimate efforts required in a particular project. The method involves the analyses of tasks, system design, model verification, and code generation for multi-agent systems.

2.3.3 Comparative discussions on prominent agent-based modelling approaches

The study [74] provides a method for analyzing agent-based approaches and assesses the prominent agent-modelling approaches such as Tropos, Prometheus and MaSE methodologies. Four main characteristics are used in analyzing these modelling methodologies, they include: concepts, modelling language, process, and pragmatics. The *concepts* analyze the properties of agents captured in the methodology. The *modelling language* analyzes the symbolic (graphical and textual) properties of the language. The *process* analyzes the activities and steps in the modelling language. *Pragmatics* analyses practical organizational issues that may affect the use of a given approach. The result of the analysis shows that the three models examined provide easily understandable notations for representing agents and their concepts. All three modelling languages provide clearly understandable steps that can be replicated by developers in designing agent-based systems. For the pragmatic aspects, it is not clear how many technical or organizational issues are encountered in adopting any of the three modelling languages since they do not provide managerial support.

Applying the same analysis criteria for evaluating agent-based modelling languages to the AOM framework will produce a similar result. The AOM symbols and diagrams are easily understandable objects. The steps in applying AOM are well defined in the framework and easily replicable. Potential technical and managerial issues for adopting AOM in organizations are not clear since they are not covered in the framework. The AOM framework is chosen as the base model for developing the T-DM approach because of the inter-organizational nature of blockchain applications that can be well represented in the AOM framework [78]. This thesis focuses on addressing problems that exist in IOCs. Thus, an IOC is a good representation of STS comprised of both human- and non-human actors across several organizations for the achievement of a given business goal. The AOM provides an uncomplicated technique in assigning roles to specific functions [73]. Also, the software properties associated with any given goal in AOM provides the possibility for integrating novel blockchain concepts in the models. Furthermore, these two doctoral thesis [63, 46] provide overviews of the common agent-modelling approaches and applied AOM concepts in the design of intelligent systems and socio-technical systems respectively. In this thesis, AOM principles are applied in developing a modelling approach for building trustable blockchain DApps.

Considering the amount of work that has been done in developing these prominent agent-based frameworks, This thesis aims to contribute to, and extend, the knowledge-base in the agent-modelling domain. The T-DM approach extends the agent-modelling methodology, particularly AOM, by integrating blockchain properties such as on-chain functions into the agent-based approach. Integrating blockchain properties in system design enables trust among collaborating parties in business operations. Thus, the modelling framework developed in this thesis helps agent-based software designers and developers in building trustable systems that support organizational collaborations.

2.3.4 AOM modelling concepts

The AOM-viewpoint describes the diagram types, elements and relationships that exist in the AOM framework. The viewpoint shows the AOM framework by mapping the abstraction levels to specific viewpoint aspects. The abstraction levels include the conceptual layer, platform-independent design, and the platform-dependent implementation. While the aspects are represented by the modelling types such as interaction, information and behaviour modelling, only the first two layers of the AOM-viewpoint framework are useful for designing a complex system. Therefore, these relevant layers of the AOM-viewpoint framework are presented.

The first abstraction level of the viewpoint is the conceptual domain containing the following models: role model, organizational model, domain model, and goal model. The role and organizational models outline the roles and organizations that exist in the modelled system and specify the interactions between the roles and the organizations. The domain and knowledge model outlines and specifies the classes of information that exist in the modelled system. The goal model specifies the behaviour of the system by outlining the functions, sub-functions and software properties of the functions executed in the modelled system. Roles are assigned to agents such as human- and software agents. Agents are mapped to specific organizations to which they belong. UML diagrams are used in presenting the information contained in the system. Goals represent the executed functions and are refined into sub-goals. Agents are assigned to specific goals which they execute. The software property of a given goal represents the quality goal which describes how a given goal is to be achieved.

2.3.5 Adaption of the AOM for secure software development

The study [2] adapted and extended the AOM concepts in developing a secure method of agile software development called Agile Agent-Oriented Modelling (AAOM). The method involves the identification of security challenges in the goal modelling stage of the AOM process, and mapping the security requirements as user-stories to the traditional agile software development process. The main benefit of AAOM process is that it provides the possibility for identifying security attributes of software functions in the early stages of software development; thereby, ensuring that the resulting output is a secure software artefact.

2.4 Background on UML Diagrams and Concepts

This section of the thesis provides background on the UML approach, the common diagrams, and UML extensions used in realizing custom modelling notations. The UML concepts, diagram types and specifications are contained in the UML user guide [10].The UML provides a general language for describing software systems and also provides standard notations for visualizing modelled systems. The UML diagrams provide concepts outlining both static and dynamic aspects of software systems. The static aspects of a modelled system comprise structural properties for describing the system, and the dynamic aspects describe the behavioural properties of a modelled system.

2.4.1 Common UML diagrams

The UML diagrams are generally organized into two groups: diagrams that describe structural properties of a modelled system, and diagrams that describe the behavioural properties of the system. The structural diagram outlines objects that have to be present in a modelled system. The behavioural diagram outlines the changes that occur in a modelled system. Common structural diagrams include class diagrams and component diagrams. For the behavioural UML diagrams, common examples include activity, use-case and sequence diagrams. These common UML diagrams are described as follows:

- **Class diagram**: this diagram type is used to describe objects within a system, and outlines their attributes and relationships with other objects. The diagram also specifies operations that can be performed on an object.

- **Component diagram**: this diagram type shows the structural composition of a modelled system. It is used to outline the different components that may exist in a complex system and their interfaces for exchanging information with other components.

- **Activity diagram**: this diagram type is used to represent ordered activities and actions within a modelled system. A Control flow object is used to show the direction of a resulting activity from the previous activity.

- **Use-case diagram**: this type of diagram is used to outline all the possible actions of users in a modelled system. Thus, the diagram shows various groups of users and different use-cases linked to each group.

- **Sequence diagram**: this diagram is used to outline time-dependent interactions that occur in the modelled system. The interactions occur between objects and contain messages exchanged between the interacting objects.

2.4.2 Extending UML

The diagram elements and notations of different modelling approaches can be realized and prototyped by extending standard UML diagrams. The UML profile diagram is used in implementing custom diagram notations of different modelling software.

UML profile diagram: This diagram type provides a general approach for customizing UML for applications in a specific domain. As a result, custom notations that are used in various domain-specific languages (DSL) can be implemented by using the UML profile diagram. The UML profile diagrams are composed of stereotypes, tagged values and constraints. Stereotypes are elements used in extending or modifying the properties of any given UML element. Tagged values are used to provide additional information on an extended UML element. Constraints are used to specify conditions that are constant in an extended UML element. The UML profile diagram concepts are applied in this thesis in realizing the T-DM custom modelling notations.

2.5 Running Cases

To properly present and outline the concepts of the T-DM approach in IOCs, two running cases are used in this thesis. The first running case addresses the problem of identity verification and attestation of know your customer (KYC) processes for financial service providers and collaborators. The second running case addresses the problem of information management in the collaborative construction of complex building projects.

Both running cases are used to demonstrate the general applicability of the T-DM approach in multiple organizational domains such as the finance and building sectors. For each of the running cases, a modelled DApp using the T-DM approach is proposed to address the problems presented in the case [6, 55] and the actual implementation of a DApp exists for the first running case [6]. To demonstrate the syntactical correctness, semantic usefulness of the framework, and the effectiveness of the support tool, the focus of the first running case is on identify verification.

2.5.1 Case 1: System for the attestation and authorization of digital identity assets

The first running case of this thesis as elaborated in [6] and published in [79], describes the status quo in know-your-customer (KYC) authentication and the difficulties experienced in the verification and authorization of digital-assets for online financial transactions. These problems are experienced by both the client who wishes to purchase a service and the online merchant who provides the service.

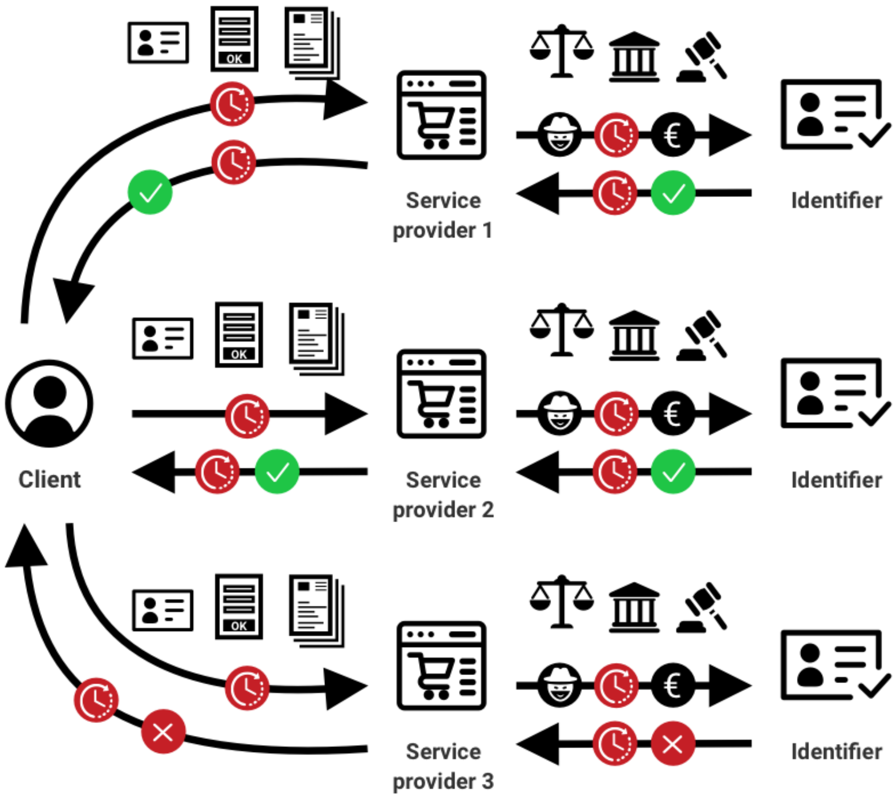


Figure 2.1: Running case about the status quo in identity-document authentication for the financial industry [6].

Figure 2.2: Running case about the problems associated with complex building projects.

When a client wishes to access services from different online merchants, the client is required to submit their personal identification documents to the service provider. The submitted data is then analyzed to determine if they are original and compliant with the institution's requirements. If the client's documents meet the requirements, the transaction can proceed, if not, the transaction fails. The process is repeated for each merchant. This process is time-consuming, and there is no certainty that the client will pass identity verification when the process is iterated over several times by different identifiers. Figure 2.1 shows a pictorial representation of the case described.

2.5.2 Case 2: collaborative construction-project management for transparancy, traceablity and information symmetry

The second running case of this thesis as elaborated in [55] and published in [81], describes the numerous problems that can arise with the construction of complex building projects involving several stakeholders, as witnessed in the construction of the new Berlin Brandenburg international airport. From a technical and management perspective, the problems of the new Berlin Brandenburg international airport were caused by a lack of collaboration-automation, distrust between collaborators and inefficient knowledge management and knowledge sharing among the project participants. Examples of these problems included numerous construction defects, legal conflicts between contractors and project owners, and the cost of the project which increased from roughly 2 thousand million euros, to over 5 thousand million euros at the last count. As a result, the opening date of the airport was delayed by almost 20 years.

The long planning period and the many subcontractors employed, resulted in a high level of complexity. The fact that individual areas and tasks, which would have been better executed by either a single contractor, or several well-coordinated contractors, were spread out over many contractors that neither efficiently coordinated nor efficiently exchanged knowledge and data among each other, also added significantly to the complexity and cost of the project. In addition, numerous and significant changes were made to the size of, and standards set for, the airport while construction was already underway [21, 42]. As a result, it was extremely difficult to administratively manage and coordinate the work of the project. The low level of digitization in the construction industry as a whole also contributed to the chaos on the construction site.

3 Research Methodology

This chapter discusses the research methodology adopted for this thesis and the instantiation of the concepts in the research methodology to the work carried out in this thesis.

3.1 Design Science Research Method

The research conducted in this thesis follows the Design Science Research (DSR) methodology that provides a rigorous framework for the creation and evaluation of information systems (IS) artefacts [33]. The framework provides the methodology for creating new theories, represented as constructs, models, and methods. This thesis focuses on the creation and evaluation of the T-DM approach for designing and developing DApps.

The DSR method, as illustrated in Figure 3.1, has three main pillars: the environment pillar on the left, the knowledge base pillar on the right and the information systems research pillar in the middle. The environment represents the relevance of the research in addressing existing problems in organizations. The knowledge base represents the scientific foundation for providing useful models, methods, frameworks, and validation criteria for the artefacts produced in this research. The IS research in the middle shows the artefacts created in DSR and how they are evaluated [33].

3.2 Instantiation of DSR Research Pillars

Figure 3.1, illustrates the application and instantiation of the DSR methodology adapted from [33] to the work carried out in this thesis. The environment pillar captures the application domain of the thesis, the organizational concept to which the application domain belongs, and the organizational problem addressed. The application domain of this research is the blockchain technology used specifically for designing and developing a blockchain. The organizational concept addressed in this thesis involves inter-organizational business collaborations. Two use-cases relating to trust problems experienced in inter-organizational collaboration are addressed in this thesis. The first use-case involves collaborative processes for identity verification in accessing online-based financial services. The second use-case involves information management in complex building construction projects. The organizational problem addressed is how to develop a modelling approach

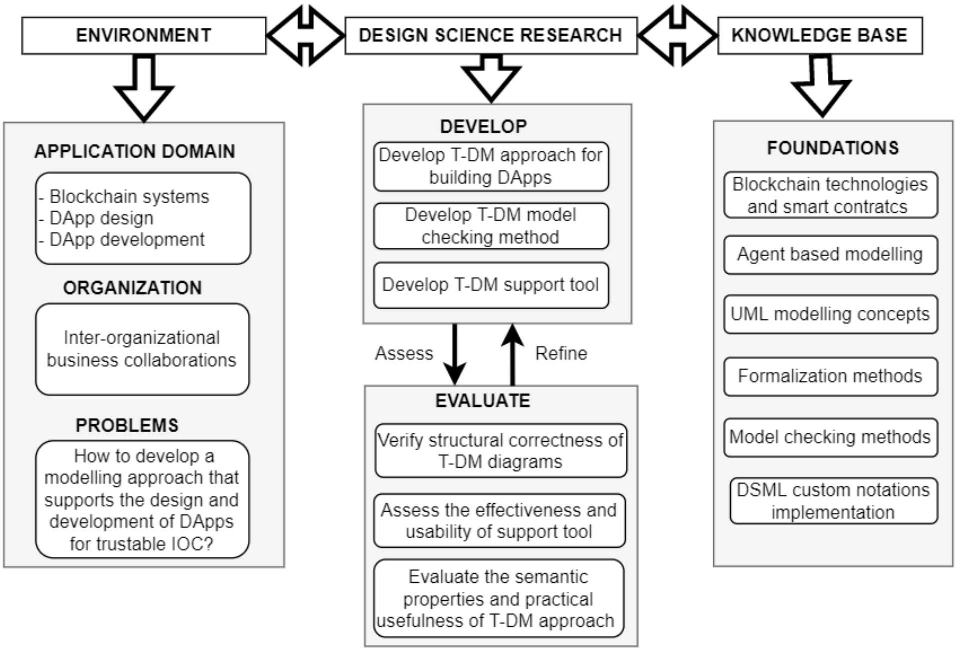


Figure 3.1: Design-Science Research method, adapted from [33].

that supports the development of blockchain applications that enable trustable inter-organizational collaborations between parties in these two use-cases.

The knowledge base provides the scientific foundations for the research works conducted in this thesis. These foundations are presented in research articles that show the main blockchain and smart contracts technologies [75, 13], agent-based modelling concepts [73], UML modelling concepts for implementing architectural and behavioural concepts of software systems [9], XML-based formalization for describing the structural elements and diagrams in a developed modelling approach [15, 58] and lastly, custom notations for implementing tool supports for Domain Specific Modelling Language (DMSL), such as UML profile diagrams[9, 20].

The middle pillar of the DSR framework diagram shows the artefacts developed in this thesis and the evaluations performed to assess and refine these artefacts. Three major artefacts are developed in this thesis. The first, is the T-DM modelling approach for building DApps. The second, is a model-checking method for verifying structural correctness and consistency of T-DM diagrams produced for describing DApp models. The third, is a support tool that enables the efficient creation of DApp diagram models using the T-DM approach. The T-DM approach is developed by extending the AOM and integrating blockchain-relevant concepts [77] [73]. The support tool is developed by extending existing modelling notations for building enterprise systems. To develop the T-DM approach, this thesis first identifies and assesses the existing modelling approaches for DApps. This thesis then addresses the weaknesses in the current modelling approaches in the extended AOM methodologies. The extended AOM that is suitable for building blockchain DApps for IOC processes is represented by the T-DM approach. The model-checking method is developed by applying the XML-DTD to formally outline the elements, diagrams and relationships that exist in the modelling approach. To develop the support tool, standard modelling notations, such as UML elements, are used as the base elements for creating the custom notations for the T-DM custom framework.

As part of the DSR method shown in Figure 3.1, evaluation is necessary for artefacts developed in the research. Thus, the T-DM-approach is applied to the two IOC running cases. The T-DM modelling concepts are used to describe the requirements and architecture of the proposed DApp. As a result, the diagram concepts are modelled using the developed support tool for the T-DM approach. Furthermore, the developed models for building DApps are evaluated for their semantic usefulness in creating DApps for organizations. Also, the support tool is evaluated for effectiveness in creating the correct modelling syntax of the T-DM approach.

3.3 DSR Guidelines and Their Applications

It is necessary to clearly outline and explain how the research conducted in this thesis applies the principles of the DSR method. There are seven guidelines for conducting DSR research and they are as follows: design as an artefact, problem relevance, design evaluation, research contributions, research rigour, design as a search process, and communication of research [33]. Table 3.1 illustrates the seven (7) guidelines for conducting DSR and the description outlining how they are applied in this thesis.

Table 3.1: Guidelines of Design Science Research [33].

Guideline	Description	Application to the thesis
Design as an Artifact	It is essential that a DSR produces artefact(s). These artefacts can be represented as models, methods or frameworks.	For the research conducted in this thesis, three main artefacts are produced. These are: the T-DM approach, model checking method and the support tool. The approach outlines a method that can be used for developing blockchain applications that enable trustable collaborations between organizations. The support tool shows software that can be used to efficiently produce T-DM diagram models
Problem Relevance	DSR focuses on addressing important problems that exist in the business environment.	For this thesis, the relevant organizational problem addressed is the trust issue that exists in IOCs. Thus, addressing the research question: How to develop a modelling approach that supports the design and development of DApps for trustable IOC?
Design Evaluation	DSR proposes that the artefact(s) developed in any research that applies DSR methods must be well evaluated.	In this thesis, the artefacts produced are evaluated. The model checking method is applied in examining the structural correctness and consistency of a DApp requirement model produced in this thesis. The developed T-DM approach for building DApps is assessed to understand its semantic correctness and practical usefulness. The support tool is assessed to understand the efficiency for the correct creation of T-DM models.
Research Contributions	The contributions of well-conducted DSR research must be clear.	Original contributions resulting from this researcher's work have been extensively presented in the initial part of this thesis. The main contribution of this thesis in the area of artefact design is the extension of the AOM framework and UML concepts to develop a modelling approach (T-DM) for building DApps that enable trustable IOCs. Thus the development of T-DM in this thesis is considered a verifiable contribution to the research community.
Research Rigor	This involves the application of rigorous methods in the development and evaluation of artefacts in DSR research.	The rigor in this research is demonstrated in the development of the T-DM approach. Data is collected from related articles and from expert interviews to identify the current methods and their weaknesses to develop the modelling approach. Rigour is also demonstrated in this thesis by adapting existing model checking methods to formalize the syntax properties of the modelling approach. In evaluating the developed modelling approach and support tool, rigour is demonstrated by analyzing various applicable evaluation methods and adapting the most suitable method to evaluate these artefacts.
Design as a Search Process	This guideline shows that effective artefact developed to address specific organizational problems must also satisfy the rules in the problem environment.	Evaluation results from this thesis demonstrates that the developed artefact (T-DM approach) obeys the laws of the applicable domain. The semantic properties of the T-DM are evaluated by experts in the blockchain domain and the result shows that the developed modelling concepts correctly describes the steps required in building DApps.
Communication of Research	This involves effective presentation of the research to audiences in the technology and management field.	The main results of the thesis have resulted in several conferences- and journal publications where audiences of these publications are drawn from experts in technology and management-related fields.

4 Results of Publications

This chapter summarizes the objectives and research contributions of the five (5) publications on which this thesis is based. It also shows the relationship between the main results of the publications to the overall objectives of this thesis. The original publications are included in the appendix of this thesis.

4.1 Publication I: The State of the Art for Blockchain-Enabled Smart-Contract Applications in the Organization

4.1.1 Background and Research objectives

Publication I provides a systematic review of previous studies comprising frameworks, methods, working prototypes and simulations that demonstrate the application of smart-contracts in organizations. Three main research questions provide the guideline for the research conducted in this paper and they are as follows:

- What are the domains of smart-contract applications in established organizations?
- What are the main benefits of smart-contract applications in these organization domains?
- What are the issues limiting the gains of smart-contract usage in the organizations?

Addressing these research questions provides clarity on the different types of organizations where blockchain applications are used, the main reasons blockchain applications are used in organizations, and the major limitations affecting the adoption of blockchain applications in organizations.

4.1.2 Summary of main results

The main results of Publication I are summarized as: i) the current blockchain application domains in organizations, ii) their purposes, and iii) the main limitations affecting the use of blockchain applications in organizations.

DApps application domains and purposes: Application domains represent organizational sectors to which various blockchain DApp use-cases have been applied. The purposes represent the objectives and problems which the DApp use-cases are addressed in the organization. Based on the data collected and analyzed in this paper, the major application areas for blockchain DApps in organizations include supply-chain management (SCM), finance, healthcare, information security, smart city and internet of things (IoT) solutions. The following have been identified as the main motives for applying DApps in organizations: enabling transparency and trust management, data security and privacy, resource management, data tamper-proofing, and data interoperability. Thus, the main purpose for adopting and applying blockchain in organizations is to address trust issues by enabling process transparency, securing assets through tamper-proofing data storage systems, and making data readily available, and interoperable, between organizations.

The use-cases assessed in SCM show how blockchain DApps are used in enabling transparency and trust in IOCs for data that is exchanged among the collaborating parties. The use-cases assessed in finance show that blockchain DApps are used for enabling automated payments for the exchange of values in IOCs; automating collaborative business rules, and enabling financial information security. In healthcare, blockchain DApps are used in enabling patient data security among the collaborating parties providing healthcare services. Also, blockchain DApps are used for auditability and transparency in the use of patient data between healthcare service providers. For the smart-cities and IoT use-cases, blockchain DApps are used to create automated rules for trusted information exchange and for enabling information and data interoperability between smart IoT devices.

Blockchain limitations mitigating DApp adoptions in organizations: The limitations affecting blockchain technology that mitigate the adoption of blockchain DApps in organizations are categorized according to their severity. The significant and critical limitations are those issues that affect most blockchain supporting technologies such as: consensus mechanism, public-key cryptography, decentralized storage and smart-contracts. These blockchain supporting technologies are presented in Section 2.2 of this thesis.

Critical (main) blockchain limitations include: usability and complexity issues, standardization, lack of testing and practical experience, and design architecture issues. Other significant limitations include storage scalability, regulation, the soundness of smart-contracts, security flaws and bugs, privacy leakage, smart-contract life-cycle management and non-tested consensus methods. Less important limitations include: anonymity, scalability-time, transaction cost, cryptocurrency unpredictability, unsustainable consensus methods, trusted third-party involvement and cryptocurrency liquidity problems.

The use of smart-contract applications in organizations is complicated due to blockchain complexity and usability issues. Blockchain networks have complexity and usability challenges, especially for first-time users. Also, blockchain technology has architecture-design issues that are not acceptable to organizational processes. Two main limitations affect blockchain-technology consensus mechanisms: the transaction cost limitation is the cost attached to the execution of transactions on a blockchain (public) network. The second limitation affecting blockchain consensus, is the volatility of cryptocurrencies, which makes long-term economic decision-making difficult for organizations.

The mechanism of a digital signature which is part of the public-key cryptography in the blockchain, has several limitations; these include: anonymity issues, privacy leakage in transactions, and the involvement of a trusted third-party. Anonymity, implies difficulty in linking blockchain identities to real-life individuals, or businesses. Privacy, implies that information recorded on a blockchain (public) network is visible and available to all participants. In addition, in some blockchain DApp use-cases, third parties are involved in recording or updating transactions to the network. The use of smart-contract applications that generate a significant amount of data also creates storage issues. There is a limited amount of data that can be stored uniformly by the participants in the network, and this limitation affects smart-contract applications in processing and storing a significant amount of data. The storage issue is also associated with transaction-cost scalability, because the larger amount of data processed by the network results in higher costs for the initiator of the transactions.

The Solidity programming language is the one most commonly used for coding smart contracts. Solidity has several design issues such as security flaws, soundness and life-cycle management. There are many unknown attack vectors in the ecosystem, and there are also bugs in Solidity code. Additionally, Solidity lacks a formal foundation, and smart-contracts cannot be automatically verified before they are deployed.

4.1.3 Relation to the thesis

The results of Publication I on the application domain analyses for blockchain adoption in organizations, provided justification for the two running cases selected for this thesis. The first running case stems from the finance sector, which is one of the top application domains for DApps. The second running case is represented by the building sector; a domain that has historically lacked proper digitization. The building sector is one of the least digitized business domains, just above the agriculture and hunting industries [1]. The motive for the selected business domains is to show the general applicability of the T-DM approach in sectors that have shown early adoption of digital solutions (Finance) and in sectors which have been slow in the adoption of digital innovation (construction).

The purpose of the two running cases selected for this thesis is to demonstrate the design and development of DApps that enable transparency and data interoperability for collaborating parties in both the finance and the building sector. Thus, the trust issues experienced in the business collaborations presented in these running cases are addressed by blockchain applications that are designed using the modelling approach developed in this thesis. The critical and significant blockchain limitations identified in Publication I, are useful in assessing the strength and weaknesses of existing approaches for building DApps. The result from the assessment of the existing approaches, thus provides the necessary inputs for developing the T-DM approach that addresses the weaknesses of the current approaches for building DApps.

4.2 Publication II: Evaluation of Approaches for Designing and Developing Decentralized Applications on Blockchain

4.2.1 Background and Research objectives

The objective of Publication II is to develop a modelling approach for building blockchain applications that support inter-organizational collaborations. This continues the work done in publication I, which shows that there is significant progress in the application and use of blockchain technology in organizations.

[1]https://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/imagining-constructions-digital-future

There is a growing interest for organizations to move functions that enable trustable inter-organizational collaboration to the blockchain. For organizations, blockchain provides the possibility to secure shared data, provides transparency of processes, enables immutability of information exchanged between organizations, and the automating of rules for business collaborations. To achieve this, the following research questions are developed to provide a guideline for the research conducted in Publication II.

- What are the existing software design- and -development techniques in the blockchain domain?
- What are the main strengths and weaknesses of the identified techniques?
- What is the suitable approach for designing and developing decentralized applications?

In answering the research questions, two methods are used in collecting data related to existing software development techniques for building DApps. They include a review of related literature, and expert interviews. The current limitations that affect the adoption of blockchain applications in organizations provide the criteria for analyses to understand the strength and weaknesses of the identified methods. These blockchain limitations that form the analyses criteria are part of the main findings of publication I, and they include issues such as: usability, transaction cost scalability, privacy leakage, and security flaws. Based on the results of the analyses, a well-aligned software-engineering approach is developed for building blockchain applications.

4.2.2 Summary of main results

The main results from Publication II are summarized as follows: i) the existing methods for developing DApps, ii) the analysis of frameworks, methods and techniques for building DApps and iii) the model-driven T-DM approach for building blockchain DApps for IOC.

The existing methods for developing DApps: Table 4.1 shows the existing frameworks, methods and techniques (FMT) for building blockchain applications collected from related literature and expert interviews. The first column shows the selected framework, method or technique. The second column shows the source, which is either related literature (RL) or expert interviews data (ID). The last column shows a summary description of the selected items.

Analyses of frameworks, methods and techniques for developing DApps: Table 4.2 shows the analyses criteria and the assessment conducted to understand the strength and weaknesses of the analyzed FMTs. The symbols '-', '+', '++' show that the selected item is not applicable, applicable, or very applicable, respectively for each evaluation criteria. The results from Publication II show that although all the selected items (except agile scrum — kanban) are applicable and useful in designing DApps, only the *software-engineering strategy for DApp* framework shows very useful applicability in the design of DApps because of the ability to support detailed requirements-modelling of DApps. For the development phase, only the extended Agile method shows much applicability in this area because it outlines clear steps and tasks in the DApp development.

In the aspects of existing blockchain limitations, the *software-engineering for DApp*, *standard BPMN diagrams* and *freehand sketch activity diagram* all show very useful applicability in addressing blockchain usability and complexity issues because of the clearly defined steps, and easily understandable notations for communicating with project stakeholders. For the security flaws associated with smart-contracts, none of the items provide any applicability in this area, because they do not provide any formal means for verifying the correctness of smart-contract codes before execution on a blockchain. Regarding transaction costs associated with blockchain systems, only the *process-reduction* method shows very useful applicability in addressing this issue. In the problem of privacy of data in blockchain systems, only the *Ancile* method shows potential applicability in addressing this issue.

The model-driven T-DM approach for building blockchain DApps for IOC: The result of the assessment of existing FMTs for building DApps shows that the main blockchain limitations addressable by software-engineering design- and development- approaches are the problems associated with usability and complexity of blockchain technology, and transaction cost issues.

A modelling approach (T-DM) is developed to address the weaknesses in current methods for building DApps. This is achieved by providing a three-phase approach for deriving the requirements, describing the architecture and outlin-

Table 4.1: Existing FMT for building blockchain applications.

Item	Source	Description
Privacy-preserving technique - Ancile	RL	The primary goal of the Ancile project is to address data privacy and security in the management and handling of electronic health records of patients in healthcare systems.
Ontology-driven approach	RL	The project proposes ontology-driven data-modelling techniques for designing the blockchain system for tracking items in a supply chain. The major benefit is in designing blockchain systems that support organizations in understanding data distributed across different networks.
Process reduction method	RL	Business process modelling notation (BPMN) diagram is used to outline the operations that are executed. The derived process is then transformed and reduced into a Petri-net (PN) model using workflow-net semantics.
Software engineering strategy for DApps	RL	The method proposes the use of UML models such as usecase diagrams, class diagrams and activity diagrams to describe the requirement and architecture of a DApp.
Agile + Scrum Kanban	ID	The use of the agile methodology in developing DApps due to its flexibility. The steps range from initiation, planning, execution, and control, to closure of the project. The experts combined the agile method with common task management tools such as scrum and kanban.
Standard UML diagrams	ID	The experts use standard UML diagrams in outlining the static and dynamic properties of the designed DApp.
Standard BPMN diagram	ID	The experts use BPMN to outline the business process of a DApp software that is to be developed.
Freehand sketch activity diagram	ID	The freehand sketch involves the use of hand-drawn activity diagrams to represent different stages of software development.

Table 4.2: Evaluations of FMT for designing and developing DApps.

Item	Source (RL/ID)	Category (F/M/T)	Application Domain	Design support	Development support	Blockchain limitation addressed				Other benefits
						Usability	Security flaws	Cost Scalability	Privacy leakage	
Ancile	RL	M	healthcare	+	+	-	-	-	+	-
Ontology driven modelling	RL	M	Supply chain	+	+	-	-	-	-	data interoperability
Process reduction	RL	M	Supply chain	+	+	-	-	++	-	code generation
Software Engineering for Dapp	RL	F	All	++	+	++	-	-	-	stakeholders communication
Agile + scrum|kanban	ID	M	All	-	++	-	-	-	-	task management
Standard UML diagrams	ID	T	All	+	+	-	-	-	-	technical communication
Standard BPMN diagrams	ID	T	All	+	-	+	-	-	-	stakeholders communication
Free hand sketch activity diagram	ID	T	All	+	-	+	-	-	-	flexibility

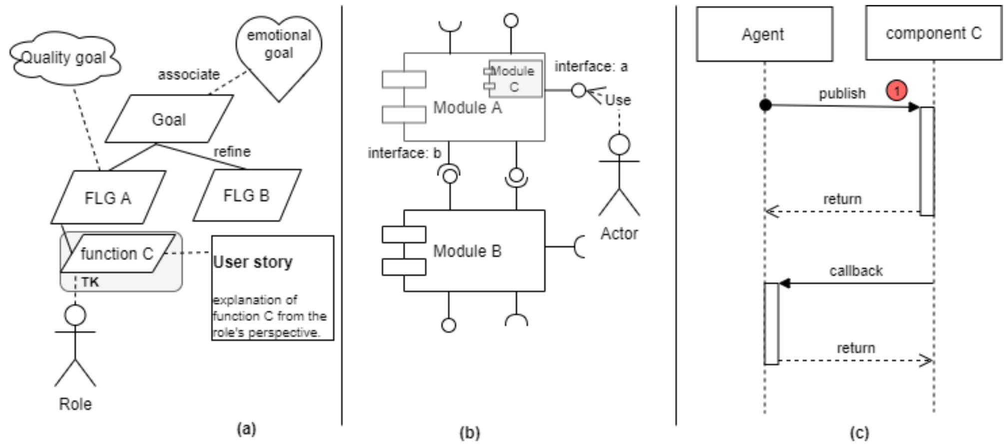


Figure 4.1: Icons of DApp modelling notations (a) T-DM requirement-model notations (b) T-DM static architecture notations (c) T-DM behavioural diagram notations.

ing the behaviour of a designed DApp. The first phase addresses blockchain usability and complexity issues by applying AOM concepts in deriving the initial requirements of the DApp [73, 52, 51]. The AOM is also suitable for designing a complex multi-party system of IOCs that the T-DM seeks to specifically address. The second phase involves the use of UML component diagrams in describing the project architecture for clear communication of project requirements to developers. Finally, the third phase employs the use of UML sequence diagrams to describe the use-cases, and to identify activities that are performed on a blockchain. This addresses the scalability issue by optimizing the number of operations that are executed on a blockchain because only on-chain operations are connected to transaction costs.

4.2.3 Relation to the thesis

The results from Publication II provide inputs for answering the first research question for this thesis which is: How to develop a model-driven (T-DM) approach that supports the design and development of blockchain applications for IOCs? The results show systematic development of the T-DM approach by first understanding the current methods used in building DApps, identifying the weaknesses of the current methods, and developing a modelling approach that addresses those existing weaknesses. Thus, the developed model-driven T-DM approach provides a well-structured approach for capturing the static and dynamic behaviour of blockchain DApps. Furthermore, the findings from the first publication, showing the current issues affecting the use of blockchain in organizations, provide analyses criteria for assessing the strength and weaknesses of the existing methods for building DApps.

4.3 Publication III: Deriving and Formalizing Requirements of Decentralized Applications for Inter-organizational Collaborations

4.3.1 Background and Research objectives

The previous publication, Publication II, shows the three diagram models contained in the T-DM approach and the usefulness of these diagram models in developing blockchain applications. The objective of Publication III is to systematically describe the ontology of the T-DM approach. This is achieved by first specifying the reference requirement sets and heuristics of the T-DM requirement diagram. Then, the ontology of the T-DM approach is developed. The following research questions provide a guideline for the research conducted in this third publication.

- What are the reference requirement sets and stakeholders for inter-organizational collaboration?
- What is the set of heuristics for specifying a T-DM requirement model?
- What is the ontology for creating T-DM support tool?

The T-DM requirement is developed by adapting the AOM Goal Model diagram as already shown in publication II [73]. The T-DM requirement diagram provides the basis for deriving the ontology of the T-DM approach because the rest of the T-DM diagrams are directly mapped from the requirement diagram. This is shown in the meta-model of the T-DM approach presented in publication V, which follows. The ontology of the T-DM approach provides a method for formally describing the elements in the requirements diagrams and for checking the structural correctness of the developed diagram models.

4.3.2 Summary of main results

The main results from publication III include i) The reference requirement sets containing the functional and non-functional requirements, and stakeholders that execute the DApp functions. ii) Graphical and textual descriptions of element syntax of the T-DM requirement. iii) Sample DApp requirement model developed by applying the T-DM reference requirement sets. iv) A model checking method by formally specifying the elements in the T-DM requirement model, and v) Model-check validation results for the sample DApp requirement model developed using the T-DM approach.

Reference requirement sets of T-DM: *Functional requirements* of applications that support organizational collaborations outline specific behaviours of the described software system [53]. The result of the publication shows that a functional requirement of a DApp software can either be executed on the blockchain network (on-chain), or outside of the blockchain network (off-chain). The following are the summary steps for deriving the functional requirements of a DApp. The first step in modelling functional requirements is to identify the main value or problem the system seeks to address, and outline, as the main value proposition. Functional requirements are grouped together if they can be logically categorized as a sub-function of the topmost goal. The first-level requirement represents the topmost goal of requirements that are grouped together. The same approach can be used to further refine the first-level requirements to the second-level requirements, until the *n*th-level requirement. The next step is to associate the on-chain properties, non-functional requirements, and stakeholders, to each goal element. Since the goal models have hierarchical inheritance attributes, this implies that properties assigned to each functional requirement are inherited by the sub-requirements.

Non-functional requirements (NFR) are a set of criteria for assessing the operations of a system [53]. Table 4.3 provides the aggregated list of NFRs relevant for designing blockchain DApps, and also relevant for designing systems for IOCs. From this aggregated list, specific NFRs that are useful for designing blockchain-DApps-based IOC systems, are identified. The first column of the table identifies the specific NFR, the second column shows literature sources, such as IOC systems-related literature and blockchain-DApp-related literature. The third column provides a general explanation of the NFR, and the last column provides a blockchain DAppIOC context for the NFR. For the NFRs where no possible blockchain-DApp and IOC context explanations exist, such an NFR is not considered useful for designing blockchain DApps for IOCs.

Stakeholders are agents that play active roles in executing functions in a system [73]. In a collaborative blockchain system, stakeholders are peers or agents that execute particular functions identified by the functional requirements of a proposed system. The result of Publication III shows that stakeholders in blockchain-based systems can be classified as software agents and human actors. Software agents are smart-contract actors that execute or enforce the conditions outlined in a smart-contract code stored on a blockchain. Software agents may include a web-based API that provides interfaces for interacting with smart-contracts and blockchain data. These autonomous agents could also be represented as oracles to provide data for smart-contracts in a blockchain-based system. Human actors are the participants or parties involved in organizational collaborations. Different roles are assigned to human actors based on the tasks (functions) they perform in a system. Since the functions are project-specific, it is not possible to automatically classify human actors in a collaborative process on a blockchain.

Heuristics for specifying T-DM: *Graphical and textual descriptions of T-DM requirement elements* outlined in Publication III shows that seven (7) elements are used in outlining the requirements of the T-DM approach. They are: i) goal, ii) quality goal, iii) emotional goal, iv) agents, v) decomposition, vi) association, and vii) on-chain. The syntax description of the T-DM requirement is summarized in Table 4.4.

Requirements of a sample DApp, shows the initial requirement of the IdCredit platform is a blockchain DApp that seeks to solve the IOC problems identified in the running case described in Section 2.5.1. The functional, and non-functional requirements, and stakeholders used in describing the IdCredit system are summarized as follows. The main value proposition of the IdCredit system is to provide a decentralized blockchain system that enables the attestation and authorization of digital assets. The value proposition of the system is refined into the first-level goal (FLG) of the system. These are: manage key; onboard user; create a communication channel; store asset information; attest asset; manage transactions, and manage blocks. The quality goals are the non-functional requirement of the DApp that are necessary for producing a system that satisfies the main value proposition of the IdCredit platform. These are as follows: scal-

Table 4.3: Non-functional requirements mapping for blockchain DApps.

Standard NFR	NFR Literature source	Explanation	Blockchain DApp for IOC context
Modifiable	IOC[53]	System changes and adapts during its lifecycle to the business context.	Smart-contract business collaboration lifecycle modifications.
Integrable	IOC[53]	System consists of separately developed and integrated components for which there are interface protocols between the components.	Blockchain DApp for business collaboration consists of separately developed and integrated components for which there are interface protocols between the components
Interoperable	IOC[53], Blockchain-DApp[35, 78, 89]	System must interoperate at runtime with information systems supporting other business functions.	Digital assets stored in the blockchain are accessible in real-time to the collaborating parties.
Secure (tamper-proof, privacy)	IOC[53], Blockchain-DApp[35, 78]	The unauthorized access, or manipulation of digital assets stored in the system.	Digital assets generated from business collaborations stored on a blockchain system are secured from unauthorised access, and free from manipulation.
High automation	IOC[53]	Function that is performed by a software agent.	Function performed by a software agent in a DApp system that supports business collaborations.
Flexible	IOC[53]	The ability of the system to allow diverse collaboration scenarios and permit the inter-organizational harmonization of heterogeneous concepts and technologies.	Not applicable.
Usable	IOC[53]	The learnability, and the easiness of understanding the systems of various user functions.	The learnability and the easiness of understanding the DApp collaborative systems'user functions by the participants.
Scalable	IOC[53], Blockchain-DApp[89]	The system property to grow and manage an increased number of users.	Create and manage an increased number of digital identities represented by users'public/private key pairs.
Applicable	IOC[53]	Specifies for designers the ability to efficiently assess the quality and comprehensiveness of existing systems.	Not applicable.
Feasible	IOC[53]	The possibility of building a system is provable with the existence of application systems.	Not applicable.
Portable	IOC[53]	Ability of a system function to be executed on multiple classes of device types.	Ability of the Dapp collaborative system functions to be executed on multiple classes of device types by the participants.
Performance	IOC[53]	Amount of useful work done by the system.	Useful work measured by the number of transactions recorded on the blockchain DApp for a given time.
Transparent (verifiable)	Blockchain-DApp[35, 78]	System openness and the ability of the participants to view and verify stored records.	The ability of the nodes participating in a business collaboration to verify transactions recorded on the blockchain network.
Trustable	Blockchain-DApp[35, 78]	System ability to behave consistently in expected ways and enforce the rules established by the network.	Ability of a smart-contract to enforce the conditions specified in a smart-contract in business collaborations.

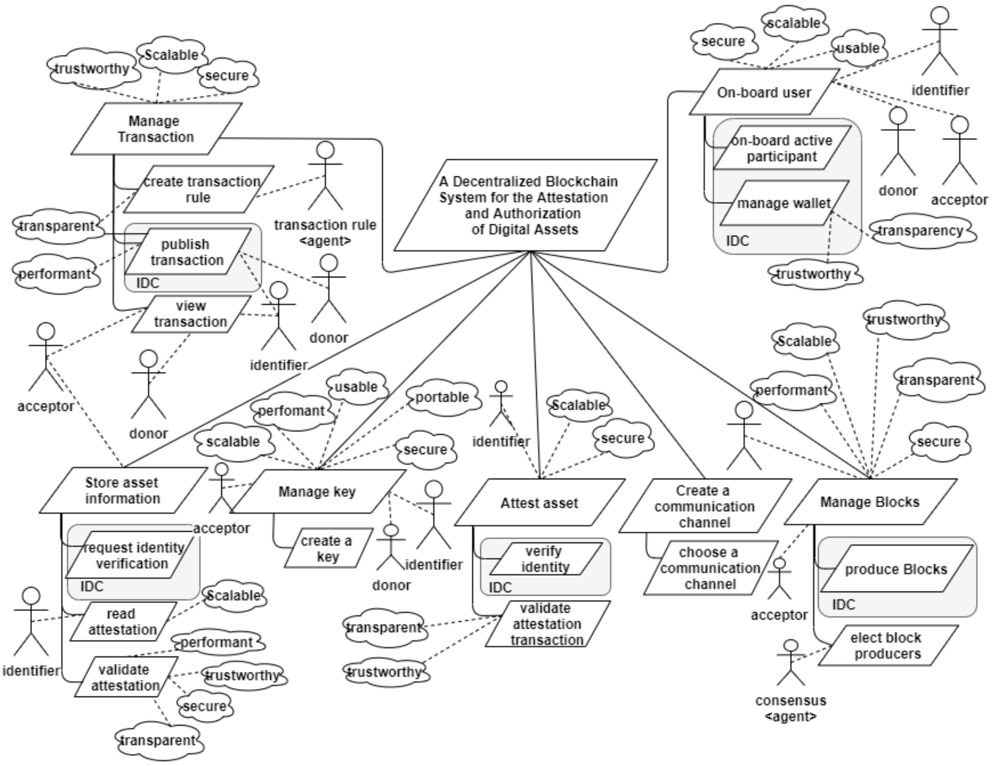


Figure 4.2: IdCredit requirement diagram [6].

Table 4.4: Descriptions of the T-DM graphical notation.

Element	Description	Notation	Properties
Goal	A goal is a condition or state of affairs in the world that the stakeholders would like to achieve		1. Intended action words (or verbs) are used to textually notate a goal. 2. A goal can have sub-goals showing the refinements 3. Additional elements such as quality goal, emotional goals and agents are used in describing the properties of a goal. 4. A parallelogram is used in graphically representing the goal element.
Quality Goal	A quality goal defines specific criteria for achieving a goal.		1. Descriptive terms (adjectives) are used to textually notate a quality goal. 2. Several quality goals can be associated with a goal. 3. Cloud symbol is used in graphically representing a quality goal.
Emotional Goal	An emotional goal defines the feelings of a user towards a particular function or attribute.		1. Descriptive terms showing users feelings are used textually notate an emotional goal. 2. Several emotional goals can be associated to a goal. 3. An emotional goal is represented with a heart symbol.
Agents	An agent is an actor that actively carries out actions to achieve goals by exercising its know-how.		1. Identification words(nouns) are used in textually representing an agent. 2. Several agents can be associated with a goal 3. Two classes of agents exist: Software and human agents. 4. A UML actor symbol is used in graphically representing an agent.
Decomposition	An agent is an actor that actively carries out actions to achieve goals by exercising its know-how.		1. Decomposition shows the relationship between two goals: a parent goal and a sub-goal. 2. A bold straight line is used in graphically representing a decomposition relationship.
Association	An association relationship is used to link (associate) a goal with other elements that helps in describing the goal.		1. Association shows the relationship between a goal and another goal modelling element. 2. A broken straight line is used in graphically representing a decomposition relationship.
On-chain	An on-chain element represents a goal function executed on the blockchain.		1. Descriptive term token represents the digital asset for exchange of value in the decentralized network. 2. The elements are attached to the on-chain goals. 3. A tokenization element is symbolized by curved-edge square.

able; secure; transparent; trustworthy; performant; usable; affordable; portable, and highly automated.

The network stakeholders in the IdCredit system are represented as roles as follows: donor, acceptor and identifier. The donors are the system users that provide personal data for KYC attestation. The acceptors are online merchants that require the donors' personal information to be attested before granting authorization for accessing a certain service, and the identifiers validate and attest the donors'personal data for the acceptors.
Figure 4.2 shows a reduced requirement-model diagram of the IdCredit system

that comprises the root goal, first- and second-level goal refinements.

Ontology of the T-DM approach: The results of publication III show the formal description of the T-DM requirement diagram using the Extensible Markup Language Document Type Definition (XML-DTD) and set theory. The result also shows XML-DTD validation rules for checking the structural correctness of the T-DM diagram.

The *Model checking method* is developed by providing the DTD specification and set property of the constraints that define the modelling elements. This is shown in Table 4.5. Furthermore, syntax validation of the T-DM requirement model is provided by the DTD specification and set property of the constraints that define the relationship elements. This is shown in Table 4.6.

The *Validation of sample DApp* shows the results obtained by applying the model checking method (Table 4.5 and 4.6) to the IdCredit requirement model (Figure 4.2). The model is checked against the following conditions: i) a root goal exists, ii) goals are refined from parent goals, iii) decomposition relationships are used in connecting the goal elements, and iv) association relationships are used in connecting the other elements to the goal elements. To check the model, the IdCredit requirement diagram is transformed into corresponding XML elements. Figures 4.3 and 4.4 show the XML document obtained by transforming the IdCredit DApp requirement model. An XML-path query is used to parse through the generated XML file to return results for which the validation conditions are checked. The model checking results returned 'TRUE' for all the conditions validated.

Table 4.5: T-DM requirement-modelling syntax with element-validation rules.

Element	Contraints	DTD and XPATH specification	Formal definition
Diagram	1. a goal model diagram must have a name. 2. must contain several goal elements. 3. can have non or several agents. 4. can have non or several quality goals. 5. can have non or several emotional goals.	<!ELEMENT diagram (diagram_value,qualitygoal*,emotionalgoal*,agent*, goal+)> <!ELEMENT diagram_value (#PCDATA)> ***diagram elements*** *= XPATH: {'/diagram'}*	*a,b,c,d,e ∈A* *where A = diagram* *a = diagram_value* *b = quality_goal* *c = emotional_goal* *d = agent* *e = goal*
Goal	1. a goal element must have a name and unique identification. 2. must have a refinement level: 0 refinement represents the top level/root goal. 3. can be refined/decomposed from a parent goal. 4. can be associated to quality goals, emotionals and agents.	<!ELEMENT goal (refinement_level,goal_value)> <!ELEMENT refinement_level (#PCDATA)> <!ELEMENT goal_value (#PCDATA)> <!ATTLIST goal goal_id ID #REQUIRED> <!ATTLIST goal decomposed_from IDREFS #IMPLIED> <!ATTLIST goal associated_to IDREFS #IMPLIED> **goal elements** = XPATH: {'//goal'} **verify *root goal exist*** *= XPATH: {'//goal[refinement_level=0]'} ≡ 1*	*a,b ∈A* *where A = goal* *a = refinement_level* *b = goal_value* *A = root goal → a = 0*
Quality Goal	1. a quality goal must have a name and unique identification.	<!ELEMENT qualitygoal (qualitygoal_value)> <!ELEMENT qualitygoal_value (#PCDATA)> <!ATTLIST qualitygoal qualitygoal_id ID #REQUIRED> ***qualitygoal elements*** *= XPATH:{'//qualitygoal'}*	*a ∈A* *where A = qualitygoal* *a = qualitygoal_value*
Emotional goal	1. an emotional goal must have a name and unique identification.	<!ELEMENT emotionalgoal (emotionalgoal_value)> <!ELEMENTemotionalgoal_value (#PCDATA)> <!ATTLIST emotionalgoal emotionalgoal_id ID #REQUIRED> ***emotionalgoal elements*** *= XPATH:{'//emotionalgoal'}*	*a ∈A* *where A = emotionalgoal* *a = emotionalgoal_value*
Agents	1. an agent must have a name and unique identification. 2. can either be a software or human agent .	<!ELEMENT agent (agent_type)> <!ELEMENT agent_type (software_agent|human_agent)> <!ELEMENT software_agent (#PCDATA)> <!ELEMENT human_agent (#PCDATA)> <!ATTLIST agent agent_id ID #REQUIRED> **agent elements** = XPATH:{'//agent'}	*a ∈A* *where A = agent* *a = agent_type* *a = {b ∨c}* *where b = software agent* *c = human agent*

Table 4.6: Syntactic validation for T-DM.

Attributes	Contraints	DTD and XPATH specification	Formal definition
Decomposed _from	1. a decomposition attribute decomposes or refines a goal into sub-goal(s) 2. a decomposition attribute must be connected from goal element to a parent goal element.	<!ATTLIST goal decomposed_from IDREFS #IMPLIED> **verification of parent goals** = XPATH: {//@decomposed_from} ⊆ {/diagram/goal/@goal_id}	*A ⊆B* *where A = decomposed elements id* *B = goals id*
Associated_to	1. an asociation attribute associates or links a goal element to other element types. 2. an association attribute must be connect from a goal to either a quality goal, emotional goal or agent.	<!ATTLIST goal associated_to IDREFS #IMPLIED> **verification of elements associated to goal elements** = XPATH: {//@associated_to} ⊆ {//@qualitygoal_id | //@agent_id | //@emotionalgoal_id }	*A ⊆{B ∨C ∨D}* *where A = associated elements id* *B = qualty goals id* *C = agents id* *D = emotional goals id*

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE diagram [

        <!ELEMENT diagram (diagram_value,qualitygoal*,
        emotionalgoal*, agent*,goal+)>
        <!ELEMENT diagram_value (#PCDATA)>

        <!ELEMENT qualitygoal (qualitygoal_value)>
        <!ELEMENT qualitygoal_value (#PCDATA)>
        <!ATTLIST qualitygoal qualitygoal_id ID #REQUIRED>

        <!ELEMENT emotionalgoal (emotionalgoal_value)>
        <!ELEMENT emotionalgoal_value (#PCDATA)>
        <!ATTLIST emotionalgoal emotionalgoal_id ID #REQUIRED>

        <!ELEMENT agent (agent_type)>
        <!ELEMENT agent_type (software_agent|human_agent)>
        <!ELEMENT software_agent (#PCDATA)>
        <!ELEMENT human_agent (#PCDATA)>
        <!ATTLIST agent agent_id ID #REQUIRED>

        <!ELEMENT goal (refinement_level,goal_value)>
        <!ELEMENT refinement_level (#PCDATA)>
        <!ELEMENT goal_value (#PCDATA)>
        <!ATTLIST goal goal_id ID #REQUIRED>
        <!ATTLIST goal decomposed_from IDREFS #IMPLIED>
        <!ATTLIST goal associated_to IDREFS #IMPLIED>
]>
<diagram>
    <diagram_value>IdCredit value proposition,
    first- and second level goal model</diagram_value>

<!--list of quality goals in the diagram-->
    <qualitygoal qualitygoal_id="q1">
        <qualitygoal_value>trustworthy</qualitygoal_value>
    </qualitygoal>
    <qualitygoal qualitygoal_id="q2">
        <qualitygoal_value>scalable</qualitygoal_value>
    </qualitygoal>
    <qualitygoal qualitygoal_id="q3">
        <qualitygoal_value>secure</qualitygoal_value>
    </qualitygoal>
    <qualitygoal qualitygoal_id="q4">
        <qualitygoal_value>transparent</qualitygoal_value>
    </qualitygoal>
    <qualitygoal qualitygoal_id="q5">
        <qualitygoal_value>performant</qualitygoal_value>
    </qualitygoal>
    <qualitygoal qualitygoal_id="q6">
        <qualitygoal_value>usable</qualitygoal_value>
    </qualitygoal>
    <qualitygoal qualitygoal_id="q7">
        <qualitygoal_value>portable</qualitygoal_value>
    </qualitygoal>

<!--list of agents in the diagram-->
    <agent agent_id="a1">
        <agent_type >
            <software_agent>transaction rule</software_agent>
        </agent_type>
    </agent>
    <agent agent_id="a2">
        <agent_type >
            <human_agent>donor</human_agent>
        </agent_type>
    </agent>
    <agent agent_id="a3">
        <agent_type >
            <human_agent>identifier</human_agent>
        </agent_type>
    </agent>
    <agent agent_id="a4">
        <agent_type >
            <human_agent>acceptor</human_agent>
        </agent_type>
    </agent>
    <agent agent_id="a5">
        <agent_type >
            <software_agent>consensus</software_agent>
        </agent_type>
    </agent>

<!--root goal or value proposition and associations -->
    <goal goal_id="g0">
        <refinement_level>0</refinement_level>
        <goal_value>A Decentralized Blockchain System for the
        Attestation and
        Authorization of Digital Assets</goal_value>
    </goal>

<!--No1 first level refinements and sub goals-->
        <goal goal_id="g1" decomposed_from="g0"
        associated_to="q1 q2 q3">
            <refinement_level>1</refinement_level>
            <goal_value>manage transaction</goal_value>
        </goal>

            <goal goal_id="g1.1" decomposed_from= "g1 g0"
            associated_to="a1 q4">
                <refinement_level>2</refinement_level>
                <goal_value>create transaction rule</goal_value>
            </goal>

            <goal goal_id="g1.2" decomposed_from= "g1 g0"
            associated_to="q4 q5 a2 a3">
                <refinement_level>2</refinement_level>
                <goal_value>publish transaction</goal_value>
            </goal>

            <goal goal_id="g1.3" decomposed_from= "g1 g0"
```


Figure 4.3: IdCredit requirement diagram XML transformation-’a’

```xml
            associated_to="a2 a3 a4">
                <refinement_level>2</refinement_level>
                <goal_value>view transaction</goal_value>
            </goal>

<!--No2 first level refinements and sub goals-->
        <goal goal_id="g2" decomposed_from="g0"
        associated_to="a4">
            <refinement_level>1</refinement_level>
            <goal_value>store asset information</goal_value>
        </goal>

            <goal goal_id="g2.1" decomposed_from= "g2 g0">
                <refinement_level>2</refinement_level>
                <goal_value>request identity
                verification</goal_value>
            </goal>

            <goal goal_id="g2.2" decomposed_from= "g2 g0"
            associated_to="q2 a3">
                <refinement_level>2</refinement_level>
                <goal_value>read attestation</goal_value>
            </goal>

            <goal goal_id="g2.3" decomposed_from= "g2 g0"
            associated_to="q1 q3 q4 q5">
                <refinement_level>2</refinement_level>
                <goal_value>validate attestation</goal_value>
            </goal>

<!--No3 first level refinements and sub goals-->
        <goal goal_id="g3" decomposed_from="g0"
        associated_to="q2 q3 q5 q6 q7 a2 a3 a4">
            <refinement_level>1</refinement_level>
            <goal_value>manage key</goal_value>
        </goal>

            <goal goal_id="g3.1" decomposed_from= "g3 g0">
                <refinement_level>2</refinement_level>
                <goal_value>create key</goal_value>
            </goal>

<!--No4 first level refinements and sub goals-->
        <goal goal_id="g4" decomposed_from="g0"
        associated_to="q2 q3 a3">
            <refinement_level>1</refinement_level>
            <goal_value>attest asset</goal_value>
        </goal>

            <goal goal_id="g4.1" decomposed_from= "g4 g0">
                <refinement_level>2</refinement_level>
                <goal_value>verify identity</goal_value>
            </goal>
            <goal goal_id="g4.2" decomposed_from= "g4 g0"
            associated_to="q1 q4">
                <refinement_level>2</refinement_level>
                <goal_value>validate attestation
                transaction</goal_value>
            </goal>

<!--No5 first level refinements and sub goals-->
        <goal goal_id="g5" decomposed_from="g0">
            <refinement_level>1</refinement_level>
            <goal_value>create a communication
            channel</goal_value>
        </goal>

            <goal goal_id="g5.1" decomposed_from= "g5 g0">
                <refinement_level>2</refinement_level>
                <goal_value>choose a communication
                channel</goal_value>
            </goal>

<!--No6 first level refinements and sub goals-->
        <goal goal_id="g6" decomposed_from="g0"
        associated_to="q1 q2 q3 q4 q5 a3 a4">
            <refinement_level>1</refinement_level>
            <goal_value>manage blocks</goal_value>
        </goal>

            <goal goal_id="g6.1" decomposed_from= "g6 g0">
                <refinement_level>2</refinement_level>
                <goal_value>produce blocks</goal_value>
            </goal>

            <goal goal_id="g6.2" decomposed_from= "g6 g0"
            associated_to="a5">
                <refinement_level>2</refinement_level>
                <goal_value>elect block producers</goal_value>
            </goal>

<!--No7 first level refinements and sub goals-->
        <goal goal_id="g7" decomposed_from="g0"
        associated_to="q2 q3 q6 a2 a3 a4">
            <refinement_level>1</refinement_level>
            <goal_value>onboard user</goal_value>
        </goal>

            <goal goal_id="g7.1" decomposed_from= "g7 g0">
                <refinement_level>2</refinement_level>
                <goal_value>onboard active
                participant</goal_value>
            </goal>

            <goal goal_id="g7.2" decomposed_from= "g7 g0"
            associated_to="q1 q4">
                <refinement_level>2</refinement_level>
                <goal_value>manage wallet</goal_value>
            </goal>
</diagram>
```


Figure 4.4: IdCredit requirement diagram XML transformation-’b’

4.3.3 Relation to the thesis

The research questions addressed in Publication III provided the reference requirement sets, graphical heuristics and model formalization for specifying the ontology of the T-DM approach. The results generated provided the input showing how to systematically develop the ontology of the T-DM approach that enable trustable inter-organizational collaboration. Thus, the outputs of the third publication which include: functional requirements for DApp; non-functional requirements for IOCs; stakeholders in blockchain DApps; graphical and textual descriptions of T-DM requirement elements; requirements of a sample DApp; model checking method, and validation of sample DApp, provide the answer to the third research question for this thesis.

4.4 Publication IV: Designing a Collaborative Construction-Project Platform on Blockchain Technology for Information Symmetry

4.4.1 Background and Research objectives

The objective of Publication IV is to develop a blockchain DApp that addresses the trust-related issues in complex building construction projects by applying the T-DM concepts outlined in Publications II and III. The blockchain application developed is referred to as the CoPM, which is a construction project management automation system that provides the actual requirements for achieving high-quality construction-project collaborations, without the fractures and frictions, that result in unnecessary time delays and cost increases. The following research questions provided a guideline for the research conducted in Publication IV.

- What are the sets of requirements assigned to the diverse stakeholders who collaborate?
- What is the architecture topology of a requirements-satisfying construction-collaboration system?
- What are the dynamic exchange protocols between stakeholders and system components into which embedded on-chain transactions assure immutable traceability?

The requirement sets of CoPM DApp comprise functional and non-functional requirements, and the stakeholders that execute these functions, as outlined in the

T-DM requirement diagram. The architecture topology of the CoPM shows the main components and sub-components, and interfaces for information exchange between the CoPM components as outlined in the T-DM static architecture diagram. The dynamic exchange protocols show the use-cases of on-chain events that occur in different operations in the DApp application as outlined in the T-DM behavioural diagram.

4.4.2 Summary of main results

The main results from Publication IV are summarized as follows: i) CoPM requirements, ii) architecture and iii) platform use-cases.

CoPM requirements: *CoPM Non-functional Requirements* are software properties that show how the selected functions in the system are executed. The following are identified as the relevant NFR for designing the CoPM platform: usable; portable; scalable; secure; transparent; trustable; performant, and scalable. The NFRs are each connected to the applicable functional requirement of the CoPM.

CoPM Functional Requirements are specified by the main value proposition and functions executed on the platform. The main value propositions of the platform are as follows: i) *onboard new user*, ii) *manage digital asset*, iii) *onboard new project*, iv)*manage project*, v) *manage payment* and vi)*manage wallet*. The *onboard new user* function enables the execution of KYC-related goals and assignment of roles to the users. The *manage digital asset* function enables the users to create digital assets such as building information modelling (BIM) objects and models, and manage the ownership and usage of such assets. The *onboard new project* function enables users to initiate construction projects, manage technical- and business- requirements associated with the project, manage the tender process, and assign roles to relevant stakeholders. The *manage project* function addresses the possibility to manage an ongoing, or already completed, project. The *manage payment* function provides the possibility of tracking milestones associated with a project, and processing payments under clearly stated conditions. The *manage wallet* function caters for allowing all involved stakeholders to manage their different token types, and also for transfer between users of the platform. The on-chain functions contained in each value proposition are captured in their further refinements.

CoPM Stakeholders are platform participants and software agents that execute different functions. The classes of users identified for CoPM are the *designers*, *contractors*, *auditors*, *project owners* (project managers), original equipment manufacturers *(OEM)*, *admin*, and *software agents*. The designers are the architects, planners, engineers, and specifiers that perform goals that result in the creation of digital assets on the CoPM. Auditors are certified individuals who perform checks to verify that construction work meets the required-, or stated standards. The project owners and managers represent the individuals responsible for coordinating the building projects. The OEM produces physical building objects used by contractors. The admins are the platform administrators responsible for onboarding new users. The software agents represent the autonomous and semi-autonomous agents. They include *IoT agent*, *KCY agent* and *auditor agent*. The IoT agent records and transmits the status of building projects. The KYC agents perform verification of identity and other documents submitted by new users. The auditor agent, together with the human auditor, verifies the status of building projects in a semi-autonomous way.

CoPM Architecture: The CoPM architecture comprises the main components, sub-components and interfaces, for information exchange between components. The main components and sub-components are directly mapped from CoPM value propositions and refined functions.

CoPM Components and Sub-components include user manager; project onboarder; completed project manager; digital asset manager; payment manager, and wallet manager.

The user manager component is for onboarding new users and has the following sub-components: KYC; Key manager; and Role manager.

The digital asset manager component enables the creation and exchange of digital assets, such as building objects and models, on the platform. It has the following sub-components: building object creator, building model creator and digital asset publisher. The project onboarder component enables the project owner to initiate a new project by outlining the technical requirements and business requirements of the project. It has the following sub-components: requirement logger, and tender manager. The completed project manager component enables the project owner to maintain existing building projects and update the requirements of an ongoing building project when necessary. The project manager has the sub-components: maintain the project container, and requirement updater.

The Payment manager component enables the platform to automatically monitor ongoing projects and sends payments to the contractors and designers when both technical- and business- criteria of the project are satisfied. The payment manager has two sub-components: the tracking logger, and the milestone logger. The wallet manager component provides interfaces for other components to access tokens and exchange tokens with other users. The wallet manager also allows users to access and view their token balances. The wallet manager has the sub-components: token manager, and token history component.

CoPM Components interfaces enable data exchanges between different components of the platform. The user manager has two interfaces: the first for KYC data verification and the second for the assignment of public/ private key pairs to new users. The project onboarder has interfaces communicating project requirements with the completed project manager and digital asset manager. The project onboarder also has another interface for communicating project milestones with the payment manager. The payment manager also has an interface for communicating project status with the wallet manager.

CoPM platform use-cases: Four different operations are used to demonstrate the behaviour of the CoPM platform; they include: digital asset creation operations; tender initiation operations; project payment operations, and building maintenance operations. The events that result in a status change of the blockchain are also captured and sequentially numbered.

Digital asset-creation operations show the set of activities and information exchanges necessary for creating digital assets such as: building objects (BO) and building models (BM) by designers, such as OEMs and architects.

Tender-initiation operations outline the sequential activities required for setting up a project tender, and identify project stakeholders that are assigned different roles during the execution of the building project.

Project-payment operations show a set of sequential activities that result in payment when a particular project milestone is reached or when the entire project has been completed.

Building-maintenance operations show the activities necessary for monitoring completed projects and initiating a tender process for building-object repairs, or replacements.

4.4.3 Relation to the thesis

Publication IV presents the design of the CoPM platform by creating the DApp models. The CoPM DApp models consisting of the requirement, static architecture and behaviour diagrams show a sample blockchain application that enables trustable IOCs developed by applying the concepts defined in the T-DM approach (outlined in Publications II and III). Thus, the results from Publication IV provide the answer to the third research question for this thesis on how to develop a blockchain application for trustable organizational collaborations using the T-DM approach. Based on the design of the CoPM, trust is achieved in complex building construction projects by clearly specifying the requirements of a building project, the tasks of the collaborating parties, and using smart-contract enabled software agents to automate information exchange between parties in building construction.

4.5 Publication V: Implementation and Evaluation of the DAOM Framework and Support Tool for Designing Blockchain Applications

4.5.1 Background and Research objectives

The objective of Publication V is to implement the T-DM approach and support tool, and further provide evaluations for the modelling approach and the support tool separately. Publications II and III show the development of the T-DM approach and the ontology descriptions. Publication V provides a single representation of the modelling approach and the relationships of constituting elements, and shows the implementation of the support tool. The following research questions provide a guideline for the research conducted in Publication V.

- What is the model representation of the T-DM showing relationships between all modelling elements?
- What are the modelling constructs for implementing the T-DM diagram models?
- What is the usefulness of the T-DM framework in designing DApps, and the effectiveness of the tool support in producing T-DM diagram models?

To show a single implementation or representation of the T-DM approach, a metamodel is used to capture all the diagram types, elements and relationships in the

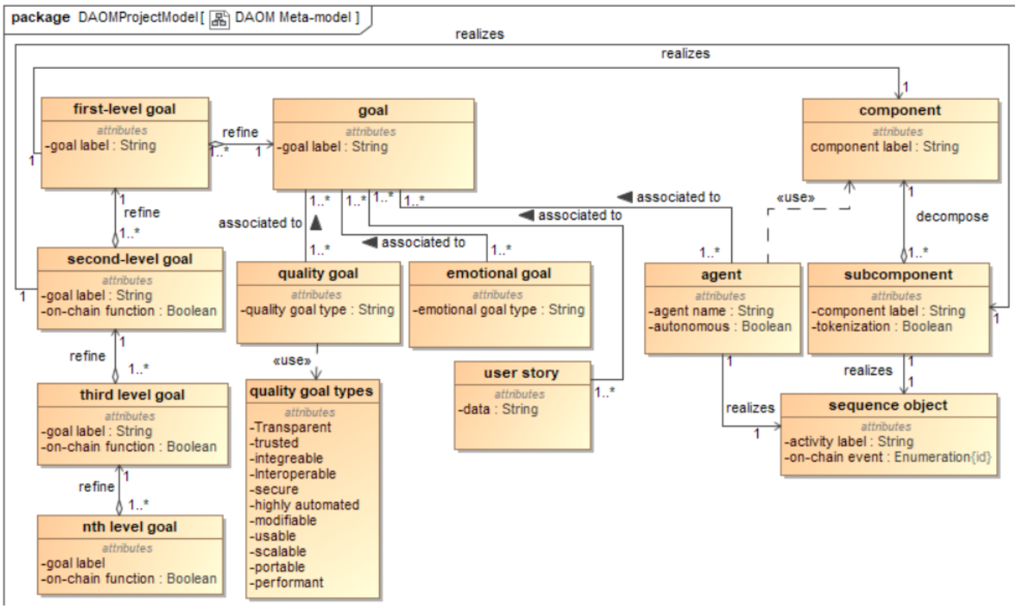


Figure 4.5: T-DM meta-model class diagram.

T-DM approach. Thus, the developed T-DM meta-model provides the semantic description of the modelling approach. To implement the support tool, a UML profile is used to extend the standard UML notations to realize the T-DM custom elements. The T-DM approach is evaluated by examining the semantic correctness of the approach and the practical usefulness of the modelling approach in organizations for building blockchain applications. The support tool is evaluated by examining its usability and efficiency in reproducing correct T-DM DApp models.

4.5.2 Summary of main results

The main results from Publication V are summarized by providing the metamodel of the T-DM approach, the UML profile diagrams used in implementing the support tool, and the evaluation results for the modelling approach and support tool respectively.

T-DM metamodel: The metamodel of the T-DM is implemented using UML class diagrams showing the heuristic mapping used in deriving the diagram types and elements in a T-DM model from the base-diagram elements. Figure 4.5 shows

Table 4.7: Mapping of UML elements to T-DM elements, adapted from [9].

Type	UML base elements	Detail	Extended	T-DM element	T-DM Diagram Type
element	Information Item	represents abstraction of information that can be exchanged between elements.	Yes	Goal	Requirement
			Yes	Quality goal	Requirement
			Yes	Emotional goal	Requirement
	Artifact	an information produced or used by a system.	Yes	On-chain goal	Requirement
			Yes	On-chain transaction	Use-case
	Actor	a use that interacts with a system.	Yes	Agent	Requirement
			Yes	Agent	Static architecture
	Comment	information note about the associated element.	Yes	User-story	Requirement
	Component	a modular part of a given system.	No	Component	Static architecture
			Yes	Tokenized component	Static architecture
	Interface	support point for interaction in a system.	No	Interface	Static architecture
	Sequence object/lifeline	a connectable that participates in a system interaction.	No	Sequence object	Use-case
relationship	Abstraction	level of relationship between elements.	Yes	Decomposition	Requirement
	Dependency	depent relationship between elements.	Yes	Association	Requirement
	Interface realization	realization of a given interface as output.	No	Interface realization	Static architecture
	Interface usage	usage of a given interface as input.	No	Interface usage	Static architecture
	Send message	depicts a given sequential activity.	No	Send message	Use-case
	Reply message	return value of a given sequential activity.	No	Reply message	Use-case

the T-DM meta-model.

The summary of the heuristic mapping and elements relationships in the T-DM approach used in realizing the metamodel is presented as follows. The goal elements are refined into sub-goals, up to an *n*th-level goal. The level of the requirements captures the depth of the requirement details described for a DApp. The quality goals, emotional goals, agents, and user stories are associated to the goal elements. The quality goal is defined by the element's attributes in the quality-goal-type class. The quality-goal types include: transparency; trusted; integrable; interoperable; secure highly automated; modifiable; usable; scalable; portable, and performant. The goal attribute on-chain function has a boolean value that determines if a given goal is executed on a blockchain. The agent has an attribute autonomous with a boolean value that determines if a given goal is executed by a software-, or a human agent. The first-level goal refinements are used in realizing the main components of the static architecture, while the second-level goal refinements realize sub-components. An agent uses the components it is associated with. The sub-components have attribute tokenization that shows if an agent interaction with a given component results in exchanging digital assets (tokens). A respective sub-component and agent realize the sequence objects of the use-case description.

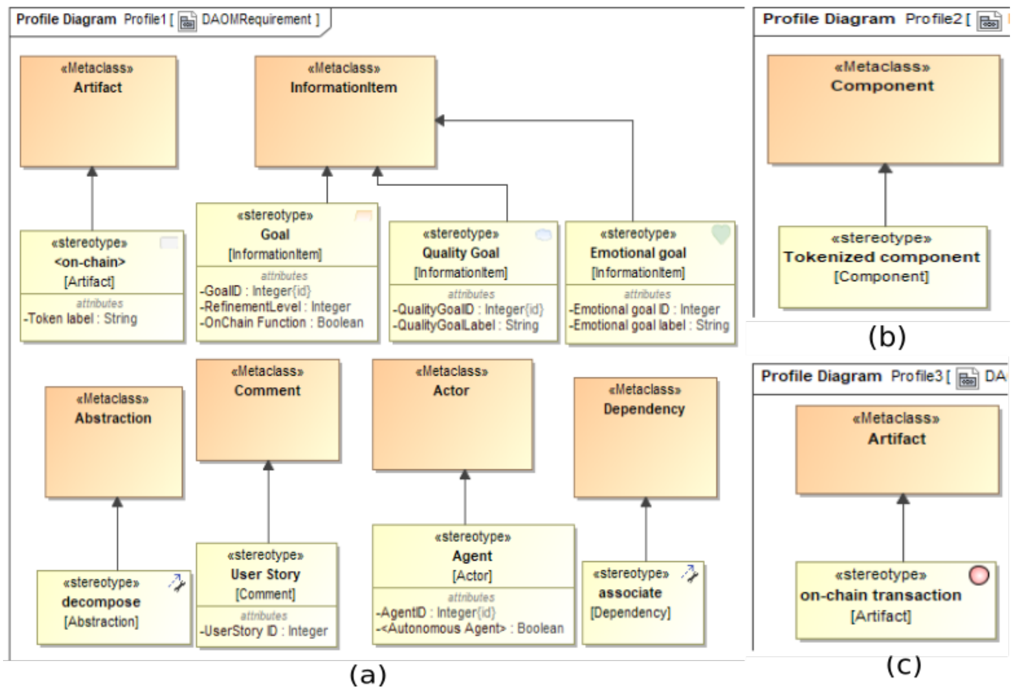


Figure 4.6: T-DM profile diagrams: a) T-DM requirement UML profile. b) T-DM static architecture UML component-diagram extension profile. c) T-DM use-case UML sequence-diagram extension profile.

T-DM support tool profile diagrams: To implement the T-DM tool support, UML base elements are first mapped to specific T-DM model elements. Table 4.7 shows the mapping of UML base elements to T-DM modelling elements and diagram types. The UML base elements are then extended using UML profile diagram properties such as stereotypes, and attributes to realize the T-DM custom diagram notations.

Figure 4.6(a) shows a UML profile of the T-DM requirement diagram. The *Information item* meta-class is extended to the stereotypes, quality and emotional goal. The goal-stereotype comprises attributes for the *refinement level* and *on-chain function*. The actor metaclass is extended by the agent stereotype with the attribute *autonomous agent*. The comment metaclass is extended by the user-story stereotype. The artefact metaclass is extended by the on-chain stereotype with the attribute *token label*. The relationship metaclass 'abstraction' and 'dependency' are extended by stereotypes to decompose and associate respectively. For all the meta-classes, the visual representation of each T-DM requirement is captured in the stereotype icon attribute.

T-DM evaluation results: *T-DM modelling approach evaluation results* are generated from a webinar event that is conducted involving domain experts in the blockchain field and the T-DM approach is presented. The experts are drawn from different industry backgrounds such as supply chain, healthcare, finance, education and research. The webinar has a total of 15 participants. During the workshop, all the major concepts of the T-DM approach such as the elements, relationships and diagram types are presented. Furthermore, the running case of a DApp presented in this paper is introduced to the participants. At the end of the webinar, the semantic and pragmatic qualities (practical usefulness) of the T-DM approach are captured as feedback from the participants. The result of the evaluation is shown in Figure 4.7.

The average score recorded by the experts in assessing the realisticity, completeness, relevance and correctness of DApp models produced using the T-DM approach is ranked 86% (4.3/5). Also, the average score recorded by the experts in assessing the usefulness of the T-DM approach in increasing performance, productivity, quality etc. when applied in organizations for developing blockchain applications is ranked 80% (4.0/5). Thus, the result of this evaluation shows that blockchain domain experts agree that the T-DM approach is semantically correct and pragmatically useful in designing and developing DApps.

T-DM support tool evaluation results are generated from a workshop conducted involving users who are not experts in the blockchain domain but possess the capacity to easily grasp the concepts of blockchain-DApp design. These users are drawn from master's students in the related IT departments; a total of 7 participants involving master's students actively conducting thesis in blockchain-related topics. Using a scale of 1 to 5, the feedback of the students is collected and analyzed to understand the effectiveness of the T-DM support tool in comparison with a standard freehand drawing tool.

The results in Figure 4.8 show that all the students participating in the workshop have relatively the same understanding of T-DM concepts and the running case presented. Yet, the effort required to re-create the modelling task and the ease of using the drawing tools differs among the students. The students who complete the design task using the support tool find the effort required in re-creating the T-DM requirement diagram relatively low, unlike the students who use a freehand

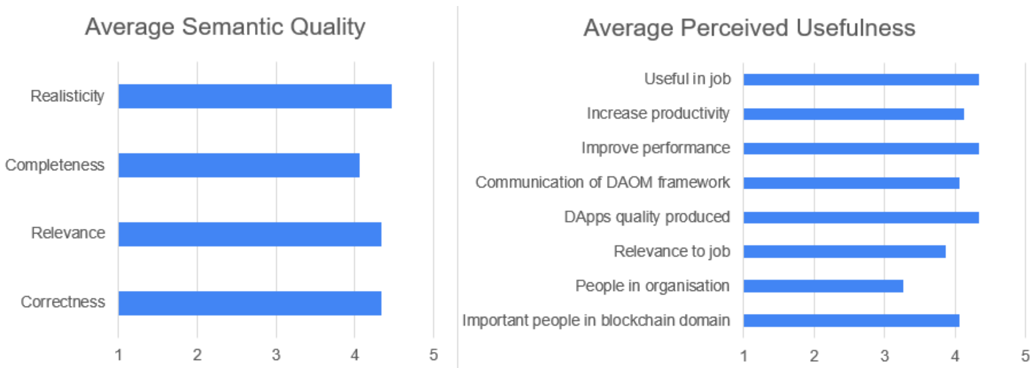


Figure 4.7: T-DM approach evaluation result.

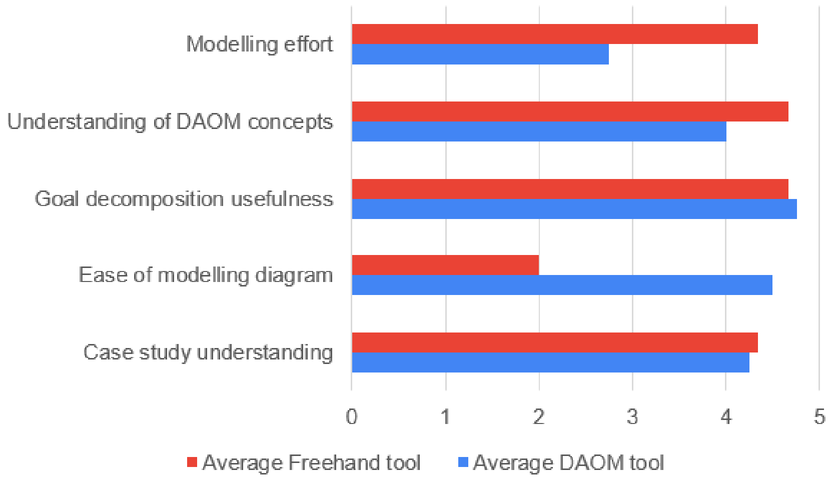


Figure 4.8: T-DM support tool evaluation result.

drawing tool that requires a high effort in completing the modelling task. The ease of modelling the diagram is high for the students who use the support tool. For the students who use the free-hand tool, the modelling task is reported as being difficult to complete. The results confirm that the automations' available in the T-DM support tool make designing T-DM DApp models easy with little effort.

4.5.3 Relation to the thesis

Publication V provides the implementation of the model representation of the T-DM approach, implementation of the support tool, and evaluation of the modelling approach and the support tool. The model representation of the T-DM and the support tool builds on previous works. Publication II and III, describe the developments of the modelling approach and the ontology. Thus, the results generated from this Publication V such as: T-DM metamodel (for modelling approach representation), T-DM profile diagrams (for support tool implementation), and the evaluation results, provide the answers to the fourth research question of this thesis by outlining how to implement, and evaluate, the T-DM approach and support tool for designing blockchain applications.

5 Discussions

In this chapter of the thesis, the main contributions are analyzed in relation to how they extend earlier scientific works. The contributions are the results that provide answers to the four research questions developed for this thesis.

5.1 Implications From the Development of the T-DM approach

The first research question of this thesis is: *How to develop a model-driven (T-DM) approach that supports the design and development of blockchain applications for IOCs?* The results from Publications I and II presented in this thesis provide the systematic development of the T-DM approach, and the main result is the graphical presentation of the T-DM approach describing the diagram types and how it addresses the gaps in existing methods of building DApps. The development of the T-DM approach extends earlier scientific works [66, 73, 76, 65].

The article [66], provides a framework for understanding research direction and opportunities in the blockchain information systems research domain. The research framework is developed by organizing the key concerns in blockchain research into a research agenda that provides for a better understanding of interactions between blockchain applications and supporting technological protocols. Two main research levels are identified in the blockchain IS system; they are the protocol level and the application level. Consensus mechanisms are the foundational protocol of blockchains, because they provide a governance mechanism for applications built on blockchain. The application layer focuses on software programs built on top of blockchain protocols that address business and societal problems. The T-DM modelling approach developed in this thesis falls into the application level of blockchain research. This is because the T-DM approach focuses on developing blockchain applications that address business problems by enabling trustable inter-organizational collaborations. Furthermore, the DApp models produced by the T-DM approach are blockchain protocol technology agnostic because the models can be implemented in any blockchain networks that support smart-contract program executions.

The modelling approach represented by T-DM is achieved by extending the AOM and UML modelling constructs. AOM provides simplicity in describing the requirements of complex systems, which can also be applied to complex IOCs; typically the case types focused on in this thesis [73]. Similar studies have investigated the application of agent-modelling techniques in deriving the requirements of software applications. The study [76] applies case-study research in developing an AOM modelling method to support agile software development processes. The main contribution of the study is the description of a visualized approach suitable for agile requirements engineering. The study improves on the existing body of knowledge in AOM by providing a method for mapping the requirements contained in the AOM goal- model diagram to agile software development tasks by using user-stories. Still, the study [76], does not capture the architectural aspects of software design. The modelling concepts defined in the DApp modelling framework presented in this thesis have already been applied in developing blockchain systems in organizations [54] and [65]. However, actual enterprise modelling tools to support modelling with this framework are still lacking. This thesis is, therefore, necessary to provide a scientific foundation for the framework, and provide a clear explanation of, the concepts before the implementation of the DApp modelling tool.

5.2 Implications From the Development of T-DM Ontology

The second research question of this thesis is: *How to develop the ontology of the T-DM approach for building DApps?*. The results from Publication III outline the ontology of the T-DM approach. This ontology is presented by syntax (graphical and textual) descriptions of elements and diagram types in the modelling approach. The syntax of the T-DM approach is formalized by using XML-DTD to represent the modelling elements and relations; thus, ensuring structural correctness and consistency of T-DM diagrams. The main artefacts produced in the development of the T-DM ontology are the T-formal description of the T-DM requirement diagram, and diagram validation rules. The development of the T-DM ontology extends earlier scientific works [45, 4].

The study [45] describes a method for developing AOM goal-model diagrams. The result shows a modelling tool that can be accessed from Agent Oriented Modelling for Socio-Technical Systems (AOM4STS)[2]. This tool provides modelling elements and connections for building a goal-model diagram. Three connection types ensure that the goal element is correctly connected with other modelling elements. Still, it is not possible to verify the correctness of a goal-model diagram with the tool because it lacks the means to verify whether the diagram contains

[2]AOM4STS http://www.tud.ttu.ee/im/Msury.Mahunnah/AOM4STS/

the correct combination of elements. However, The correctness of element combinations in a goal diagram can be verified through the DTD specification of the diagram elements as presented in this thesis. The tool from the study [45], has additional limitations; such as the inability to verify that a root goal exists; an inability to verify the uniqueness of diagram elements; and an inability to verify the two classes of agents, such as software- and human agents. In addition, an emotional diagram-element is not captured by the tool. The results of this thesis show that all of the verification limitations of the AOM4STS can be correctly verified with the AOM goal-model formalization method as described in this thesis.

The study [4], shows how software requirements outlined in XML can be transformed into UML specifications. A framework exists for mapping XML documents to UML specifications, and subsequently to software-requirement specifications. An XML-based schema is used in describing the elements and sub-element attributes and properties. This thesis demonstrates a method for the mapping of goal-model and T-DM requirement diagrams to XML. This method verifies the correctness of the diagram using DTD-based XML validation to describe the attributes of elements, sub-elements, and properties.
The main benefit of using DTD over XML schema in this research is that DTD provides an abstraction layer for separating validation rules and element transformation. As a result, it is possible to separately define a validation rule, and apply the same rule in any other XML document that contains the transformation of goal diagrams as XML elements. For this case, it is possible to validate the whole document (diagram), unlike in a schema where only portions of the document can be specified and validated.

5.3 Implications From the Application of the T-DM Approach in Blockchain DApp Development

The third research question of this thesis is: *How to develop a blockchain application for trustable organizational collaborations using the T-DM approach?* The results from Publication IV show the application of the T-DM approach in developing a blockchain DApp that enables trustable and symmetric information exchange among several parties involved in complex building construction projects. The DApp models developed show the requirement, architecture and behavioural use-cases of the DApp developed. The application of the T-DM approach in blockchain DApp development extends earlier scientific works [66, 38].

The application level of the blockchain research framework described in [66] captures the research agendas behavioural, design science research and economical areas of inquiry for blockchain applications. The following relevant areas of inquiry are raised for application-level research. In the behavioural aspect: ”How does the use of blockchain affect actors’ behaviour in the light of defacto immutability of data?” In design science research: ”How can we ensure that oracles provide correct information?” In economics: ”How does blockchain affect the cost of doing business?” For the economic aspects, the second running case developed for this thesis shows the inefficiency that arises in complex building construction projects due to distrust between collaborating parties, which resulted in poor information management and the ever-increasing cost of the building construction. The CoPM blockchain DApp developed in Publication IV, and expanded in [56], shows the creation and mapping of construction project requirements to their various stakeholders. These requirements are stored on-chain, and a smart-contract ensures that automated payments are made to contractors and collaborators when project requirements that are linked to specific project milestones have been satisfied. By implementing and adopting the CoPM in building construction, it is expected that the inefficiencies due to poor information management in complex building construction will be eliminated, thereby, reducing the cost of building construction. Also, with the adoption of such blockchain-based project management for construction, it is expected that the collaborators and actors in building construction projects will behave and execute their business operations sincerely. This is because, the information and requirements describing a building construction project are stored on-chain; readily available to the collaborating parties without fear of manipulation. For the question about the correctness of information provided by oracles, IoT agents serve as oracles in the CoPM by executing automated functions outside the blockchain network. IoT-based agents are applied in the design of the CoPM for monitoring and communicating the status of ongoing building construction. However, ensuring the correctness of information provided by IoT agents is beyond the scope of this thesis. Already, the concepts of decentralized agents that provide accurate and trusted information to blockchain DApps have been developed on blockchain layer projects, such as the Chainlink project [3]. These decentralized oracle protocols can be applied in implementing the software agents realized in the design of DApps using T-DM because the modelling approach is technology agnostic.

The dissertation [38], applied case-study research in exploring the how different information systems existing in organizations can be integrated for streamlining information in building construction. The thesis produced a set of theories showing how organizations can share data and create improved business value. The result suggests that the first step in establishing proper business collaboration is to map the inter-dependencies among the actors and clarify their collaboration requirements. This is similar to the process of DApp design using the T-DM approach. The first step in the T-DM approach is to develop the requirement model of a DApp showing the functions executed in business collaboration, how these functions are executed, and to identify the stakeholders/ actors that execute each of these functions. Thus, the requirement model of the CoPM developed in Publication IV shows the mapping of functions, software properties of the functions, and the stakeholders necessary for symmetric information exchange in complex building construction projects. The dissertation [38], also identifies the exchanging of detailed and standardized information as a method for managing complexity within each construction project. In the design of the CoPM using the T-DM approach, information standardization is realized using data formats; such as building objects and building models in deriving the requirements of building projects. These building objects and models are part of building information modelling which provides digital representation of the physical properties of a building element [22].

5.4 Implications From T-DM Implementation and Evaluation Results

The fourth research question of this thesis is: *How to implement and evaluate the T-DM approach and support tool for designing blockchain applications?* The results from Publication V show the implementation of the T-DM approach and the evaluation results.

5.4.1 Implications from T-DM implementation

The T-DM approach is implemented by using meta-modelling concepts to show the heuristics mapping used in deriving the elements and diagrams in the T-DM from the base model. The T-DM meta-model, thereby, describes the semantics aspects of the modelling approach. The support tool is implemented by extending the UML notations to realize the T-DM custom elements. These are part of the results generated in Publication V. The implementation of the T-DM approach and the support tool extends previous scientific works [64, 50, 73, 43, 45].

Various approaches exist for describing the semantics of DSML. Meta-modelling is a common approach for describing the logical aspects and relationships of a modelling approach. The study [64] applies meta-modelling in describing the semantics of a requirement modelling approach. The study [50] applied meta-modelling in describing the risk assessment modelling approach for information systems. The same meta-modelling approach used in [50] is also adopted in this thesis which is a UML class diagram showing the elements, diagram types and their relationships. The element-diagram-type relationships in the T-DM approach are represented by the heuristic mappings used in deriving the different diagram types from the T-DM base requirement diagram. The modelling approach used to describe the AOM in [73] applied an approach different from the meta-modelling. The viewpoint framework showing the mappings of relationships between diagram types in the AOM is used in describing the AOM approach semantics. In the viewpoint framework, the diagram types are grouped into three categories that identify if a given diagram is an interaction model, an information model, or a goal model. Still, this method does not capture the actual mappings of elements in each diagram type, unlike the class diagram meta-model that shows a complete overview of a modelling approach by showing both elements and diagram types mappings.

Extending standard UML notations by the use of UML profile diagrams is one of the common approaches for implementing the notations of a Domain Specific Modelling Language (DSML). The study [43], applies a UML profile diagram in extending the standard UML sequence diagram in developing a custom approach for agent mobility modelling. In this thesis, the UML profile diagram is applied in realizing the custom diagrams such as T-DM requirement, static architecture and use-case diagrams. The requirement diagram is realized by extending the AOM model. Since AOM lacks standardized notations in enterprise-modelling software, the T-DM requirement notations are realized by extending the UML information diagram. The T-DM static architecture and use-case diagrams are realized by extending the UML component and sequence diagrams respectively.

A similar study describes an AOM-based approach for modelling a socio-technical system and prototyped the AOM support tool using a web-based approach [45].

The notations of the AOM are realized using a standard programming language for web application development. Still, with this approach, the modelling notations developed cannot be executed on any enterprise-modelling software. Therefore, to develop an AOM diagram, an analyst must rely on modelling software that only exits in a single web domain. With the prototyping approach of implementing the T-DM on standard enterprise-modelling software, the support tool can easily be exported and used by an analyst in any standard enterprise-modelling software. Although the prototyping of the T-DM presented in this thesis is only implemented on the MagicDraw[3] modelling software, the UML profile diagrams developed can be replicated and exported to other standard modelling software such as the Enterprise Architect[4].

5.4.2 Implications from the T-DM evaluations results

The T-DM approach is assessed for semantic correctness and its practical usefulness in developing blockchain applications. The support tool is examined for its effectiveness and usability in reproducing T-DM diagrams. The evaluation results are part of the findings of Publication V. The evaluation results from the assessment of the T-DM modelling approach and support tool have implications on the earlier scientific works [11, 45].

The T-DM framework evaluation results show that the framework is highly realistic for describing the steps required in building blockchain DApps. The result also shows that T-DM models correctly represent elements and processes for building DApps. For the practical (pragmatic/perceived) usefulness evaluation results, the framework shows very good potential in producing high-quality blockchain DApp software, greatly improving the performance and productivity of analysts and developers in building DApps. Furthermore, the framework shows general applicability and usefulness in jobs related to the design and development of blockchain DApps. Comparing these findings with the results from a similar study [11] showing a process-based approach for modelling innovations in organizations, some similarities and differences exist in the results. The results from [11] show that the modelling approach is highly relevant for tasks related to innovation management and the practical usefulness of the approach is high. Still, the average semantic qualities score is low when compared with the other

[3]MagicDraw https://www.nomagic.com/products/magicdraw
[4]Enterprise Architect software: https://sparxsystems.com/

results as the average score lies just above the 2nd quartile (above 50%). For the T-DM approach, the average score for both the semantic and practical usefulness of the modelling approach lies in the third quartile (above 75%). This shows that the T-DM approach is not only practically useful for building blockchain DApps, but also correctly represents the elements and processes in the blockchain domain.

The evaluation results from the T-DM support-tool assessment show the tool is highly effective and usable in producing correct T-DM models when compared with a freehand sketch tool. The result shows that the modelling effort required for producing T-DM DApp models with the support tool is relatively low while using a freehand sketch, and the modelling effort is very high. The ease of modelling with the T-DM support tool is high and lies above the 3rd quartile (above 50%), while the ease of modelling with the freehand is very low, below the 2nd quartile (below 50%). Comparing these findings with the result from a similar study [45] showing the assessment of the support tool for developing AOM models, there exist some similarities in the results. The result from the assessment of the AOM support tool in [45], shows that modelling efforts for users modelling with the support tool and freehand sketch is the same. Still, the difficulties experienced by the users with freehand sketches are a little higher than the users with the support tool. This is similar to the results of this thesis, showing that the T-DM support tool has higher ease of use than a freehand tool. Therefore, both studies show that to increase the usability and effectiveness of correctly modelling a custom diagram of a given DSML, the implementation of a support tool is necessary.

6 Conclusion

This chapter summarizes the work done in this thesis by showing how the contributions, and research results generated, answer the research questions that provide a guideline for this thesis. In the final part of this chapter, the limitations and future works are presented. The objective of this thesis is to systematically develop a modelling approach for building blockchain DApps that enable trustable IOCs. The modelling approach developed in this thesis is achieved by adapting and extending the AOM and UML modelling concepts by adding blockchain decentralization concepts. The research questions addressed are as follows: RQ1 How to develop a model-driven (T-DM) approach that supports the design and development of blockchain applications for IOC? RQ2 How to develop the ontology of the T-DM approach for building DApps? RQ3 How to develop a blockchain application for trustable organizational collaborations using the T-DM approach? RQ4 How to implement and evaluate the T-DM approach and support tool for designing blockchain applications?

6.1 Answer to RQ1

To answer this research question, a systematic literature review is conducted to identify the problems limiting the use of blockchain applications in organizations. Then, existing approaches for building blockchain DApps are gathered from relevant literature and expert interviews. The identified approaches are then assessed to establish the strengths and weaknesses of the current approaches by using the findings from blockchain limitations as analyses criteria. Next, a model-driven T-DM approach is developed to address the weaknesses that exist in the current approaches for building blockchain DApps. These are shown in Publications I and II.

6.1.1 RQ1 Summary of results

The SLR shows that the purposes for using blockchain in organizations include enabling transparency and trust management, data security or privacy, resource management, data tamper-proofing, and data interoperability. The main problems that affect the use of blockchain applications in organizations include: usability problems, security flaws, transaction cost scalability and privacy leakage problems. The results of the analyses of data collected from the expert interviews and related literature reviews on the existing methods for building DApps, show that some modelling frameworks and methods that support the design and development of DApps currently exist. However, the available methods for building DApps do not satisfactorily address the current limitations of blockchain technology. The T-DM approach is developed to address these limitations by extending AOM and UML concepts. Thus, providing a methods for deriving initial requirements, architecture and behavioural use-cases of blockchain DApps.

6.1.2 RQ1 Summary of contributions

The main contributions generated by answering the first research question include:

- Identification of blockchain as a potential technology that addresses the trust issues that exist in inter-organizational collaborations (Publication I).
- Findings that the examined modelling approaches and common software development techniques are not suitable for building blockchain DApps for organizational use as they do not address the inherent limitations in blockchain technology (Publication II).
- Development of modelling approach for building DApps that addresses weaknesses in current software development techniques by combining the modelling concepts in agent-based systems and standard UML models (Publication II).

6.2 Answer to RQ2

To answer the second research question, the ontology of the T-DM approach is developed by capturing the syntax descriptions of the elements and diagram types in the modelling approach. The syntax aspect outlines the textual, graphical and formal descriptions of the elements and diagram types in the models to ensure the structural correctness and consistency of diagrams produced by applying the T-DM approach.

6.2.1 Summary of RQ2 results

The syntax of the T-DM is presented by first describing the requirement elicitation method of the T-DM approach that shows the non-functional requirements, functional requirements and stakeholders of a given blockchain DApp for IOCs. A running case of the DApp requirement is developed using the requirement elicitation method presented. The syntax correctness of the T-DM approach is captured by an XML-based approach used to formally describe the T-DM-requirement diagram elements, and an XML-based approach for verifying the correctness and consistency of diagram models. The results from the model verification show that the requirement model artefact generated using the requirement-elicitation method described and developed in this thesis is correct and consistent with the T-DM requirement-modelling concepts.

6.2.2 RQ2 Summary of contributions

The main contributions generated by answering the second research question include:

- Development of heuristics containing standardized modelling elements and elements relationships for describing the ontology of the DApp modelling approach (Publication III).
- Development of a T-DM model requirements diagram of an application that solves problems associated with identity document verification validation for web-based financial service providers (Publication III).
- Development of a model-checking method for verifying the syntax correctness of the developed modelling approach for building DApps (Publication III).

6.3 Answer to RQ3

The third research question is answered by designing a CoPM platform that enables symmetric information exchange between multiple parties in building construction using the T-DM modelling approach. The design DApp captures the requirements, architecture and behavioural use-cases of the CoPM DApp.

6.3.1 Summary of RQ3 results

The main value proposition of the platform is to provide a decentralized system for supply chain management, and project management, in the construction industry. The first-level requirements of the designed system are: *onboard new users; onboard new projects; manage digital assets; manage projects; manage* *payments, and manage wallets*. The stakeholders that participate in the platform include human users and software agents. The latter automate the auditing and bidding functions, while the human users are mostly responsible for creating digital assets containing project requirements and executing projects. The main components of the static architecture are derived from the heuristics mapping of the first-level requirements of the platform. The use-cases of the designed platform outlining the necessary activities that are required in executing on-chain events include: *digital-asset creation operations; tender-initiation operations; project-payment operations, and building-maintenance operations*.

6.3.2 RQ3 Summary of contributions

The main contributions generated by answering the third research question are the creation of CoPM DApp models for realizing symmetric information sharing in the multi-dimensional collaboration that exists in executing large building construction projects:

- Development of the CoPM requirement model (Publication IV).
- Development of the CoPM architecture model (Publication IV).
- Development of CoPM behavioural use-cases models (Publication IV).

6.4 Answer to RQ4

The fourth research question is answered by implementing the modelling approach using the meta-model and support tool, using UML profile diagrams. The modelling approach is evaluated to understand the semantic correctness and practical usefulness of the modelling approach. The support tool is evaluated to understand the effectiveness and usability of the tool in reproducing the T-DM diagrams.

6.4.1 Summary of RQ4 results

The T-DM meta-model model provides the semantic representation of the modelling approach and also shows the heuristics mapping used in realizing T-DM diagrams and elements from the base requirement diagram. The UML class diagram is used in developing the T-DM approach meta-model used in this thesis. The support tool is implemented using the UML profile diagrams in standard enterprise-modelling software. The T-DM support tool is represented by the toolboxes that contain the T-DM custom modelling notations in standard modelling software, and the toolboxes are realized in the modelling software by creating and importing three different profile diagrams for implementing the T-DM diagram types. The results from the semantic evaluation show that the T-DM approach provides a realistic and correct representation of the steps required in creating DApps. The pragmatic evaluation results show that the modelling approach is perceived as very useful in producing quality DApp designs. The results from the support tool evaluation show that the tool requires less modelling effort and provides more usability in producing T-DM-diagram models.

6.4.2 RQ4 Summary of contributions

The main contributions generated by answering the fourth research question include:

- Development of a T-DM metamodel representing the semantics of the modelling approach (Publication V).
- Development of a support tool that enables efficient creation of DApp models using the concepts defined in the developed modelling approach (Publication V).
- Evaluation results confirming the semantic correctness and practical usefulness of the T-DM approach (Publication V).
- Evaluation results confirming the effectiveness of the support tool in creating T-DM diagram models (Publication V).

6.5 Thesis Limitations and Future Works

The main limitation of the research conducted in this thesis is the lack of proper support for the software-development aspect of the model-driven approach. This is evident by the absence of code generation for the T-DM approach and support tool. The main ideas behind model-driven software development are to automate both the design and development phase. The current work conducted in this thesis focuses mostly on the design aspect of software requirements engineering. The development support provided by the T-DM approach in building DApps only covers the user-story mappings of on-chain functions in DApp requirement models.

Another important limitation of the thesis is that the significance of emotional goals applicability to the design of blockchain DApps has not been fully explored. Although the AOM emotional goal concepts are captured in the semantic description of the T-DM approach, the applicability of emotional goals in the design of DApps use-cases that address IOC problems are not well captured in this research. Thus, this weakness is considered a key limitation of this thesis.

Other limitations of this research are in the subjectivity of the researcher in analyzing (and interpreting) data, and the risk of generalization. Subjectivity relating to the values and viewpoint of this researcher could have an impact on the interpretation of results when analyzing the strengths and weaknesses of current approaches for building DApps. The subjectivity could also impact the interpretation of evaluation results of the T-DM approach and support tool. Considering the low number of experts interviewed in this study, there is the possible impact of risk of generalizations for the results generated in this thesis. Still, blockchain technology is a new innovation space and the number of experts with the required knowledge to participate in the interviews, webinar and workshop conducted in this thesis, are limited. As a result, gathering a large number of experts to participate in the evaluations conducted in this thesis is a noteworthy limitation.

6.5.1 Future Works

Addressing the main limitations of this thesis is considered as the next logical step for future work. Therefore, it is necessary to conduct well-grounded research to solve the limited development support provided by the T-DM approach in building blockchain DApps. Only then, it will be possible to generate actual smart-contract code for implementing the various on-chain functions captured in the use-stories of blockchain DApps in the T-DM requirement models. First, the common on-chain functions that are usually encountered while designing DApps for IOC have to be identified. Next, the smart-contract codes for realizing these common functions are developed and mapped to the code generation part of the graphical interface of the modelling software.

Additional research is necessary to explore and understand the application of the AOM emotional goals in the design of IOCs. Thus, a standardized list of emotional goals that are applicable in the design of blockchain DApps for IOCs can be developed and formalized.

References

[1] E. Abodei, A. Norta, I. Azogu, C. Udokwu, and D. Draheim. Blockchain technology for enabling transparent and traceable government collaboration in public project processes of developing economies. In *Conference on e-Business, e-Services and e-Society*, pages 464–475. Springer, 2019.

[2] S. H. Adelyar. Towards secure agile agent-oriented system design. In *2015 IEEE International Conference on Cloud Engineering*, pages 499–501. IEEE, 2015.

[3] H. Al-Breiki, M. H. U. Rehman, K. Salah, and D. Svetinovic. Trustworthy blockchain oracles: Review, comparison, and open research challenges. *IEEE Access*, 8:85675–85685, 2020.

[4] H. Aman and R. Ibrahim. Reverse engineering: from xml to uml for generation of software requirement specification. In *2013 8th International Conference on Information Technology in Asia (CITA)*, pages 1–6. IEEE, 2013.

[5] B. Andersson, P. Johannesson, and J. Zdravkovic. Aligning goals and services through goal and business modelling. *Information systems and e-business management*, 7(2):143–169, 2009.

[6] V. Balahontsev, A. Tsikhilov, and A. Norta. A Decentralized Blockchain System for the Attestation and Authorization of Digital Assets. Technical report, Admiral Markets, 06 2019. `0B2jTnqrVhU6uT2lKZVkzeFl1eDdYR1JkaHVMVzlQS2xSRS1r/view`.

[7] M. Bartoletti and L. Pompianu. An empirical analysis of smart contracts: platforms, applications, and design patterns. In *International conference on financial cryptography and data security*, pages 494–509. Springer, 2017.

[8] J. Benet. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*, 2014.

[9] G. Booch. *The unified modeling language user guide*. Pearson Education India, 2005.

[10] G. Booch, J. Rumbaugh, and I. Jacobson. *The unified modeling language user guide*. Addison-Wesley, Upper Saddle River, NJ, 2nd ed edition, 2005.

[11] P. Brandtner and M. Helfert. Multi-media and web-based evaluation of design artifacts-syntactic, semantic and pragmatic quality of process models. *Systems, Signs and Actions: An International Journal on Information Technology, Action, Communication and Workpractices*, 11(1):54–78, 2018.

[12] C. Brodersen, B. Kalis, C. Leong, E. Mitchell, E. Pupo, A. Truscott, and L. Accenture. Blockchain: Securing a new health interoperability experience. *Accenture LLP*, pages 1–11, 2016.

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

[14] A. K. Chopra, F. Dalpiaz, P. Giorgini, and J. Mylopoulos. Modeling and reasoning about service-oriented applications via goals and commitments. In *International Conference on Advanced Information Systems Engineering*, pages 113–128. Springer, 2010.

[15] R. Conrad, D. Scheffner, and J. C. Freytag. Xml conceptual modeling using uml. In *International Conference on Conceptual Modeling*, pages 558–571. Springer, 2000.

[16] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, et al. On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security*, pages 106–125. Springer, 2016.

[17] S. Cullen, P. B. Seddon, and L. Willcocks. *Managing outsourcing: The life-cycle imperative*, volume 139. London School of Economics and Political Science London, 2006.

[18] C. Dannen. *Introducing Ethereum and Solidity*. Springer, 2017.

[19] S. A. DeLoach. The mase methodology. In *Methodologies and software engineering for agent systems*, pages 107–125. Springer, 2004.

[20] D. Djurić, D. Gašević, V. Devedžić, and V. Damjanović. A uml profile for owl ontologies. In *Model driven architecture*, pages 204–219. Springer, 2004.

[21] R. During. *Lachnummer BER: Das Debakel um den Hauptstadtflughafen, Eine Chronik*. Rotbuch Verlag, 2013.

[22] C. M. Eastman, Y.-S. Jeong, R. Sacks, and I. Kaner. Exchange model and exchange object concepts for implementation of national bim standards. *Journal of computing in civil engineering*, 24(1):25–34, 2010.

[23] A. Egberts. The oracle problem-an analysis of how blockchain oracles undermine the advantages of decentralized ledger systems. *Available at SSRN 3382343*, 2017.

[24] M. Farnaghi and A. Mansourian. Blockchain, an enabling technology for transparent and accountable decentralized public participatory gis. *Cities*, 105:102850, 2020.

[25] S. E. Fawcett, P. Osterhaus, G. M. Magnan, J. C. Brau, and M. W. McCarter. Information sharing and supply chain performance: the role of connectivity and willingness. *Supply Chain Management: An International Journal*, 2007.

[26] J. Ferber and G. Weiss. *Multi-agent systems: an introduction to distributed artificial intelligence*, volume 1. Addison-Wesley Reading, 1999.

[27] L. García-Bañuelos, A. Ponomarev, M. Dumas, and I. Weber. Optimized execution of business processes on blockchain. In *International Conference on Business Process Management*, pages 130–146. Springer, 2017.

[28] W. George and C. Lesaege. A smart contract oracle for approximating real-world, real number values. In *International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2019)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[29] P. Giorgini, J. Mylopoulos, and R. Sebastiani. Goal-oriented requirements analysis and reasoning in the tropos methodology. *Engineering Applications of Artificial Intelligence*, 18(2):159–171, 2005.

[30] B. Gray. Assessing inter-organizational collaboration: Multiple conceptions and multiple methods. *Cooperative strategy: Economic, business, and organizational issues*, pages 243–260, 2000.

[31] H. R. Hasan and K. Salah. Blockchain-based solution for proof of delivery of physical assets. In *International Conference on Blockchain*, pages 139–152. Springer, 2018.

[32] H. R. Hasan and K. Salah. Proof of delivery of digital assets using blockchain and smart contracts. *IEEE Access*, 6:65439–65448, 2018.

[33] A. R. Hevner, S. T. March, J. Park, and S. Ram. Design science in information systems research. *MIS quarterly*, pages 75–105, 2004.

[34] C. A. Iglesias, M. Garijo, J. C. González, and J. R. Velasco. Analysis and design of multiagent systems using mas-commonkads. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 313–327. Springer, 1997.

[35] E. Irannezhad. The architectural design requirements of a blockchain-based port community system. *Logistics*, 4(4):30, 2020.

[36] A. Jain, S. Arora, Y. Shukla, T. Patil, and S. Sawant-Patil. Proof of stake with casper the friendly finality gadget protocol for fair validation consensus in ethereum. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 3(3):291–298, 2018.

[37] N. R. Jennings, T. J. Norman, P. Faratin, P. O'Brien, and B. Odgers. Autonomous agents for business process management. *Applied Artificial Intelligence*, 14(2):145–189, 2000.

[38] C. Jose-Carlos. *INTEGRATING INFORMATION SYSTEMS ACROSS ORGANIZATIONS IN THE CONSTRUCTION INDUSTRY*. PhD thesis, Lappeenranta-Lahti University of Technology LUT, 2021.

[39] T. Juan, A. Pearce, and L. Sterling. Roadmap: extending the gaia methodology for complex open systems. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pages 3–10, 2002.

[40] J. Kembro, D. Näslund, and J. Olhager. Information sharing across multiple supply chain tiers: A delphi study on antecedents. *International Journal of Production Economics*, 193:77–86, 2017.

[41] A. Kormiltsyn, C. Udokwu, K. Karu, K. Thangalimodzi, and A. Norta. Improving healthcare processes with smart contracts. In *International Conference on Business Information Systems*, pages 500–513. Springer, 2019.

[42] G. Kostka and J. Fiedler. *Large Infrastructure Projects in Germany: Between Ambition and Realities*. Springer, 2016.

[43] M. Kusek and G. Jezic. Extending uml sequence diagrams to model agent mobility. In *International Workshop on Agent-Oriented Software Engineering*, pages 51–63. Springer, 2006.

[44] O. López-Pintado, L. García-Bañuelos, M. Dumas, and I. Weber. Caterpillar: A blockchain-based business process management system. In *BPM (Demos)*, 2017.

[45] M. Mahunnah, K. Taveter, and R. Matulevičius. An empirical evaluation of the requirements engineering tool for socio-technical systems. In *2018 IEEE 7th International Workshop on Empirical Requirements Engineering (EmpiRE)*, pages 8–15. IEEE, 2018.

[46] M. R. Mahunnah. *Simulation and Prototyping of Sociotechnical Systems Using Agent-Oriented Modelling*. PhD thesis, Tallinn University of Technology, 2019.

[47] M. Marchesi, L. Marchesi, and R. Tonelli. An agile software engineering method to design blockchain applications. In *Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia*, pages 1–8, 2018.

[48] R. Matulevičius, A. Norta, C. Udokwu, and R. Nõukas. Security risk management in the aviation turnaround sector. In *International Conference on Future Data and Security Engineering*, pages 119–140. Springer, 2016.

[49] R. Matulevičius, A. Norta, C. Udokwu, and R. Nõukas. Assessment of aviation security risk management for airline turnaround processes. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXVI*, pages 109–141. Springer, 2017.

[50] N. Mayer. *Model-based management of information system security risk*. PhD thesis, University of Namur, 2009.

[51] T. Miller, B. Lu, L. Sterling, G. Beydoun, and K. Taveter. Requirements elicitation and specification using the agent paradigm: the case study of an aircraft turnaround simulator. *IEEE Transactions on Software Engineering*, 40(10):1007–1024, 2014.

[52] T. Miller, S. Pedell, A. A. Lopez-Lorca, A. Mendoza, L. Sterling, and A. Keirnan. Emotion-led modelling for people-oriented requirements engineering: the case study of emergency systems. *Journal of Systems and Software*, 105:54–71, 2015.

[53] A. Norta, P. Grefen, and N. C. Narendra. A reference architecture for managing dynamic inter-organizational business processes. *Data & Knowledge Engineering*, 91:52–89, 2014.

[54] A. Norta, D. Hawthorne, and S. L. Engel. A privacy-protecting data-exchange wallet with ownership-and monetization capabilities. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[55] A. Norta, C. Wenna, and C. Udokwu. Designing a collaborative construction-project platform on blockchain technology for transparancy, traceablity and information symmetry.

[56] A. Norta, C. Wenna, and C. Udokwu. Designing a collaborative construction-project platform on blockchain technology for transparancy, traceablity and information symmetry. 2020.

[57] L. Panait and S. Luke. Cooperative multi-agent learning: The state of the art. *Autonomous agents and multi-agent systems*, 11(3):387–434, 2005.

[58] Y. Papakonstantinou and V. Vianu. Dtd inference for views of xml data. In *Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 35–46. ACM, 2000.

[59] J. Patel, W. L. Teacy, N. R. Jennings, M. Luck, S. Chalmers, N. Oren, T. J. Norman, A. Preece, P. M. Gray, G. Shercliff, et al. Conoise-g: agent-based virtual organisations. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 1459–1460, 2006.

[60] J. Pavón, J. J. Gómez-Sanz, and R. Fuentes. The ingenias methodology and tools. In *Agent-oriented methodologies*, pages 236–276. IGI Global, 2005.

[61] S. Pike, G. Roos, and B. Marr. Strategic management of intangible assets and value drivers in r&d organizations. *R&D Management*, 35(2):111–124, 2005.

[62] M. Pilkington. Blockchain technology: principles and applications. In *Research handbook on digital transformations*. Edward Elgar Publishing, 2016.

[63] K. QAYUMI. *A NEW DESIGNING APPROACH FOR MULTI-AGENT BASED BUSINESS INTELLIGENCE GENERATION Agent-Oriented Modelling*. PhD thesis, TALLINN UNIVERSITY, 2018.

[64] M. Rosemann and P. Green. Developing a meta model for the bunge–wand–weber ontological constructs. *Information systems*, 27(2):75–91, 2002.

[65] R. Rossar, M. Parve, L. Laas-Billson, and A. Norta. A decentralized insurance marketplace on blockchains for achieving a high level of open market-information symmetry (version: 0.9 b).

[66] M. Rossi, C. Mueller-Bloch, J. B. Thatcher, and R. Beck. Blockchain research in information systems: Current trends and an inclusive future research agenda. *Journal of the Association for Information Systems*, 20(9):14, 2019.

[67] S. Ruohomaa et al. The effect of reputation on trust decisions in inter-enterprise collaborations. *Department of Computer Science Series of Publications A*, 2012.

[68] S. Ruohomaa, L. Viljanen, and L. Kutvonen. Guarding enterprise collaborations with trust decisions: the tube approach. In *Interoperability for Enterprise Software and Applications: Proceedings of the Workshops and the Doctorial Symposium of the Second IFAC/IFIP I-ESA International Conference: EI2N, WSI, IS-TSPQ 2006*, pages 237–248. Wiley Online Library, 2006.

[69] R. Saadi, J.-M. Pierson, and L. Brunie. T2d: A peer to peer trust management system based on disposition to trust. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1472–1478, 2010.

[70] K. Salah, N. Nizamuddin, R. Jayaraman, and M. Omar. Blockchain-based soybean traceability in agricultural supply chain. *IEEE Access*, 7:73295–73305, 2019.

[71] K. S. Soliman, B. D. Janz, T. Puschmann, and R. Alt. Successful use of e-procurement in supply chains. *Supply Chain Management: an international journal*, 2005.

[72] L. Sterling and K. Taveter. *The Art of Agent-Oriented Modeling*. The MIT Press, 2009.

[73] L. Sterling and K. Taveter. *The art of agent-oriented modeling*. MIT press, 2009.

[74] A. Sturm and O. Shehory. A framework for evaluating agent-oriented methodologies. In *International Bi-Conference Workshop on Agent-Oriented Information Systems*, pages 94–109. Springer, 2003.

[75] M. Swan. *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.", 2015.

[76] T. Tenso, A. Norta, and I. Vorontsova. Evaluating a novel agile requirements engineering method: A case study. In *ENASE*, pages 156–163, 2016.

[77] C. Udokwu, H. Anyanka, and A. Norta. Evaluation of approaches for designing and developing decentralized applications on blockchain. In *Proceedings of the 2020 4th International Conference on Algorithms, Computing and Systems*, pages 55–62, 2020.

[78] C. Udokwu, A. Kormiltsyn, K. Thangalimodzi, and A. Norta. The state of the art for blockchain-enabled smart-contract applications in the organization. In *2018 Ivannikov Ispras Open Conference (ISPRAS)*, pages 137–144. IEEE, 2018.

[79] C. Udokwu and A. Norta. Deriving and formalizing requirements of decentralized applications for inter-organizational collaborations on blockchain. *Arabian Journal for Science and Engineering*, 2021. in Press.

[80] C. Udokwu, A. Norta, and C. Wenna. Designing a collaborative construction-project platform on blockchain technology for transparency, traceability, and information symmetry. In *2021 2nd Asia Service Sciences and Software Engineering Conference*, pages 1–9, 2021.

[81] C. Udokwu, C. Wenna, and A. Norta. Designing a collaborative construction-project platform on blockchain technology for transparancy, traceablity and information symmetry. In *Service Sciences and Software Engineering Conference*. ACM, 2021. in Press.

[82] S. Vangen, J. P. Hayes, and C. Cornforth. Governing cross-sector, inter-organizational collaborations. *Public Management Review*, 17(9):1237–1260, 2015.

[83] S. Vangen and C. Huxham. Nurturing collaborative relations: Building trust in interorganizational collaboration. *The Journal of applied behavioral science*, 39(1):5–31, 2003.

[84] M. Wilson, A. Arenas, D. Chadwick, T. Dimitrakos, J. Doser, P. Giambiagi, D. Golby, C. Geuer-Pollman, J. Haller, S. Ketil, et al. The trustcom approach to enforcing agreements between interoperating enterprises. In *Enterprise Interoperability*, pages 365–375. Springer, 2007.

[85] M. Winikoff and L. Padgham. The prometheus methodology. In *Methodologies and software engineering for agent systems*, pages 217–234. Springer, 2004.

[86] M. Wooldridge. *An introduction to multiagent systems*. John wiley & sons, 2009.

[87] R. Xu, Y. Chen, E. Blasch, and G. Chen. Blendcac: A blockchain-enabled decentralized capability-based access control for iots. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1027–1034. IEEE, 2018.

[88] E. Yu. Modeling strategic relationships for process reengineering. *Social Modeling for Requirements Engineering*, 11(2011):66–87, 2011.

[89] P. Zhang, J. White, D. C. Schmidt, and G. Lenz. Design of blockchain-based apps using familiar software patterns with a healthcare focus. In *Proceedings of the 24th Conference on Pattern Languages of Programs*, pages 1–14, 2017.

[90] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom. Fhirchain: applying blockchain to securely and scalably share clinical data. *Computational and structural biotechnology journal*, 16:267–278, 2018.

[91] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE international congress on big data (BigData congress)*, pages 557–564. IEEE, 2017.

Publication I

Udokwu, C., Kormiltsyn, A., Kondwani, T., and Norta, A.
The State of the Art for Blockchain-Enabled Smart-Contract Applications in the Organization

Reprinted with permission from
Ivannikov Ispras Open Conference (ISPRAS)

2018 Ivannikov Ispras Open Conference (ISPRAS)

The State of the Art for Blockchain-Enabled Smart-Contract Applications in the Organization

Chibuzor Udokwu*, Aleksandr Kormiltsyn†, Kondwani Thangalimodzi‡, Alex Norta§
Department of Software Science
Tallinn University of Technology, Tallinn, Estonia
Email: *chibuzor.joseph@ttu.ee, †alexandrkormiltsyn@gmail.com, ‡kthang@ttu.ee, §alex.norta.phd@ieee.org

***Abstract*—The application and use of smart contracts in organizations require a holistic overview. This overview helps to understand the current adoption of this technology and also deduces factors that are inhibiting its use in the modern organization. This study provides a systematic review of previous studies comprising of frameworks, methods, working prototypes and simulations that demonstrate the application of smart contracts in organizations. Understanding the current state and usage of smart-contract technology in an organization is a focal point of this paper. Much progress occurs in developing technologies that support smart contracts, while little understanding exists pertaining to their usage in organizations. In this study, we identify properties of smart-contract applications in different domains of modern organizations. We further analyze and categorize challenges and problems mitigating the adoption of smart-contract applications.**

***Index Terms*—blockchain; applications; smart contract; limitations; use cases; decentralized autonomous organization**

I. Introduction

Organizations face new challenges such as information security [1], trust and transparency between different stakeholders [2], decentralization of working processes [3] and so on. The development of blockchain technology and smart contracts provides new opportunities for organizations to address these problems. Blockchain-enabled smart contracts are computer programs that are consistently executed by a network of mutually distrusting nodes, without the arbitration of a trusted authority [4]. Smart contracts provide organizations the possibility to collaborate and execute self-enforcing contract clauses in a blockchain network without the involvement of a third-party. While smart contracts provide new options for organizations and several studies have been carried out on how smart contracts can be applied to solve several issues affecting modern organizations, little is known about the adoption of smart contracts in organizations. This paper fills the gap by presenting an overview of the business applications supporting smart contracts. The primary research question of this paper is how to successfully adopt smart contracts in modern organizations? The answer to this question helps the reader to understand main domain problems that can be solved by smart contracts and current limitations of this technology. We deduce the following sub-questions from the primary research question. What are the domains of smart-contract applications in established organizations? The understanding of domains using smart contracts helps organizations during decision making processes about smart-contract adoption. What are the main benefits of smart-contract applications in these organization domains? Definitions of a domain help to focus on the main benefits that organizations can gain from smart-contract technology. What are the issues limiting the gains of smart contract usage in the organizations? Understanding the limitations helps to avoid serious problems while adopting smart contracts in working processes.

The remainder of the paper is structured as follows. Section I-A presents important background information. Section II provides the description of the literature-review method used in this study. Section III discusses the analysis of results. Section IV presents problem discussions and provides directions for future research. Section V concludes the results of the overview and defines limitations of the study.

A. Basic Concepts of Smart Contracts

Smart contracts are supported by blockchain technologies [5]. In this section, we summarize the concepts of blockchain technologies as follows - basic blockchain concepts and Merkle hash tree, time stamping nodes and virtual machines, consensus and solidity programming language for coding smart contracts.

Blockchain: The blockchain is a distributed ledger that allows participants to write and update records on the ledger and cryptography ensures that records stored remain the same once added [6]. Records are added to the ledger in form of transactions, and these transactions are hashed and grouped in blocks. Each block is cryptographically linked to the next block. Markel tree or hash tree is a cryptographical method that ensures transactions stored in blockchain are linked with mathematical hashes [7], [8]. This guarantees that no modification can invalidate the entire record. The hashes provide an efficient method to verify any transaction on the blockchain. With this method, records can be verified without going through the entire data stored in the network [8].

Nodes and virtual machines: The blockchain network is represented by the nodes that are connected in peers and each participating node has a copy of the ledger [6]. The nodes are

978-1-7281-1275-6/18/$31.00 ©2018 IEEE
DOI 10.1109/ISPRAS.2018.00029

run by virtual machines, e.g., an Ethereum blockchain node is powered by the Ethereum Virtual Machine (EVM) [7]. Once a new block is accepted in the network, each node updates its record by adding the new block. Transactions are timestamped and sent from the participating nodes. All the nodes in the network agree on a consensus method for adding new records to the ledger [5]. Transactions are grouped in blocks and once a block has been accepted by the network, all the participating nodes add the new block to their copy of the ledger [7].

Consensus mechanism: A consensus method is an agreed method for adding new records to the blockchain by the participating nodes. Consensus methods are grouped into two - voting based consensus and proof-based consensus [5]. Proof-of-Work (PoW) is an ex-ample of proof-based consensus method and its currently being used by the Bitcoin and the Ethereum blockchain at the time of this writing [6], [7]. PoW is a consensus mechanism that allows all the participating nodes to solve a difficult mathematical problem and rewards the first node that solves the problem by selecting the node to add the next block [5]. The Proof-of-Stake (PoS) is another example of proof-based consensus method. However, while PoW motivates a centralization of computing power, PoS moves the decision basis from computing power to possession of stake in the system, such as an amount of cryptocurrency [9].

Programming smart contracts: Solidity is an example of a programming language that provides a method for running a computer code on blockchain nodes [10]. Computer programs that digitally verify, enforce contracts and run on a blockchain network are referred to as smart contracts. The smart contracts are stored and executed in blockchain nodes. With the right access, any user can run and execute smart contract functions from any participating node in the network [7]. In our study, we focus on smart contracts written in Solidity as it is the programming language adopted by the largest blockchain network that supports smart contracts [7].

II. Method of Literature Review

This research uses a systematic literature review method [11]. Literature review gives a good foundation for research in information systems and strengthens information system as a field [12]. A review of literature of smart-contract applications strengthens the field of blockchain within information systems. We conduct the review in four phases [11]. Phase 1 is the review of the purpose and protocol of the study. Phase 2 involves searching the literature and practical screening. In Phase 3, the quality appraisal and data extraction is presented. In Phase 4, we analyze the findings. This literature review method is chosen because it is developed specifically for information-system research [11].

Planning Phase: In the first phase, we design a review protocol as this is an essential element in conducting a systematic literature-review study [13]. Furthermore, a review protocol minimizes biases in a detailed plan [13]. We discuss the purpose of the review and design a protocol, a searching plan, selection criteria, data extraction method, data analyses and present the review results.

Selection Phase: In the second phase of the review, we search for academic articles using the Google scholar database. Since smart contract is a new technology in information systems, we search for journal papers, conference papers and select white papers from 2013–2018. This time frame helps us to find relevant articles from the search engine.

In our search for articles, the keywords and Boolean operators used are as follows: smart contract + business, smart contract + organization, smart contract + organization, blockchain + business, blockchain + organization, blockchain + organization, distributed autonomous organization + business, decentralized autonomous organization + organization, to find papers with limitations and problems, the following key words are used: problem + blockchain, problem + decentralized autonomous organization, and problem + smart contracts. These pairs we use independently in every search.

From the results of our searches, we conduct an efficient screening process, discarding articles not relevant to the study, duplicates, and articles that we do not obtain the full text. After this initial screening, we got a total of 469 papers. This process is depicted in Figure 1 in the study [14].

Exclusion and Inclusion criteria: In the first step of article selection, the exclusion criteria include the following: Articles that are not relevant to the study, articles for which a full paper cannot be downloaded or accessed, articles not published between 2013 and 2018 and articles not related to blockchain technology, distributed autonomous organization, or smart contracts. In the second step, the inclusion criteria include high-quality white papers, journal papers, peer-reviewed conference papers, articles on smart-contract applications in an organization.

Execution phase: In the third phase of the review, we extract data from eligible articles based on the research questions guiding this research and collect information from articles to serve as raw material for the analyses [11].

III. Analysis of Result

In this section, we analyze the reviewed literature to understand how smart contracts are currently applied in the organizations and current limitations that mitigate the adoption of smart-contract applications in the organization. The subsections are structured as follows: Section III-A provides an analysis of smart-contract applications in the organization. Section III-B defines the blockchain issues mitigating smart-contract adoption in organizations.

A. Analyses of smart-contract applications in the organization

We base the analyses of smart-contract applications in the organizations on the following categories: the year of the publication, type of publication and subcategories to identify the properties of the smart contract. The year and type of publication provide information on the quality of the literature reviewed. We identify when the paper was published and if the paper is a peer-reviewed publication. Though there has been too much proliferation of non-peer reviewed papers in the form of white papers in the blockchain smart-contract community, we cannot ignore so many contributions by none peer-reviewed literature in this field.

We further analyze the characteristics of the smart contract that are listed in properties categories in Table 1 in [14]. We identify the following properties of the smart contract: organization, blockchain technology, type of network, application area, problem intended to be solved and status of contribution. In the organization properties of the smart contract, we identify the type of organization that the contribution was developed for. We further divide the organizations into business and public. Under technology property, we identify the blockchain the smart contract was designed to be implemented on. With this property, we identify current blockchain technologies that support smart contracts. In the application area, we identify the type of business process/operation the smart contract is designed to simulate. In the purpose category, we identify the primary goal of applying smart contract in a particular application area, and this can be security reasons, privacy concerns, to build trust, etc. In the status section, we identify if the projects analyzed are theoretical descriptions, prototypes, or working implementations in an organization.

1) Presentation of smart-contract application analyses results: The overview of analyses of smart-contract applications in the organization is presented in the Table 1 in [14]. If a project analyzed does not address the corresponding item or provide sufficient information on the classification, we leave it unmarked. The table shows that 81% of the projects/studies analyzed are published as peer-reviewed publications. 59% of the projects are published in 2017 and 75% of all implemented projects are published from 2017 and later.

Business organizations are the top organizations for smart-contract applications as 87.5% of the projects analyzed are designed for business organization. For the projects that are implement, 75% are specifically designed for business organizations.

In all the projects analyzed, 62.5% are either prototyped, or implemented (working). Only 37.5%, fall into the category theoretical descriptions and proposed frameworks/methods.

In the application area/domain of smart-contract projects analyzed, 71.87% have their application areas in supply chain management (SCM), finance, healthcare, information security, smart city and IoT solutions. Therefore, we identify those at the top domains of smart-contract applications in the organization. Besides, for the projects that are already implemented, 75% are in the domain of healthcare, SCM, and finance. Furthermore, the table shows that transparency and trust are main reasons why an organization may use smart-contract applications, about 44% of the projects mention either transparency, or trust as the primary purpose of the smart-contract application. Other top reasons include data security/privacy, resource management, tamper proof, and interoperability.

Ethereum and Hyperledger fabric are the leading technology for smart-contract applications in organizations, as 50% of the implemented projects are hosted on these platforms. Ethereum network is the technology of choice for prototyping smart-contract projects in the organization as 66.67% of the projects analyzed are prototyped with the Ethereum network. Also, 50% of the prototypes are carried out in public blockchains. Still for implemented projects, 75% are carried out as private networks.

B. Analysis of blockchain issues mitigating smart-contract adoption

We examine existing issues and technical limitations that affect blockchain technology. Our analyses are based on the following factors: the timestamping virtual machine that runs the blockchain nodes, cryptography behind the digital signature, consensus mechanism for confirming transactions and the Solidity programming language for developing smart contract [7], [15]. We further examine how the issues identified affect different application areas of smart contracts in the organization. We only consider top application domains in this analysis.

1) Presentation of blockchain limitation analyses results: We identify 18 limitations of blockchain technologies. From the analyses as shown in Figure 2 in study [14], we found that technologies affected are a digital signature (55.6%), consensus (50%), Solidity programming language (38.9%), consensus mechanism PoW (27.8%) and nodes (27.8%).

We analyze application areas that are affected by presented limitations. Most of the limitations (61.1%) describe issues that affect all application areas that we investigate. 72.2% of the limitations define the issues that affect all applications that use tokens. Limitations that affect an application involving PoW are presented in 66.7% and applications in public blockchains are presented each by 72.2%. Limitations for applications in finance domain are presented in (72.2%) and 66.7% describes limitations that affect IoT, Smart City and SCM domains.

Figure 3 in the study [14], shows how the current limitations of blockchains affect smart contracts in public networks and private networks. The figure shows that there are specific issues in blockchains that affect only the public networks. However, most limitations affect both private and public blockchains. No specific issue affects only private blockchains. The issue of unsustainable consensus method presented in PoW does not affect private blockchains, because they mostly use voting-based consensus method for approving transactions [16]. Due to the control that exists in permissioned blockchains when approving members, the trusted-party-requirement issue is eliminated since all participating nodes are known and trusted.

We further analyze Table 2 in [14] to determine the severities of the current limitations that affect blockchains. With this, we identify important issues mitigating smart contracts usage in the organization. The classification for this analysis is based on the following severity levels: low important, significant and critical.

Figure 4 in the study [14] shows the severity levels of the current issues that affect blockchain technologies. The issues are ordered in their level of importance. The less important issues are in at the bottom of the triangle, significant issues are in the mid-level of the triangle while the critical issues are located at the top of the triangle. The study [14] identifies the following main limitations of blockchains technologies: usability and complexity issues, standardization, lack of testing and practical experience, and design architecture issues. Other significant issues include storage scalability, regulation, soundness of smart contracts, security flaws and bugs, privacy leakage and smart contract lifecycle management and non-tested consensus methods. The less important issues are as follows anonymity, scalability-time, transaction cost, cryptocurrency unpredictability, unsustainable consensus method, trusted third-party involvement and cryptocurrency liquidity problem.

IV. Discussions

In this section, we discuss the results of the analyses we performed in section III. The results of the analyses are presented in Table 1 and 2 of the study [14]. In the first part of this section, we discuss the results of the analyses showing smart-contract applications in the organization. In the second part of this section, we discuss the current blockchain technology issues that affect smart contracts usage in the organization.

A. Application discussion

The results from the Table 1 in study III show that most are mostly peer-reviewed academic publications. This is a necessary step in determining the quality of the projects we evaluate in our study. Though the idea of smart contracts begins with the unpublished manuscript by Szabo Nick in 1994[1] and the implementation starts with the development of the Ethereum virtual machine that provides the possibility of running Turing complete programs on a blockchain in 2014 [7], our study shows that serious effort to develop organization-blockchain applications start in 2017. This is evident as most of the implemented smart-contract projects are carried out in 2017 and later. Blockchain provides an opportunity for both public- and private organizations to run trustless smart-contract applications. The current study shows that business organizations lead in development, implementation, and adoption of smart-contract projects. Our study did not identify any specific smart-contract project designed for military organizations. A search on Google Scholar for "military blockchain applications" returns about 1200 results. However, our study cannot identify an actual prototype, or working blockchain based military organization application. This could be linked to the fact that military applications are usually classified and not available in public domains.

In our study, we identify top application areas, or organization domains that comprise a large number of smart-contract projects. We identify the following organization domains as the top application areas of smart contracts - SCM, finance, healthcare, information security, smart city and IoT solutions. These organization domains have some similarities because their processes involve the participation of several collaborating parties. For instance, SCM involves parties from a supplier, buyer, transporter, etc., who do not trust each other. Blockchain enabled smart contracts therefore are very relevant in these domains as they provide a trustless and transparent system for storing and executing transactions in an immutable way. To validate this point, our study also shows that trust and transparency are the top reasons to adopt blockchains for all the projects we evaluate.

Though there are many blockchains for executing smart contracts, our study shows that Ethereum blockchain remains the blockchain of choice for prototyping smart-contract projects. Still, projects are developed using Ethereum and Hyperledger fabric blockchains. Hyperledger fabric is part of the blockchain tools developed in the open source Hyperledger project. The Hyperledger project seeks to develop compatible and interoperable blockchain frameworks across organizations. IBM is a leading contributor to the Hyperledger fabric project, and also a leading service provider for organizations adopting blockchain projects [17]. Finally, our study shows that most of the working projects are implemented on permissioned networks. This is because of the privacy leakage blockchain issue that causes transactions to be viewable to all participants of the network. Even though the privacy leakage problem affects both private- and public networks, as shown in our study, this problem is reduced in private blockchains because permissioned networks can regulate the membership and participation in their network.

B. Discussion of Limitations

In this section, we discuss current limitations affecting blockchain technologies. In the second part of the section, we discuss current research addressing important limitations that mitigate the adoption of smart contracts.

1) Smart contract and blockchain limitations: Both public- and private blockchain networks face challenges regarding to who is allowed to take part in the network, who is allowed to execute the consensus protocol, and who is responsible for maintenance of the shared ledger [18]. The usage of smart-contract applications in organizations is complicated because of the blockchain complexity- and usability issues. Blockchain networks have complexity and usability challenges, especially for first-time users [19]. Furthermore, the blockchain technology has architecture-design issues that are not acceptable for organization processes [17]. Smart contracts are supported by a few number of programming languages such as Solidity, Cardano, Tezos, Neo etc. The consensus mechanisms are not flexible and are hardcoded into the blockchain [17].

Several limitations affect blockchain-technology consensus mechanisms. Cost is attached to performing transactions in the blockchain to compensate miners, and this may limit the usage of smart contracts in organization applications. Still, this does not apply to private blockchain networks as voting-based consensus methods are usually adopted in permissioned

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

blockchains [5], [17]. The other limitation of the blockchain consensus mechanism is volatility of cryptocurrencies and presents difficulty in making long-term economic decisions [20].

Some of the newly proposed consensus methods lack testing and reliability, similar to the well-known PoW and PoS [5]. The mechanism of a digital signature in the blockchain has several limitations. These include anonymity issues, privacy leakage in transactions and trusted third-party involvement [6], [21]. The usage of smart-contract applications that generate a significant amount of data, is limited by the storage scalability issue of blockchain nodes [21]. This affects smart-contract applications that process and store a significant amount of data.

Some of the newly proposed consensus methods lack testing and reliability, similar to the well-known PoW and PoS [5]. The mechanism of a digital signature in the blockchain has several limitations. These include anonymity issues, privacy leakage in transactions and trusted third-party involvement [6], [21]. The usage of smart-contract applications that generate a significant amount of data, is limited by the storage scalability issue of blockchain nodes [21]. This affects smart-contract applications that process and store a significant amount of data.

The Solidity programming language has several design issues such as security flaws and bugs, soundness and lifecycle management. There are many unknown attack vectors in the ecosystem and there is also the issue of bugs in Solidity code [22]. Additionally, Solidity lacks a formal foundation and smart contract cannot be verified with an algorithmic tool before they are deployed [22]. Another big challenge affecting both public- and private blockchains is the regulation of the network. There is no proper legal framework to address legal issues of tokens, tax and intellectual property in a blockchain network [23], [24].

2) Current research efforts that address significant and critical blockchain limitations: Some of the issues limiting smart-contract adoption have already been addressed. For instance, the PoS consensus method is designed to address the time scalability issue and resource wastage issue in PoW. In PoS, a consensus is achieved by randomly selecting one of the nodes to create the next block based on their stake in the network. The chances to be chosen depends on wealth in the system [25]. As a result, some of the prominent blockchain platforms such as Ethereum are adopting PoS as a consensus method for their blockchains [26]. Qtum which is also a popular blockchain platform, comprises a PoS-consensus method since inception [27].

3) Scalability, third-party involvement and privacy leakage: The issue of storage scalability is a significant limitation of smart-contract applications, as shown in our study. A study [28] proposes using decentralized database storage systems that are linked to an existing blockchain network for storing large sets of data from smart-contract applications. In these systems, immutability and trust can be achieved by applying voting mechanisms and shared replications of stored data. The main drawback of this proposal is that the storage is located outside a blockchain network and the immutability feature that blockchain provides cannot be guaranteed in these systems.

In the case of a third-party involvement, some smart-contract applications require a third party to provide information on the status, or value of an asset. IBM develops microcomputers to address this issue. These computers are so small that they can be attached directly to an asset and provide an update on the asset[2] . This is particularly useful in the SCM domain. The current setup of smart-contract applications in the SCM domain requires an RFID chip, or similar tool to provide information on the status of assets [3], [29]. We consider this a third-party involvement because the information is not directly provided by a blockchain node. With the microcomputers that can be attached to an asset, the asset itself turns into a blockchain node, providing information on its status without the involvement of a third party.

Other research addresses the issue of information confidentiality in blockchains. The design of blockchain enables all participants of a network to view the transactions in the network. Some studies [30], [31] propose the use of private blockchain networks to address this issue. Though participation in private blockchain networks can be regulated, privacy leakage is still an issue, even in permissioned networks. Business processes require that only certain members of an organization have access. Therefore, privacy leakage is still an issue because there is currently no method to control access to information on blockchain networks. Still, study [32] addresses this issue proposing a cryptographical protocol called HAWK that ensures the privacy of data stored in a public blockchain. This is achieved by adding an additional compiler to transform standard smart contracts to a cryptographical protocol among the users of a blockchain. To achieve confidentiality, the public key of a trusted third-party node is used to encrypt and decrypt information among the parties involved in the contract. The use of a trusted third party is a significant drawback of this protocol as this violates the decentralization and transparency principles of a blockchain. There is also an issue of additional cost for verifying the transactions performed using this protocol [32].

4) Testing, design and usability issues: One of the testing issues identified in our study is a lack of proper testing frameworks for blockchain applications. As a recent technology, blockchain applications have not been properly tested and may fail at some point. There are currently no fault injection frameworks available for testing blockchain implementations. Available techniques do not cover Byzantine failures as represented in blockchain applications [33]. The study [33] proposes a new generation of fault injection frameworks for deployment in production to challenge blockchain-based distributed systems. The study proposes using the framework to perturb and verify permissioned blockchain technologies with Byzantine failures. As there is not much practical experience in usage of such projects, blockchain implementations cannot be tested

[2] IBM's New 'World's Smallest Computer' Is Built For Blockchain http://bitcoinist.com/ibms-new-worlds-smallest-computer-built-blockchain/ (accessed May 2018)

appropriately. As a result, organizations do not quickly move their business processes to blockchain solutions.

Blockchain technology is new and therefore, there are unknown attack vectors in the blockchain ecosystems. The Solidity programming language has known security bugs. When used as a payout address, a smart contract acquires the control of a sender's contract and withdraw funds from it [34].

Non-flexible consensus methods limit the usage of specific platforms and thus, not all business requirements can be implemented in such an environment. Organizations should be able to decide what consensus method is most suitable for their applications. To solve this problem, the Hyperledger Fabric platform provides the possibility for smart contract developers to choose the most suitable consensus method for their projects before deploying with a blockchain [17]. Still, the main limitation is that PoS and PoW as the most popular consensus methods, are not available for the Hyperledger platform.

5) Regulation, standardization and smart contract lifecycle management: The development of blockchain standards and regulations is at an early stage [35]. Blockchain is a new technology and therefore it is too early to determine the regulations required [36]. Little research exists about standardization and regulation. Still, standards development is underway to address the risks and abuses. On the other hand, it is essential to avoid overregulation that stifles innovation because establishing laws will have an impact on blockchain technology development [35], [37]. Different blockchain platforms currently exist that are not interoperable, developing blockchain standards are necessary in ensuring interoperability between multiple blockchain implementations, stronger consensus, security and resilience, privacy and trust [38].

Although smart contract lifecycle management is identified in this study as part of the issues mitigating smart-contract adoption in the organization, significant research efforts have been made in addressing this issue. The study [16], proposes a method for managing web-based electronic contract. Although, the study focuses on electronic contracts that are run from web systems instead of blockchain systems as used in smart contracts, we consider the study useful because the latter is a form of electronic contract deployed in a blockchain network. Therefore, the same management cycle is applicable to both smart contracts and other forms of electronic contracts. The study proposes a six-step approach to manage smart contracts comprising proposal, configuration, publication, negotiation, operation, and closure. Thus, the design of a specific contract/agreement holds for a single business operation and once the goal of the operation is achieved and all parties are satisfied, the contract is closed and a new one created. The major drawbacks are that smart contracts are designed for business processes with rules that do not change quickly since modifications, or changes, cannot be performed on smart contracts once they are deployed on a blockchain [7]. Further studies must be carried out to provide an appropriate method for managing the lifecycle of a smart contract.

The study [39] provides a formalized lifecycle management framework for decentralized applications. The study describes a four-phase steps in managing smart-contracts and they as follows – setup phase, description of decentralized governance infrastructure (DGI), enactment phase and termination phase. The setup phase describes the contract negotiation stage as the parties involves describes the services required and agree on a proposal for the contract. The DGI shows in hierarchical order the elements required in executing electronic transaction in the contract proposal. Enactment phase shows the implementation and the behavior monitoring of the contract on DGI. Finally, termination phase shows steps required in dissolving the contract. The framework also provides the possibility for transaction rollback when there is conflict as well as compensation mechanisms in such cases.

6) Blockchain cloud platforms: Blockchain as a Service (BaaS) is a new cloud computing service offered to organizations for processing their business operations via blockchain networks [40], [41]. IBM[3], Microsoft[4] and SAP[5] are leading providers of blockchain-cloud service, while other cloud providers are also integrating existing blockchain platforms as part of service offerings[6]. For instance, Amazon recently adopted QTUM as part of the blockchain platforms for its web service offerings . Different methods are proposed for implementing blockchain cloud service. The study [40] proposes a functional blockchain as a service concept that offers a lighter implementation of top-level business logic applicable in BaaS. The main advantage of this approach is reducing the complexity of developing business logic over blockchain to improve performance.

Although BaaS reduces the complexities for organizations wishing to adopt blockchain for their business process, cloud-blockchain based services still experiences some blockchain limitations that are outlined in this study. Storage oversize is the main limitation of blockchain-based cloud platforms [40]. There is also an issue of third-party involvement that negates the main principles of blockchain.

V. Conclusion

In this paper, we analyze 48 peer-reviewed papers relevant to our research question how to successfully adopt smart contracts in modern organizations? These papers are filtered out from the initial search results of 496 papers. Then we categorize the selected papers by year, type of organization, blockchain technology, type of network, application area, problem intended to be solved and status of contribution. Finally, we examine existing issues and technical limitation of blockchain technology affected.

Our analysis shows that most of the organizations that adopt smart-contract application are private (87.5%). 75% of already implemented projects are designed specifically for private

[3]IBM Blockchain: https://www.ibm.com/blockchain
[4]Microsoft Azure platform: https://azure.microsoft.com/en-us/solutions/blockchain/
[5]SAP Blockchain service: https://www.sap.com/products/leonardo/blockchain.html
[6]Amazon blockchain: https://aws.amazon.com/marketplace/seller-profile?id=884fa2fc-5050-4db4-9110-c9a616d10e99

organizations. Most of the analyzed projects (62.5%) are either prototyped or implemented. Only 37.5% fall under theoretical descriptions and proposed frameworks/methods. Top domains (71.87%) of organizations adopting smart-contract applications are aupply chain management (SCM), finance, healthcare, information security, smart city and IoT solutions. The implemented projects are mostly (75%) presented in healthcare, SCM and finance domains.

After analyzing the organizations adopting smart contracts we check the purposes of doing so. According to our research, the transparency and trust are the main benefits with 44% of smart contracts used in organizations. Other benefits include data security and privacy, resource management, tamper-proof, and interoperability.

We identify 18 limitations of blockchain technologies. The technologies affected are a digital signature (55.6%), consensus (50%), Solidity programming language (38.9%), consensus mechanism PoW (27.8%) and nodes (27.8%). Most of the limitations (61.1%) describe issues that affect all investigated application areas. 72.2% of the limitations define the issues that affect all applications that use tokens. 66.7% of the limitations affect applications involving PoW, applications in public blockchains are presented each by 72.2%. Limitations for applications in the finance domain are presented in 72.2% and 66.7% describe limitations that affect IoT, smart city and SCM domains.

Our study has some limitations, and these include the scope of analyzed projects and an inadequate categorization of decentralized applications to consider viable projects. Our study considers only projects that are in academic publications while there are other smart-contract projects in organizations that are not presented in any academic publication. As future work, we propose a study to evaluate the feasibility and usability of implemented decentralized applications in the blockchain ecosystem.

References

[1] S. Shah, "Use of blockchain as a software component to send messages anonymously for a data trading platform," 2017.

[2] C. Stagnaro, "White paper: Innovative blockchain uses in health care," *Freed Associates*, 2017.

[3] S. A. Abeyratne and R. P. Monfared, "Blockchain ready manufacturing supply chain using distributed ledger," 2016.

[4] M. Bartoletti and L. Pompianu, "An empirical analysis of smart contracts: platforms, applications, and design patterns," in *International Conference on Financial Cryptography and Data Security*. Springer, 2017, pp. 494–509.

[5] G.-T. Nguyen and K. Kim, "A survey about consensus algorithms used in blockchain." *Journal of Information Processing Systems*, vol. 14, no. 1, 2018.

[6] M. Swan, *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.", 2015.

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

[8] G. Becker, "Merkle signature schemes, merkle trees and their cryptanalysis," *Ruhr-University Bochum, Tech. Rep*, 2008.

[9] Z. Liu, S. Tang, S. S. Chow, Z. Liu, Y. Long, and Z. Sui, "Fork-free hybrid consensus with flexible proof-of-activity," Cryptology ePrint Archive, Report 2017/367, Tech. Rep., 2017.

[10] C. Dannen, *Introducing Ethereum and Solidity*. Springer, 2017.

[11] C. Okoli and K. Schabram, "A guide to conducting a systematic literature review of information systems research," 2010.

[12] J. Webster and R. T. Watson, "Analyzing the past to prepare for the future: Writing a literature review," *MIS quarterly*, pp. xiii–xxiii, 2002.

[13] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of systems and software*, vol. 80, no. 4, pp. 571–583, 2007.

[14] C. Udokwu, A. Kormiltsyn, K. Thangalimodzi, and A. Norta, "An exploration of blockchain enabled smart-contracts application in the enterprise," Technical Report, DOI: 10.13140/RG.2.2.36464.97287, Tech. Rep.

[15] J. Sidhu, "Syscoin: A peer-to-peer electronic cash system with blockchain-based services for e-business," in *Computer Communication and Networks (ICCCN), 2017 26th International Conference on*. IEEE, 2017, pp. 1–6.

[16] J. B. Neto and C. M. Hirata, "Lifecycle for management of e-contracts based on web service," in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1, 2013.

[17] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*. ACM, 2018, p. 30.

[18] P. Jayachandran, "The difference between public and private blockchain," *IBM Blockchain Blog, May*, vol. 31, 2017.

[19] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology?—a systematic review," *PloS one*, vol. 11, no. 10, p. e0163477, 2016.

[20] G. C. Dumitrescu, "Bitcoin–a brief analysis of the advantages and disadvantages," *Global Economic Observer*, vol. 5, no. 2, pp. 63–71, 2017.

[21] Z. Zheng, S. Xie, H.-N. Dai, and H. Wang, "Blockchain challenges and opportunities: A survey," *Work Pap.–2016*, 2016.

[22] S. Eskandari, J. Clark, V. Sundaresan, and M. Adham, "On the feasibility of decentralized derivatives markets," in *International Conference on Financial Cryptography and Data Security*. Springer, 2017, pp. 553–567.

[23] M. Fenwick, W. A. Kaal, and E. P. Vermeulen, "Legal education in the blockchain revolution," *Vand. J. Ent. & Tech. L.*, vol. 20, p. 351, 2017.

[24] B. Notheisen, M. Gödde, and C. Weinhardt, "Trading stocks on blocks-engineering decentralized markets," in *International Conference on Design Science Research in Information Systems*. Springer, 2017, pp. 474–478.

[25] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *self-published paper, August*, vol. 19, 2012.

[26] P. Dai, N. Mahi, J. Earls, and A. Norta, "Smart-contract value-transfer protocols on a distributed mobile application platform," *URL: https://qtum. org/uploads/files/cf6d69348ca50dd985b60425ccf282f3. pdf*, 2017.

[27] T. Bocek and B. Stiller, "Smart contracts–blockchains in the wings," in *Digital Marketplaces Unleashed*. Springer, 2018, pp. 169–184.

[28] T. McConaghy, R. Marques, A. Müller, D. De Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare, and A. Granzotto, "Bigchaindb: a scalable blockchain database," *white paper, BigChainDB*, 2016.

[29] E. Hofmann and M. Rüsch, "Industry 4.0 and the current status as well as future prospects on logistics," *Computers in Industry*, vol. 89, pp. 23–34, 2017.

[30] T. Hardjono and N. Smith, "Cloud-based commissioning of constrained devices using permissioned blockchains," in *Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security*. ACM, 2016, pp. 29–36.

[31] M. Vukolić, "Rethinking permissioned blockchains," in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*. ACM, 2017, pp. 3–7.

[32] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 839–858.

[33] M. A. González, W. Rudametkin, M. Monperrus, and R. Rouvoy, "Challenging anti-fragile blockchain applications," in *11th EuroSys Doctoral Workshop EuroDW'17*, 2017.

[34] M. Knecht and B. Stiller, "Smartdemap: A smart contract deployment and management platform," in *IFIP International Conference on Autonomous Infrastructure, Management and Security*. Springer, 2017, pp. 159–164.

[35] D. He, K. F. Habermeier, R. B. Leckow, V. Haksar, Y. Almeida, M. Kashima, N. Kyriakos-Saad, H. Oura, T. S. Sedik, N. Stetsenko *et al.*, "Virtual currencies and beyond: initial considerations," International Monetary Fund, Tech. Rep., 2016.
[36] D. Blummont, "Blocking the future? the regulation of distributed ledgers," 2017.
[37] P. Yeoh, "Regulatory issues in blockchain technology," *Journal of Financial Regulation and Compliance*, vol. 25, no. 2, pp. 196–208, 2017.
[38] A. Deshpande, K. Stewart, L. Lepetit, and S. Gunashekar, "Distributed ledger technologies/blockchain: Challenges, opportunities and the prospects for standards," *Overview report The British Standards Institution (BSI)*, 2017.
[39] A. Norta, A. B. Othman, and K. Taveter, "Conflict-resolution lifecycles for governed decentralized autonomous organization collaboration," in *Proceedings of the 2015 2nd International Conference on Electronic Governance and Open Society: Challenges in Eurasia*. ACM, 2015, pp. 244–257.
[40] H. Chen and L.-J. Zhang, "Fbaas: Functional blockchain as a service," in *International Conference on Blockchain*. Springer, 2018, pp. 243–250.
[41] J. Singh and J. D. Michels, "Blockchain as a service: Providers and trust," 2017.

Publication II

Udokwu, C., Anyanka, H., and Norta, A.
Evaluation of Approaches for Designing and Developing Decentralized Applications on Blockchain

Reprinted with permission from
Proceedings of the 2020 4th international conference on algorithms, computing and systems

Evaluation of Approaches for Designing and Developing Decentralized Applications on Blockchain

Chibuzor Udokwu
Blockchain Technology Group,
Tallinn University of Technology.
Akadeemia tee 15a, 12618 Tallinn, Estonia
chibuzor.udokwu@taltech.ee

Henry Anyanka
Blockchain Technology Group,
Tallinn University of Technology.
Akadeemia tee 15a, 12618 Tallinn, Estonia
scarskos.laws@gmail.com

Alex Norta
Blockchain Technology Group,
Tallinn University of Technology.
Akadeemia tee 15a, 12618 Tallinn, Estonia
alexander.norta@taltech.ee

ABSTRACT

Blockchain as a novel technology has shown the capacity to transform multiple sectors. Due to opportunities such as data security, transparency, privacy and interoperability presented by decentralized applications on blockchain networks, organizations are now moving their inter-organizational functions to blockchain-based systems. Traditional system-design methods are not suitable for developing blockchain systems since they are not capable of capturing new functions that are associated with decentralized applications. This paper seeks to address this gap by exploring existing design methods and frameworks that support the development of decentralized application on a blockchain. To achieve this, case-study based research is used in identifying current software-development techniques used in designing decentralized applications (DApp). Experts in this field are interviewed to identify their chosen techniques for designing and developing blockchain applications. Furthermore, related literature is selected to identify current techniques for DApp-designs based on the state of the art. A systematic analysis of the identified techniques is conducted using the current limitations of blockchain technology as criteria for the analyses. Finally, based on the findings of this study, we propose a new model-driven framework for designing blockchain applications.

CCS Concepts

Software and its engineering → Software creation and management → Software development techniques → Software prototyping

Keywords

Blockchain; DApps; modeling; design; agile; smart-contract

1. INTRODUCTION

There is overwhelming progress in the application of blockchain technology in digitizing business processes and organizational governance [1], [2]. A blockchain is a peer-to-peer distributed ledger to store records that are cryptographically linked [3]. Consequently, many organizations are gradually moving from the era of centralized systems to embrace decentralization [4]. This growing interest of organizations in moving their functions to the blockchain is due to the new opportunities that blockchain provides. For instance, in inter-organizational applications, blockchain technology provides the possibility to secure shared data, provides transparency of processes, enables immutability of information exchanged between organizations and the automating of rules of inter-organizational collaborations [5]. Although the application of blockchain in organizations has resulted in increased adoption of blockchain technologies, research shows that private- and consortium networks are the preferred blockchains for enterprises in building decentralized information systems [4]. This is because public blockchains are generally not scalable in the cost-, time- and storage dimensions [4]. The most popular public blockchain for writing smart contracts still

uses a proof-of-work consensus method for updating records on a blockchain [6]. Recent blockchain platforms have adopted consensus methods such as proof-of-stake, delegated proof-of-stake, byzantine fault-tolerant based systems, etc. in addressing time scalability in blockchains. Still, the issue remains of costs associated to scaling transactions on public blockchain networks. Private blockchains are also referred to as permissioned networks, are not fully decentralized solutions [4] and as a result, issues such as lack of transparency, blockchain governance and non-tested consensus systems still affect these networks.

The previous study [4] identifies and describes common issues that mitigate the full adoption of blockchain applications in organizations. Research shows that some of the blockchain limitations have already been addressed with the introduction of new technologies and protocols [4]. In this study, we consider the following blockchain-technology issues such as usability, transaction-cost scalability, privacy leakage and security flaws to explore how existing software-design- and -development techniques help in addressing blockchain limitations. Therefore, the main goal of this paper is to identify methods, techniques and frameworks for designing and developing distributed applications (DApp), identify their strengths and weaknesses with regards to the current limitations of blockchain technology. Furthermore, the findings from these analyses provide the input for developing a new model-driven framework for designing and developing DApps.

To achieve the research goals and provide a clear guide for this study, we pose the following research question. How to develop a model-driven approach that supports the design and development of DApps? To achieve a separation of concerns and present more clarity, the main research question is further refined to sub-questions as follows. What are the existing software-design- and -development techniques in the blockchain domain? What are the main strengths and weaknesses of the identified techniques? What are the suitable approaches for designing and developing DApps? The rest of this paper is structured as follows. Section 1.1 provides the background of the study by describing the technologies that comprise the blockchain domain. Section 2 describes the case-

ICACS'20, January 6–8, 2020, Rabat, Morocco

ACM ISBN 978-1-4503-7732-4/20/01…$15.00
DOI: https://doi.org/10.1145/3423390.3426724

study research method used in conducting this study. Section 3 outlines the existing methods for developing DApps. Section 4 provides a systematic analysis of the DApp-development methods. Section 5 describes a framework for developing DApps. Finally, Section 6 and 7 provide a discussion and conclusions of the study respectively.

1.1 Background: Blockchain Technology

The blockchain conceptually represents a network comprising of data blocks that are stored in groups and each data block is cryptographically linked with the previous block. To ensure the verifiability and correctness of data stored, blockchain networks use a different kind of consensus mechanism in adding a new block to the network [7] comprising transaction data. Some of the main technologies that support the blockchain network comprise as follows, a distributed storage mechanism [8], distributed consensus framework [9], asymmetric encryption of information [10] and programmable computer code that executes on a blockchain referred to as smart-contracts [11]. These are presented in details below.

The blockchain *distributed storage* provides the possibility for exchanging and synchronizing data through a distributed network comprising of all the participants of the network [8]. The participants of the network are represented as the nodes and the data stored on the network that is replicated across all the nodes.

In the *distributed consensus*, a block of data on a blockchain comprises several transactions published by the network participants. The conditions for adding a new block of data to the network are defined in the consensus protocol. This ensures that all blocks of data added to the network are correct [9]. The two most commonly used consensus protocols include proof-of-work (PoW) and proof-of-stake (PoS) [9]. In PoW, a difficult mathematical computation is presented to the network and the first node to solve this challenge is responsible for creating the next block. Still, in PoS, the block creator is selected randomly and the chances of selection depend on each node's wealth (stake) on the network.

A *smart contract* is described as a computer program that contains conditions that are executed and stored on a blockchain. A smart contract outlines both the conditions for executing an agreement between parties and the logic for the actual execution of the code on a blockchain. If the conditions stipulated in a smart-contract are met, the corresponding logic output of the contract is automatically executed without the interference of a third party [11]. With smart contracts, inter-organizational processes can be executed between parties in a trustless manner.

The *asymmetric encryption* on blockchains involves the use of public-key cryptography in the encryption and exchange of information stored on a blockchain. This encryption system contains two keys, the public key and the private key. The public key is accessible to everyone in the network and is used to identify each participant. The private key is only accessible to the owner and it is used to sign published transactions [10].

2. RESEARCH METHOD

This section presents the case-study based research method adopted for this investigation. This research method defines a qualitative approach for exploring a running case involving detailed data collection from multiple sources such as interviews, document reviews, observations and reports [12]. In this study, we consider expert interviews and reviews of related studies as the primary source of data. Data collection is presented in Section 2.1, the analyses procedure in Section 2.2 and the case presentation in Section 2.3.

2.1 Data Collection

According to the study, the validity of the results from research can be improved by using data from multiple sources as evidence for analyzing the case selected for a study [13]. As a result, this study utilizes semi-structured interviews of experts in software development in the blockchain domain as well as an evaluation of related research outlining frameworks, methods and techniques (FMT) for developing DApps.

Interviews: The use of a semi-structured process for the interviews is employed during this exploration as it is rightly appropriate for an exploratory research design. A compilation of several open-ended questions is offered to the respondents as a means for them to better express their feelings regarding the phenomena based on occurrences and knowledge. A total number of fourteen (14) respondents are interviewed either with a video-messaging application, or in face-to-face meetings with the respondents. The interviews are recorded and transcribed to a format acceptable for further analysis. The complete description of the interview steps and data collected are presented in [14].

Related Literature Review: The analyses of related documents is a procedure for evaluating related materials to provide an additional source of information and thereby gain more insights about information related to the case of study [15]. Therefore, in this study, research works describe FMTs for designing and developing DApps. The frameworks are summarized and the main contribution(s) are identified pertaining to the design and development of DApps.

2.2 Analyses Procedure

In the case-study research, the data-collection phase is followed by the analyses of the collected data to generate useful insights and new knowledge from data [12]. A two-staged systematic analyses step is applied in this study. The first stage involves the identification of topical issues from the interviews and related document reviews. For the interviews, a thematic-analysis computer program termed Routine Data Quality Assessment (RDQA) is used in identifying the topical issues from the interview data. This study adopts the steps in carrying out thematic analysis with the RDQA tool from the study [16]. The full documentation of the interview process and interview data are available in [14]. For the related document reviews, qualitative discussions about the results of the selected studies are used in identifying the topical issues. In both the interviews and related literature reviews, the topical issues represent various FMT for developing DApps. The next phase of the analysis involves a systematic evaluation of the identified FMT for developing DApps to identify the key strengths and weakness. As already outlined in Section 1 of this study, we use the following limitations of blockchain technology as criteria for analyzing the key strengths and weaknesses of FMT for developing DApps. These include usability issues, transaction-cost scalability, privacy leakage and security flaws in DApps. The identified FMT are measured and scored based on their ability to address the listed issues when developing and designing DApps.

2.3 Case Selection and Subject Description

The case selection and subject description are an important phase in the case-study research [12]. Here, we provide a detail description of the running case of this study and purpose for the selection of the running case.

2.3.1 *Developing DApps for Inter-Organizational Collaborations:*

The gold rush in blockchain technology is represented by the peak of the Initial Coin Offerings (ICO) in the year 2017. This saw many ICO-whitepaper projects, over a thousand in count proposing various use-cases and applications of blockchain technology. Still, many of these projects have no meaningful application in the blockchain ecosystem. This resulted in the bubble collapse of most blockchain-related projects in late 2018 [17]. Still, as the technology matures driven by several research activities in the blockchain field, clear application purposes of blockchain technology have been identified. The study [4] identifies the following as the main purposes of applying blockchain technologies in organizations. These include transparency, trust, data security/privacy, resource management, tamper-proof, and interoperability to represent a clear application purpose of using blockchain technology for executing inter-organizational processes. For instance, a trustless and secure exchange of data between collaborating parties as well as a transparent- and interoperable execution of collaborative business processes. Furthermore, case studies are also identified showing collaborative models of blockchain-enabled smart-contract applications in organizations. The study [4] identifies the following as the main application areas of blockchain technology in different organizational domains. These include supply chain management (SCM), finance, healthcare, information security, smart city and IoT-solutions. Blockchain technology still has several limitations that hinder a full adoption in organizations [4]. Moreover, identifying the right software-engineering approach is difficult for designing and developing blockchain applications to mitigate existing limitations. If there exist appropriate methods for developing DApps, identifying the correct organizational application domain is still a difficult task.

3. DAPP DEVELOPMENT METHODS

This section presents various methods for designing and developing DApps based on the data gathered from related literature and expert interviews. The methods from literature is presented in Section 3.1 and the methods from domain experts in Section 3.2.

3.1 DApp-Development Methods Literature

This segment of the paper discusses recent studies that provide techniques for developing DApps. First, we present the privacy-enhancing technique (Ancile), then the ontology-driven approach together with a process-reduction technique are presented and finally, the UML model-driven strategy. Note that only FMTs are considered since they describe how to design, develop, or implement DApps on existing blockchain systems.

3.1.1 *Ancile Privacy-preserving Technique:*

The main goal of the Ancile project is to address the issue of data privacy and -security in the management and handling of electronic health records of patients in healthcare systems. The framework enables secure, interoperable, efficient access and the exchange of electronic healthcare records between patients, healthcare providers and third-parties. To accomplish this goal, the Ancile framework proposes an encryption technique for the secure, multi-party access to information on a blockchain by storing only data references (hashes) on a blockchain. With the use of a proxy, an encryption technique described in the study, Ancile assures that data owners (patients) continue to retain control of their healthcare records and can disable access to the data by any party in the healthcare system. This method does not provide any clear support in modeling the requirements of DApps. Still, the techniques described herein guide the implementation of access control of data on DApps.

3.1.2 *Ontology-driven Approach:*

The study [18] proposes an ontology-driven data modeling technique for designing a blockchain system to enable the tracking of items in a supply chain. The main benefit of adopting ontology-based data modeling in designing a blockchain system is to enable organizations to understand data that is distributed across different databases, especially in inter-organizational processes. Also, ontology-based modeling provides the possibility for the formal verification of inter-organizational operations on a blockchain. To achieve the goal, the method uses a universal modeling language (UML) methodology for transforming data models contained in business processes into a given object-oriented programming (OOP) language. The formal ontology derived from the OOP is further transformed into a smart-contract code that represents the data contained in the original business process. The method described above provides support for designing DApps since data models represented by a formal ontology contains requirements for the implementation of a DApp. Furthermore, the modeling approach provides support for the development stage of DApps since the method describes how the ontology can be transformed into smart contracts that execute on blockchains.

3.1.3 *Process Reduction Technique:*

The method for developing DApps described in the study [19] addresses the scalability issues for optimizing processes that are executed on a blockchain. Although recent blockchain networks that use a PoS-consensus method are scalable in terms of numbers of transactions that can be executed in a given time, there is still a transaction-cost associated. Therefore, the process-reduction method specifically addresses the transaction-cost scalability. Thus, a business process modeling notation (BPMN) represented diagram outlines the operations that are executed on a blockchain. The business process is then transformed into a Petri-net (PN) model using workflow-net techniques. The study [20] shows steps in the reduction of business processes with PN transformations. The workflow represented by the PN is then reduced using a special PN-transformation technique based on state transitions. The reduced process is then transformed to Solidity, a smart-contract programming language and executed on a blockchain. The evaluation of the method shows significant savings in transaction costs associated with a given business when this method is applied. This method supports both the design phase of a DApp as well as the development phase. The BPMN and PN, as well as the process- transformation- and -reduction techniques show the models for outlining the requirements and activities in DApps. The PN-transformation to Solidity provides actual development support for DApps.

3.1.4 *Software Engineering Strategy for DApps.*

The study [21] describes a software engineering modeling approach for designing and developing DApps. The objective of this approach is to ensure that project agreements between project owners and -developers are clearly defined and the requirements are understandable to all the stakeholders. The method proposes the use of UML models such as use-case diagrams, class diagrams and activity diagrams in describing the requirements and architecture of a DApp. The design steps presented in the method are presented in the study [22]. The design steps are outlined by nine activity steps for designers and they are as follows, define the goals, identify the actors, define the user stories, divide the system, design the smart-contract (SC) system, write and test the SC code, design the DApp-system, write and test the code and integrate, test and deploy the DApp-system. The details of these steps are illustrated in Figure 1. The proposed model-driven approach provides support for the design phase of DApps and limited support for the development stage. The UML diagrams in the method clearly describe and present the requirements of a DApp. The model also includes the concept of user stories for mapping the software development of the DApp. The two common traditional software-development methods such as Agile and Waterfall consider user-stories in the software development steps [25].

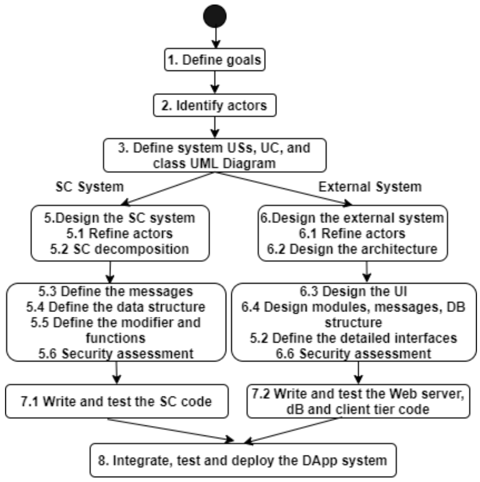


Figure 1: Software-engineering strategy for DApps [17].

3.2 Expert-Interview Results

We next identify and interpret the results obtained by analyzing data gathered from interviews. After transcribing the data with RDQA, the following categories are derived for presenting the findings. These are as follows: current practices in the development of decentralized applications on blockchain and existing designs and modeling techniques in software development.

3.2.1 Current Practices in DApp Development:

With many years of experience in software engineering and of blockchain-system development, the respondents attest to numerous methods utilized in the development of DApps. Most of the respondents (93%) attest to practicing fully the agile methodology in developing DApps due to the flexibility of the processes. The process ranges from initiation, planning, executing, the control to closure of a project. Detailed steps of the agile methodology are presented in [23]. Furthermore, some of the respondents combine the agile method with common task-management tools such as Scrum and Kanban. Scrum proposes the development of software applications through sequential iteration steps while Kanban focuses on maximizing project workflows to reduce lead time [24]. The predominant use of the agile framework is because it ensures that functioning products can be showcased to stakeholders within a short interval. Also, the respondent(s) mention that the procedures that lead to the end goal are not standardized. As a result, diagrams and flows can be easily thrashed, and new techniques adopted within the development process. Therefore, a well-structured DApp-development plan is necessary. The results of the interviews show that developers are fully aware of the current weaknesses in using the traditional mark-up languages and notations in modeling and outlining the requirements of DApps. Developers in this domain are also keen to have a design- and development tool that solves these issues.

3.2.2 Existing Techniques in DApp-Software Development.

In this category, we present the existing modeling tools for DApps gathered from the result of the interviews. The questions that correctly relate to these views are as follows. How designs are documented in their practices? What are the preferred mark-up languages or modeling notations? What are their shortcomings and strengths?

Documentation of design helps to keep track of various aspects of software development and functionalities. Some of the respondents argue since designs are for representations of a proposed system, they prefer drawing models with Adobe, or Wireframe. Next, the models are transformed into Jira[1] processes to represent clearly defined activities and tasks during program development. Other modeling tools mentioned by the respondents include ArchiMate[2], C4 tool[3] and Microsoft Visio[4]. The results of the interview show that the preferred modeling notations of the DApp developers are UML and BPMN. Due to the weakness of traditional modeling notations in capturing various novel functions in blockchain technology, about 35% of the respondents prefer free-hand sketches of their workflows. The free-hand sketch involves the use of hand-drawn activity diagrams to represent various stages of software development [25].

The results highlight the importance of text/free sketches over standard notations such as UML due to the inherent flexibility, especially in communication with stakeholders. In outlining the weakness in the existing modeling techniques, most of the respondents (85%) explain that due to the stakeholders' lack of technical backgrounds, the requirements in the designs are difficult to communicate. Instead, an additional third-party is used in communicating the project requirements. Consequently, time is lost and projects may even fail. Some of the respondents include the use of Adobe diagrams in simplifying the designs. Furthermore, in addressing the complexity issue in UML, some of the respondents prefer the use of BPMN for stakeholder communication due to its simplicity. Still, BPMN cannot present a detailed view of the architectural aspects of software design. Most of the respondents point out the weakness of standard UML is being a legacy technology that is incapable of presenting functionalities inherent to blockchain technology. Finally, the results show that most of the respondents (75%) are not satisfied with the modeling notations they currently adopt in designing DApps due to several shortcomings. The most important findings from the interviews are that developers in this domain do not use any modeling patterns, method, or framework in deriving and describing the DApp-systems they develop. Still, there exist several design frameworks for building DApps, as already shown in Section 3.1. Regarding design techniques represented by the modeling notations, the developers in this domain still use traditional modeling notations such as UML- and BPMN models.

[1] https://www.atlassian.com/software/jira
[2] https://www.opengroup.org/archimate-home
[3] https://c4model.com/
[4] https://www.microsoft.com/en/microsoft-365/visio/flowchart-software

4. ANALYSIS OF METHODS FOR DEVELOPING DAPPS

This section presents a systematic analysis of frameworks, methods and techniques for the design and development of DApps. The criteria for evaluation is presented in Section 4.1 and the evaluation results are presented in Section 4.2.

4.1 Systematic Evaluation Criteria

The criteria for evaluation are based on how the current methods of DApp-design help in mitigating the limitations associated with blockchain technology. This is shown in Table 1. The 'source' outlines the reference of the evaluated items, RL identifies items from related literature and ID identifies items from interview data. The 'category' defines the classification of the item evaluated. Category F represents framework, M represents method and T represents technique. In the context of the analyses performed in this study, a 'technique' shows the use of a single type of modeling notations. A 'method' shows the combination of several notation types. A 'framework' outlines the combination of several methods detailing the complete steps in the design of DApps. The 'application domain' outlines specific inter-organizational operations the selected item is applied to. The 'HC' and 'SC' represent healthcare- and supply-chain respectively. The 'design' and 'development' support show if the phase of DApp-development is supported by the item. The 'blockchain limitations addressed' show the existing blockchain technology weaknesses the item tries to solve. The usability describes the complexity of understanding blockchain technology by project stakeholders. The security flaws associate with the verifiability and correctness of smart-contracts. The cost scalability is linked to the transaction costs associated with blockchain systems. Lastly, privacy leakage associates with the accessibility of private information on blockchain systems. The criterium 'other benefits' shows the additional issue that the selected item seeks to solve.

4.2 Systematic Evaluation Results

We show the results of the evaluations in Table 1. Each row of the table represents a specific FMT generated from either the interview data, or related literature. The symbols '-', '+', '++' show that the selected item is not applicable, applicable, very applicable respectively for each evaluation criteria. The results show that although all the selected items (except Agile Scrum | Kanban) are useful in designing DApps, only the software engineering for DApp framework shows very useful applicability in the design of DApps because it provides the possibility for modeling detailed requirements of DApps. For the development phase, only the extended agile method is very much applicable in this area because it outlines clear steps and task in the DApp-development. In the aspects of existing blockchain limitations, the software engineering for DApp, standard BPMN diagrams and freehand sketch activity diagram show very useful applicability in addressing blockchain usability (and complexity) issues because of the clearly defined steps and easily understandable notations for communications with project stakeholders. For the issue of security flaws associated with smart-contracts, none of the items provide any applicability in this area since they do not contain any formal means of verification of the correctness of smart-contract codes before the execution on a blockchain. Regarding the issue of transaction-costs associated with blockchain systems, only the process-reduction method is very useful. For the issue of privacy of data in blockchain systems, only the Ancile method shows potential applicability in addressing this issue.

Table 1: Evaluations of FMT for developing DApps.

Item	Source (RL/ID)	Category (F/M/T)	Application Domain	Design support	Development support	Blockchain limitation addressed				Others
						Usability	Security flaws	Cost Scalability	Privacy issues	
Ancile	RL	M	HC	+	+	-	-	-	+	-
Ontology-driven	RL	M	SC	+	+	-	-	-	-	data intereoperability
Process reduction	RL	M	SC	+	+	-	-	++	-	code generation
Software Engineering for Dapp	RL	F	All	++	+	++	-	-	-	stakeholders communication
Agile + scrum| kanban	ID	M	All	-	++	-	-	-	-	task management
Standard UML	ID	T	All	+	+	-	-	-	-	technical communication
Standard BPMN	ID	T	All	+	-	+	-	-	-	stakeholders communication
Free-hand sketch diagram	ID	T	All	+	-	+	-	-	-	flexibility

5. A MODEL-DRIVEN FRAMEWORK FOR DEVELOPING DAPPS

The results of the evaluation presented in Section 4.2 show that the main blockchain limitations such as usability, complexity and transaction-cost scalability can be addressed by software-engineering methods. Therefore, we propose a model-driven framework that supports both the design and development phases of DApps. The proposed models also support non-technical communications between DApp-designers and project stakeholders as well as technical communications between DApp-designers and -developers.

We present a three-phase approach for deriving the requirements and describing the architecture of a DApp. The first phase addresses the blockchain usability and complexity issues by using Agent-Oriented Modeling (AOM) in deriving the initial requirements of a DApp. The second phase involves the use of UML component diagrams in describing the project architecture, thereby communicating project requirements to developers. Finally, the last phase employs the use of UML sequence diagrams to describe the use-case protocols and identify activities that are executed on a blockchain. This addresses the scalability issue by reducing and optimizing the number of operations that are published on a blockchain since only on-chain operations are associated with transaction-costs. These are presented as follows: Section 5.1 shows the requirement modeling of DApps. Section 5.2 describes the architecture modeling of DApps. Lastly, Section 5.3 addresses the use-case modeling of DApps.

5.1 Requirement Modeling of DApps

Figure 2(a) shows the AOM elements used for deriving the requirements of DApps. The elements are based on standard AOM-modeling principles [26]. Still, these are further extended to capture the novel concepts in blockchain technology, as we explain in the sequel. The AOM elements are identified as follows, goal, quality goal, emotional goal and the role.

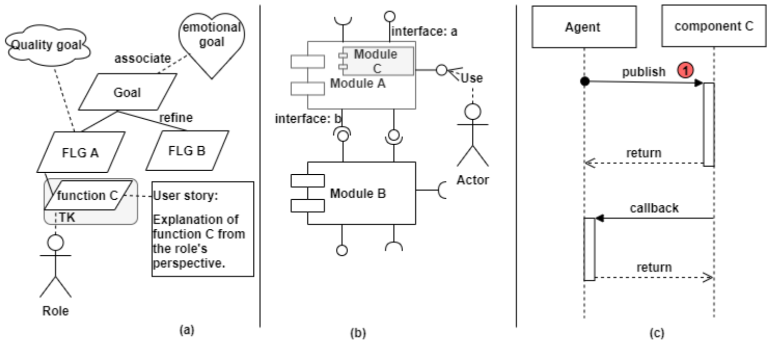


Figure 2: Icons of DApp modeling notations (a) AOM goal-model (b) UML component- (c) UML sequence diagrams

The goal element represents a state, or condition that is to be achieved in a DApp. The quality goals define the specific software-engineering criteria for implementation. The roles represent the software agents and human agents that execute a particular function in a DApp. The emotional goals describe the feelings of stakeholders associated with a particular goal. The goal elements are refined into sub-goals. The first goal from which the other goals are derived from is referred to as the root goal that is the main value proposition of a DApp. This is represented by the element labelled Goal in Figure 2(a). The first-level refinements (FLR) represent the list of objectives that a proposed DApp seeks to satisfy. These are represented by the elements labelled FLG A and FLG B in Figure 2(a). The second-level refinements represent specific functions that can be executed in a DApp. This is represented by the element labelled function C in Figure 2(a). A user story is attached to the executable goals describing the implementation of the function from the associated stakeholders' perspective. We use the greyed box labelled TK behind a goal element to identify goals/functions that are executed on a blockchain (on-chain). The on-chain functions are based on the associated quality goal, e.g., if a quality goal that requires a function to be transparent for other specific stakeholders, other than the role that executes the function, the associated goal is identified as an on-chain function. This optimization of blockchain executable functions based on the quality goal attached to a goal element addresses the transaction-cost scalability issues by ensuring that not all functions in DApps are executed on-chain.

5.2 Architecture Modeling of DApps

Using a heuristics-mapping approach, the static architecture of a DApp is derived from the goal-model diagram. Figure 2(b) shows the elements that describe the architecture of a DApp including the following; component, sub-components, interface and actors. The FLR of a goal-model diagram is used in deriving the main components of the architecture. The second-level refinements are used in deriving the sub-components of the main components. The labels Module A, Module B and Module C in the figure show the main components and sub-components of a DApp-architecture.

The interfaces show the actions performed by an actor. The interface also shows the data object exchanged between two components, or between an actor and a given component. Tokenization is a concept in blockchain systems representing exchanged of values between users and components. The grey shaded component(s) are used to represent the component where tokenization occurs. Based on the direct mapping approach used in this framework, the tokenized function C in Figure 2(a) is executed in component Module C in Figure 2(b). This also represents the heuristic mapping of goals executed on-chain to tokenized components of the static architecture.

5.3 Use-case Modeling of DApps

The use-case modeling of a DApp is described using sequence diagrams to outline the events and activities that are executed on-chain. To achieve this, all the components that involve tokenization are identified and user activities are described that result in the on-chain events. Furthermore, the use-cases of the DApp that incorporate each of the on-chain events are described in detail with a sequence diagram. Figure 2(c) shows the notations of a sequence diagram for outlining use-cases of a DApp. The horizontal columns represent the agents (software agents and stakeholders) and components. The ordered horizontal arrows show the interactions and activities between these agents and components. The red circle denotes an activity that is published (stored) on a blockchain and the numbered identity (ID) inserted in each red circle provides a unique identification for the on-chain activity.

6. DISCUSSIONS

Similar studies have investigated the application of agent-modeling techniques in deriving the requirements of software applications. The study [27] applies case-study research in developing an AOM-modeling method to support agile software-development processes. The main contribution of the study is by describing a visualized approach suitable for agile requirement engineering. The study improves the existing body of knowledge in AOM by providing a method for mapping the requirements contained in the AOM goal-model diagram to agile software-development tasks by using user-stories. The same concept of user-stories is adopted in our proposed approach in requirements modeling of DApps in Section 5.1 of this study. Still, the study [27] does not capture the architectural aspects of software design. The modeling concepts defined in our DApp -modeling framework have already been applied in developing blockchain systems in organizations [28], [29]. Actual enterprise-modeling tools to support this framework are still lacking. This research is necessary to provide a scientific foundation for the framework and provide a clear explanation of the concepts before the implementation of a Dapp-modeling tool proof of concept (PoC). The latter provides a prototype for evaluating the feasibility and reliability of software during development [30].

The UML profile diagram provides the techniques for the implementation of new modeling techniques using diagram elements such as stereotypes, tagged values and constraints to extend standard UML models, or even create a new modeling notation [31]. As a result, PoC for new modeling notations and concepts such as the one described in Section 5, can be created using existing modeling tools such as Enterprise Architect (EA). The EA modeling tool already provides a set of software-development toolkits (SDK) for the implementation of new modeling techniques based on UML profile diagrams. EA enables easy experimentation of different combinations of the AOM goal models with other standardized UML notations. The study [32] shows the implementation of a new modeling concept based on UML profile diagrams using EA.

7. CONCLUSIONS

The main goal of this study is to identify suitable approaches for the design and development of DApps. To achieve this, we adopt a case-study research method in identifying existing frameworks, methods and techniques for the design and development of DApps. The expert interviews and related literature analyses are the main sources of data collection in this study. The results show that there exist some modeling frameworks and methods that support the design and development of DApps. Still, experts in this domain use traditional techniques such as UML and BPMN models. The former has several weaknesses, especially in deriving the initial requirements of DApps. Non-technical stakeholders have difficulties in understanding the UML notations. Also, UML currently lacks the capability for modeling novel functionalities that blockchain provides. Although BPMN provides easily understandable notations, especially for stakeholder communication, BPMN cannot be used in describing the architectural aspect of a DApp.

The analyses of the existing frameworks and methods for DApp-designs show that the identified approaches do not satisfactorily address the current limitations of blockchain technology. Furthermore, the analyses of the application of the existing frameworks and methods in the DApp design identify two blockchain limitations that can be mitigated by software-engineering methods. These include the usability of blockchain and the transaction-cost scalability. Usability refers to the complexity of understanding the blockchain-technology functions and -application. Transaction-scalability costs refer to the executing organizational processes on a blockchain. Based on these findings, we propose a model-driven DApp-design method that addresses these issues. The proposed framework ensures easy communication with stakeholders in deriving the initial requirements of DApps using an extended AOM model. To address the issue of transaction-cost scalability, an optimization technique for on-chain activities is used based on quality goals associated with the functions in a DApp. Thereby, we ensure only specific activities are executed on a blockchain. The architecture and use-cases of a DApp are presented using component- and sequence UML models respectively. These standard UML models are extended to capture blockchain concepts such as tokenization and on-chain events.

The main limitation of this study is the absence of an evaluation for the proposed approach for DApp-design. Therefore, future work from this study comprises the formal description of the extended AOM-modeling techniques for deriving the initial requirements of a DApp. Furthermore, a PoC must be derived in a study showing the implementation of the tool support for creating DApps using extended AOM- and UML modeling notations.

8. REFERENCES

[1] O. López-Pintado, et al. Caterpillar: *A blockchain-based business process management system*. In BPM (Demos), 2017.

[2] J. Mendling, et al. *Blockchains for business process management challenges and opportunities*. ACM Transactions on Management Information Systems (TMIS),9(1):1–16, 2018.

[3] M. Crosby, et al. Blockchain technology: *Beyond bitcoin. Applied Innovation*, 2(6-10):71, 2016.

[4] C. Udokwu, et al. *The state of the art for blockchain-enabled smart-contract applications in the organization*. In 2018 Ivannikov Ispras Open Conference (ISPRAS). IEEE, 2018.

[5] B. Scott, et al. *Exploring the rise of blockchain technology: Towards distributed collaborative organizations*. Strategic Change,26(5):423–428, 2017.

[6] V Buterin. *Ethereum 2.0 spec–casper and sharding*, 2018.

[7] M. Crosby, et al. Blockchain technology: *Beyond bitcoin. Applied Innovation*, 2(6-10):71, 2016.

[8] H. Natarajan, et al. Distributed ledger technology and blockchain. World Bank, 2017.

[9] A. Baliga. *Understanding blockchain consensus models*. Persistent, 2017(4):1–14, 2017.

[10] M. Pilkington. Blockchain technology: principles and applications. In Research handbook on digital transformations. Edward Elgar Publishing, 2016.

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

[12] Creswell, John W., et al. "Qualitative research designs: Selection and implementation." *The counseling psychologist* 35.2 (2007): 236-264.

[13] P. Runeson et al. *Data collection.Case-study Research in Software Engineering*. Hoboken, NJ, USA: John Wiley & Sons, Inc, pages 47–60, 2012.

[14] H. Anyanka. *Systematic Analysis of Methods/Frameworks for Developing Decentralized Applications on Blockchain*. Master's Thesis, Tallinn university of technology (2017).

[15] G. A Bowen et al. *Document analysis as a qualitative research method. Qualitative research journal*, 9(2):27, 2009.

[16] V. Mahnic. *Improving software development through combination of scrum and kanban. Recent Advances in Computer Engineering*, Communications and Information Technology, Espanha, pages 281–288, 2014.

[17] D. A. Zetzsche et al. *The ico goldrush: It's a scam, it's a bubble, it's a super challenge for regulators*. University of Luxembourg Law Working Paper, (11):17–83, 2017.

[18] H. M Kim and M. Laskowski. *Toward an ontology-driven blockchain design for supply-chain provenance*. Intelligent Systems in Accounting, Finance and Management, 2018.

[19] L. García-Bañuelos, A. Ponomarev, Marlon Dumas, et al. *Optimized execution of business processes on blockchain*.

[20] B. F van Dongen, et al. *Verification of epcs: Using reduction rules and petri-nets*. In International Conference on Advanced Information Systems Engineering, Springer,2 005.

[21] G. Lallai, A. Pinna, M. et al. *Software engineering for dapp smart contracts managing workers contracts*. Proceedings of the Distributed Ledger Technology (DLT'20), 2020.

[22] L. Marchesi, et al. *Abcde–agile blockchain dapp engineering.* arXiv preprint arXiv:1912.09074, 2019.

[23] L. Williams and A. Cockburn. *Agile software development: it's about feedback and change*. IEEE computer, 2003.

[24] V. Mahnic. *Improving software development through combination of scrum and kanban. Recent Advances in Computer Engineering*, Communications and Information Technology, Espanha, pages 281–288, 2014.

[25] Bartelt C, Vogel M, Warnecke T. Collaborative Creativity: From Hand Drawn Sketches to Formal Domain Specific Models and Back Again. InMoRoCo@ ECSCW 2013.

[26] L. Sterling and K. Taveter. *The art of agent-oriented modeling*. MIT press, 2009.

[27] T. Tenso, et al. *Evaluating a novel agile requirements engineering method*. In ENASE, pages 156–163, 2016.

[28] A. Norta, et al. *A privacy-protecting data-exchange wallet with ownership-and monetization capabilities*. In2018 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2018.

[29] V. Balahontsev, et al. *A blockchain system for the attestation and authorization of digital assets* version 1.0.

[30] Bell, Jeffrey, et al. "Software design for reliability and reuse: a proof-of-concept demonstration." *Proceedings of the conference on TRI-Ada'94*. 1994..

[31] T. Ziadi, et al. *Towards a uml profile for software product lines. In International Workshop on Software Product-Family Engineering*, pages 129–139. Springer, 2003.

[32] B. Contreras and S. Stefan Krauss. *Mdg profile for cast: Uml extension for cast (causal analysis based on stamp).* 2017.

Publication III

Udokwu, C., and Norta, A.
Deriving and Formalizing Requirements of Decentralized Applications for Inter-Organizational Collaborations on Blockchain

Reprinted with permission from
Arabian Journal of Science and Engineering

Arabian Journal for Science and Engineering (2021) 46:8397–8414
https://doi.org/10.1007/s13369-020-05245-4

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Deriving and Formalizing Requirements of Decentralized Applications for Inter-Organizational Collaborations on Blockchain

Chibuzor Udokwu[1] · **Alex Norta**[1]

Received: 2 October 2020 / Accepted: 9 December 2020 / Published online: 17 March 2021

Abstract

Traditional information systems that enable organizations to collaborate, share information and resources, have several weaknesses such as security, interoperability and transparency issues. As a result, organizations are now moving their inter-organizational collaborations to blockchains. The latter are decentralized networks that allows participants to store and replicate information across multiple nodes, thereby providing immutable and trustable access to data without relying on a central authority. The state of the art shows several blockchain technologies such as smart contracts, consensus methods and decentralized storage to enable enterprises in executing their collaborations on blockchains. However, absent is a suitable software-engineering-driven framework for designing blockchain decentralized applications that enable inter-organizational collaborations. This paper fills the gap by proposing a framework for developing decentralized applications for organizational collaborations by first showing a model-driven method for deriving requirements of blockchain applications. Therefore, this paper presents a goal-modelling method to systematically describe the requirements of a running case about inter-organizational collaboration. A goal-model diagram produced from the running case is formally evaluated for correctness and consistency using the model syntax-verification method described in this paper.

Keywords DApps · Requirements · Goal models · Smart contract · Formalization · Blockchains

1 Introduction

Inter-organizational collaborations are business functions that require the participation of multiple organizations in achieving specific objectives [1]. Traditional systems for enterprise collaborations experience issues related to data interoperability, process transparency and insecurity. The security issues are caused by a lack of access control on shared information when executing inter-organizational functions [2]. Transparency issues occur due to a lack of visibility for information exchanged in collaborative processes [3]. The interoperability issues exist because different organizations participating in collaborative systems have different a information system setup and real-time sharing of data is impossible [4]. To address these concerns, organizations are moving their applications to blockchains [5] for executing inter-organizational collaborations (IOC). Blockchain systems are peer-to-peer (p2p) networks allowing participants to write and update records on ledgers where cryptography ensures immutable traceability. Blockchain's technological concepts such as consensus algorithms, smart-contracts, digital signatures and decentralized storage, enable organizations to execute their IOC securely and transparently [6].

Consensus algorithms provide clearly defined rules for adding information to a blockchain. Digital signatures are based on public-key cryptography (PKC) and provide unique identification for the participants on a blockchain. Decentralized storage is a replication system that stores information on a blockchain across all the nodes of the participants [7]. Smart contracts are computer programs that run on a blockchain network. Decentralized applications (DApps) are software applications that run on a blockchain network and apply blockchain-technological concepts in their designs [8]. With respect to IOC, the main focus of DApps adoption in the organizations is to address transparency, trust and tamper-proof issues through immutable data storage, ensuring data secu-

Chibuzor Udokwu
chibuzor.udokwu@gmail.com

Alex Norta
alex.norta@taltech.ee

1 Blockchain Technology Group, Department of Software Science, Tallinn University of Technology, Akadeemia tee 15a, 12616 Tallinn, Estonia

rity by implementing access control mechanism using PKC, solving interoperability issues by providing instantaneous access to data using replicated storage systems [5]. Some existing projects already show how blockchain applications can be used in solving transparency and security issues in IOC. The study [9] shows the application of blockchain DApp in ensuring traceability of product delivery in supply chains. The studies [10,11] show the use of blockchain DApps in implementing secure and tamper-proof delivery of digital- and physical assets in a multi-party collaboration system.

Despite the opportunities blockchains provide in solving problems that exist in IOC, still, a suitable software-engineering-driven framework for designing and developing DApps is still lacking. The study [12] evaluates approaches for designing and developing DApps and identifies several weaknesses. For instance, the citation [13] shows an ontology-driven design method for building traceability enhanced DApps. The citation [14] is a Petri-net-based process-reduction method for enhancing the scalability of cross-organizational processes executed on a blockchain. The ancile framework [15] is a privacy-preserving technique for building DApps. Still, these approaches do not provide a technique to systematically describe the requirements and architecture of DApps. Furthermore, the study [12] proposes a decentralized agent-oriented modelling (DAOM) framework for building decentralized applications referred to as the DAOM. This modelling approach is based on three diagram types, the requirement, architecture and behaviour diagrams. The requirement diagram systematically describes the functional and non-functional requirements of DApps and is adapted from agent-oriented modelling [12]. The extended UML-component diagrams and UML-sequence diagrams describe the architecture and behaviour diagrams of the DAOM [12].

Although citation [12] provides a detailed description of DAOM and its modelling notations, a proper ontology description of the modelling framework is still missing. This study addresses this gap by providing an ontological description of the DAOM-requirement diagram and also provides a method for verifying the correctness and consistency of the DAOM-requirement diagram syntax. The focus of this study is to describe the extended AOM used in realizing the DAOM-requirement diagram. There exist several variants of goal modelling techniques for software development such as Tropos [16], service-oriented goal models [17] and business-model alignment with goal models [18]. The AOM is chosen for generating initial requirements of the DApps because of the inter-organizational nature of blockchain applications in organizations [5]. The AOM provides an uncomplicated technique in assigning roles to specific functions [19].

Therefore, to systematically describe the ontology of DAOM-requirement diagrams, first, we show the elements contained in this diagram types. Next, we show the application of the DAOM framework in describing the requirements of DApps that enable inter-organizational collaboration in a given running case. Lastly, we describe a method for verifying the syntax of DAOM-requirement diagrams with an application to the presented running case. To achieve these, we pose the following research question. How to develop the requirements of decentralized blockchain applications for inter-organizational collaborations? For a separation of concerns, we derive the following sub-questions. What are the reference-architectural requirement sets and stakeholders for inter-organizational collaboration? What is the set of heuristics for specifying a DAOM goal model? What is the ontology for creating DAOM-design tool support? The rest of this paper is structured as follows: Section 2 provides the background of the study by describing the foundational blockchain technologies, the research method and a presentation of the running case. Section 3 provides an answer to the first research sub-question by describing a method for outlining the architectural requirement of a DApp for IOC as it relates to the functional requirements, software requirements and stakeholders in such systems. Section 4 provides the answer to the second research sub-question by applying the requirement-derivation method on a running case of inter-organizational collaboration, thereby developing a goal-model diagram. Section 5 provides the answer to the third research sub-question by describing a method for verifying a goal-model diagram showing the requirements of DApps for IOC. Section 6 provides the discussions of the results from this study and also compares the results with similar studies. Finally, Sect. 7 provides conclusions, limitations and future work for the study.

2 Background

We provide the background on blockchain technologies, the research method for this study and the running case. Types of blockchain consensus and smart contracts are presented in Sect. 2.1. The research method in Sect. 2.2 describes the scientific method that provides the guideline for this study. The running case of Sect. 2.3 shows an example of an inter-organizational collaboration used in evaluating the DAOM modelling and validation method developed in this study.

2.1 Blockchain Technologies

The main technologies comprising a blockchain are presented as follows.

2.1.1 Decentralized Consensus

To ensure the correctness and consistency of information added to a blockchain network, the blockchain consensus

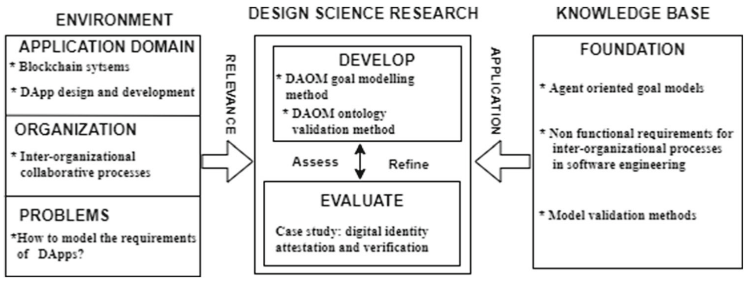


Fig. 1 Design-science research method, adapted from [27]

defines the method records that are updated in a blockchain network [7]. For an IOC that adopts DApps concepts, a blockchain-consensus system ensures that only correct information validated by the collaborating parties can be recorded in the network. This thereby increases the transparency in IOC processes. Proof-of-work (PoW) is the most popular consensus method used by Bitcoin and Ethereum. Bitcoin is the largest blockchain network, while Ethereum is the largest blockchain network for executing smart-contracts [20]. In the PoW consensus method, a difficult mathematical puzzle is presented and the participant that first solves the puzzle is selected to add the next record on a blockchain ledger. Due to scalability issues and resource consumption in PoW, researchers have proposed a proof-of-stake (PoS) consensus method as a viable alternative to PoW. In PoS, participants are selected to add the next record on the ledger based on the amount of stake deposited on the network [21].

2.1.2 Public-Key Cryptography

The PKC provides a system for unique identification of participants in a decentralized network using the public- and private-key pair [22]. With the public key, participants in an IOC can be identified and with their private keys, transactions can be signed. The PKC thereby provides a tamper-proof system of source verification for any given activity in IOC. Furthermore, with the asymmetric encryption in the PKC, access control can be implemented on IOC processes, thereby ensuring that specific IOC functions can only be executed by certain parties on the network [23]. As a result, for IOC that adopts DApp concepts, the PKC in blockchains strongly address the security issues currently experienced in traditional systems for executing IOCs.

2.1.3 Decentralized Storage

Records stored on a blockchain network are replicated across the participating nodes [23]. The existing blockchain storage can be extended further using decentralized databases that provide extended repositories for blockchain systems [24] and therefore, preserve digital assets that are exchanged in IOC-executed blockchain networks. As a result, for an IOC that adopts the DApps concepts, a decentralized storage ensures that data resulting from the execution of IOC functions is accessible in real time by all participants. The real-time access to data solves interoperability challenges in the current systems for IOC.

2.1.4 Smart Contracts

They are digitally verifiable and enforceable computer programs that run on a blockchain network [25]. Inter-organizational collaborative processes can be transformed to smart-contract workflows and executed on blockchain networks [26]. Business logic and rules can be coded into a smart contract ensuring that IOCs are executed in a trusted manner without the need to rely on a centralized authority.

2.2 Research Method

The research conducted in this study follows the design-science research (DSR) methodology that provides a rigorous framework for the creation and evaluation of information-system (IS) artefacts [27]. The framework provides methods for creating new theories represented as constructs, models and methods. This research focuses on the method for deriving requirements for collaborative systems on the blockchain. In Fig. 1, we show the application and instantiation of the DSR method in this study adapted from [27]. The three main pillars of the DSR are environment, knowledge bases and IS research. The environment and knowledge base provide the main input for conducting successful DSR. The environment represents the relevance of the research in addressing existing problems in organizations. The application domain of this research is in the blockchain systems especially for the design and development of DApps. The organizational process applicable to this study is inter-organizational, collaborative processes. Furthermore, the domain problem addressed by this study is the issue of requirement modelling for DApps.

The knowledge base represents the scientific foundation for providing useful models, methods, frameworks and validation criteria for the artefacts produced in this research. The artefact developed in this study is the DAOM requirement modelling ontology. The artefact is developed by extending the AOM and integrating blockchain-relevant requirements [12,19]. To achieve this, we identify software requirements necessary in executing IOCs from literature and then map the blockchain relevant requirements. Furthermore, we show and describe all the elements that are in the DAOM-requirement diagram.

As part of the DSR method shown in Fig. 1, evaluation is necessary for artefacts developed in the research. We apply the DAOM-requirement ontology to a running case of IOC. Thus, functional-, non-functional requirements and stakeholders in the given DApp use-case are clearly identified and described. As a result, a requirement model is developed of DApp used in the attestation and verification of digital identity. To show the syntax correctness and consistency of the DApp requirements developed, we examine related articles on model-checking methods and identify the most suitable approach for verifying and validating models. The model-checking approach identified is then further developed specifically for DAOM-requirement diagrams for checking the correctness and consistency of the diagram elements.

2.3 Running Case

We consider a running case as elaborated in [28] that describes the status quo in know-your-customer (KYC) authentication and the complexities experienced in the verification and authorization of digital assets for online financial transactions. These complexities are experienced by both the client who wishes to purchase a service and the online merchant who provides the service. Thus, a client wishes to access services from different online merchants. The client is required to submit personal identification documents to the service provider, and the submitted data are analysed to determine if they are compliant with the institutional requirements. This results in time loss needed to complete an online transaction as the client is required to repeat the same verification process for other merchants too. Moreover, it is not certain that the client passes the identity verification when the process is repeated several times by different identifiers. Figure 2 shows a pictorial representation of the running case, and we refer the reader to [28] for further details.

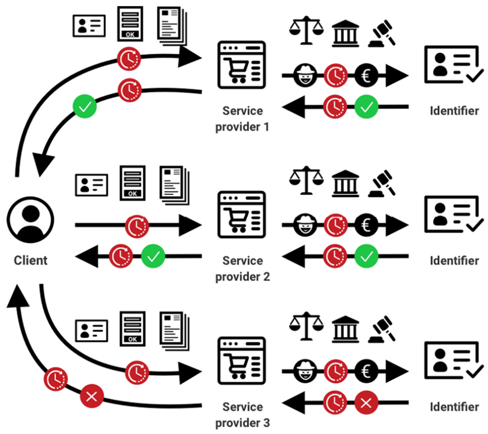


Fig. 2 Running case about the status quo in document authentication for the financial industry [28]

3 Requirements of a Decentralized Application for Inter-Organizational Collaborations

This section provides answers to the first research sub-question by describing a method for deriving the functional requirements, non-functional requirements and stakeholders of decentralized applications for IOC. Section 3.1 provides the method for deriving non-functional requirements, Sect. 3.2 gives the method for deriving functional requirements, and Sect. 3.3 shows the method for deriving stakeholders of DApps for IOC.

3.1 Non-Functional Requirements of a Decentralized Application for Inter-Organizational Collaborations

Non-functional requirements (NFR) are a set of criteria for assessing the operations of a system [29]. Therefore, it is necessary to understand and identify all the necessary criteria for assessing the behaviour of a decentralized system. To develop the standardized NFRs for designing blockchain DApps for IOC, first, we identify NFRs for systems that support IOC and NFRs for blockchain DApps separately. Based on the list, we map the relevant requirements that are useful for designing blockchain DApps for IOC.

For NFR in IOC, several studies exist that provide standardized non-functional requirements for developing software applications. The study describes an automated approach for the elicitation and classification of NFR requirements [30]. Citation [31] provides a goal-model approach for organizing NFR and also provides a standard list of NFRs for software design. A set of NFRs relevant for designing cloud-based systems is presented in the study [32]. A reference architecture containing a set of NFRs for designing collaborative systems is presented in [29].

The study [35] describes the process of outlining NFRs and integrating them to a UML-based conceptual model of software systems. Based on the listed literature, only the citation [29] provides a standardized list of NFR relevant in designing software systems for IOC. Therefore, the literature [29] provides the IOC basis for mapping the DApp-IOC relevant NFR. Hence, the useful NFR requirements for designing collaborative systems include modifiability, integrability, interoperability, security, high automation, flexibility, usability, scalability, applicability, feasibility, portability and performance.

For the blockchain DApp NFR, some studies already provide specific software requirements for designing blockchain-based systems. Although the study [36] provides a software engineering-based approach for building blockchain DApps, no specific list of NFRs is contained for designing blockchain systems. The study provides architectural requirements for building DApps and outlines specific NFR useful in design blockchain DApps [33]. The listed NFRs are as follows: trustability, interoperability, verifiability and security (privacy). The study identifies application domains of blockchain DApps in organizations and identifies a set of non-functional requirements for developing such systems [5]. The listed NFRs are as follows: transparency, trustability, interoperability and security (tamper-proof and privacy). The study [34] describes a software pattern for designing blockchain DApps and provides useful NFRs for such pattern-based development. The listed NFR are as follows: security, privacy and scalability. Since these three studies [5,33,34] provide a specific list of NFRs, they form the blockchain-DApp basis for mapping the DApp-IOC relevant NFR. Hence, the useful NFR requirements for designing blockchain DApps are interoperability, verifiability, transparency, trustability, scalability and security (tamper-proof and privacy).

Table 1 provides the aggregated list of NFRs relevant for designing blockchain DApps and also relevant designing systems for IOC. From this aggregated list, we identify specific NFRs that are useful for designing blockchain-DApps-based IOC systems. The first column of the table identifies specific NFR, and the second column shows literature sources, such as IOC systems related literature and blockchain-DApp-related literature. The third column provides a general explanation of the NFR, and the last column provides a blockchain DApp-IOC context for the NFR. For the NFRs where no possible blockchain-DApp and IOC context explanation exist, we consider such an NFR as unuseful for designing blockchain DApps for IOC.

Based on the mappings in Table 1, the following are identified as the NFRs relevant for the designing of blockchain DApps for inter-organizational collaborations: modifiability, integrability, interoperability, security, high-automation, usability, scalability, portability, performance, transparency and trustability. For the NFR usability and portability, the same collaboration-context description applies to the blockchain context description. Interoperability enables access to digital assets across collaborating parties. The modifiability provides a possibility for updating smart-contracts based on the smart-contract lifecycle management adopted by a DApp [37,38]. This is necessary since the business logic behind a DApp can be changed as the organization changes. Therefore, it is necessary to update the smart-contracts that execute corresponding business logic. Security implies access control on digital assets stored on a blockchain. High automation describes a function that is automatically executed by a software agent. Scalable represents an increased number of users executing and publishing transactions on a blockchain. Performance measures the numbers of transactions published, and transparency represents the verifiability of a blockchain function. Trusted indicates the capability to enforce a condition outlined in a smart-contract. For the remainder of NFR to design blockchain DApps for IOC, their explanations are provided in Table 1.

3.2 Stakeholders of a Decentralized Application for Inter-Organizational Collaborations

The stakeholders are agents that play active roles in executing functions in a system [19]. In a collaborative blockchain system, stakeholders are peers, or agents that execute particular functions identified by the functional requirements of a proposed system. Broadly, the stakeholders can be classified into software agents and human actors [19]. Still, in blockchain-based systems, software agents are smart-contract actors that execute, or enforce the conditions outlined in a smart-contract code stored on a blockchain. Software agents may include a web-based API that provides interfaces for interacting with smart-contracts and blockchain data. These autonomous agents could also be represented as oracles to provides data for smart-contracts in a blockchain-based system. The project [39] is a good example of a decentralized oracle that delivers trusted data to a blockchain-based system.

The human actors are the participants, or parties involved in organizational collaborations. Different roles are assigned to the human actors based on the tasks (functions) they perform in a system. Since the functions are project-specific, it is not possible to automatically classify human actors in a collaborative process on a blockchain.

Table 1 Non-functional requirements mapping for blockchain DApps

Standard NFR	NFR Literature source	Explanation	Blockchain DApp for IOC context
Modifiable	IOC [29]	System changes and adapts during its lifecycle to the business context	Smart contract business collaboration lifecycle modifications
Integrable	IOC [29]	System consists of separately developed and integrated components for which there are interface protocols between the components	Blockchain DApp for business collaboration consists of separately developed and integrated components for which there are interface protocols between the components
Interoperable	IOC [29], Blockchain-DApp [5,33,34]	System must interoperate at runtime with information systems supporting other business functions	Digital assets stored in the blockchain is accessible in real-time to the collaborating parties
Secure (tamper-proof, privacy)	IOC [29], Blockchain-DApp [5,33]	The unauthorized access, or manipulation of digital assets stored in the system	Digital assets generated from business collaborations stored on a blockchain system are secured from unauthorized access and free from manipulation
High automation	IOC [29]	Function that is performed by a software agent	Function performed by a software agent in a DApp system that supports business collaborations
Flexible	IOC [29]	The ability of the system to allow diverse collaboration scenarios and permits the inter-organizational harmonization of heterogeneous concepts and technologies	Not applicable
Usable	IOC [29]	The learnability and the easiness of understanding the systems of various user functions	The learnability and the easiness of understanding the DApp collaborative systems' user functions by the participants
Scalable	IOC [29], Blockchain-DApp [34]	The system property to grow and manage an increased number of users	Create and manage an increased number of digital identities represented by users' public/private key pairs
Applicable	IOC [29]	Specifies for designers the ability to efficiently assess the quality and comprehensiveness of existing systems	Not applicable
Feasible	IOC [29]	The possibility of building a system is provable with the existence of application systems	Not applicable
Portable	IOC [29]	Ability of a system function to be executed on multiple classes of device types	Ability of the Dapp collaborative system functions to be executed on multiple classes of device types by the participants
Performance	IOC [29]	Amount of useful work done by the system	Useful work measured by the number of transactions recorded on the blockchain DApp for a period of time
Transparent (verifiable)	Blockchain-DApp [5,33]	System openness and the ability of the participants to view and verify stored records	The ability of the nodes participating in a business collaboration to verify transactions recorded on the blockchain network
Trustable	Blockchain-DApp [5,33]	System ability to behave consistently in expected ways and enforce the rules established by the network	Ability of a smart-contract to enforce the conditions specified in a smart-contract in business collaborations

3.3 Functional Requirements of a Decentralized Application for Inter-Organizational Collaborations

Function requirements outline specific behaviours of a software system [29]. Therefore, they are project-specific, which implies that each project, or system design has a unique set of functional requirements that describes the project. It is not possible to automate the steps required in outlining the functional requirements of a blockchain-based system for executing inter-organizational collaborations. We describe steps that an analyst can apply to systematically deduce the functional requirements of a system from a domain expert. Based on agent-oriented modelling (AOM), a goal model can be structured as follows: main goal, first-level goal, second-level goal, third-level until *n*th level goal [19]. In our approach, the main goal is referred to as the main value proposition that is the key objective a blockchain-based system provides.

The first step in modelling functional requirements is to identify the main value, or problem the system seeks to address and outline as the main value proposition. To identify the first-level requirements, all the possible functional requirements of the system are captured and then grouped in columns. Functional requirements are grouped together if they can be logically categorized as a sub-function of the topmost goal in the column. The first-level requirement represents the topmost goal of each column and the requirement that captures all the other (sub)requirements. The same approach can be used to further refine the requirements to the second requirement until the *n*th level requirement of a goal-model diagram that shows how detailed the requirement elicitation is.

The next step is to associate the non-functional requirements, i.e., quality-goals, and stakeholders to each goal. Since the goal models have hierarchical inheritance attributes [40], this implies that properties assigned to each functional requirement are inherited by the sub-requirements. Using a top-down approach, non-functional requirements and stakeholders are first assigned to the first-level, second-level, third-level and *n*th-level goal in that order. Based on the set of non-functional requirements already defined in Sect. 3.1 and specific stakeholders already identified as described in Sect. 3.2, the quality-goals and stakeholders are assigned to the functional requirements while ensuring that the assigned properties are applicable to the associated sub-goals.

3.4 AOM Goal Modelling Elements

Table 2 shows the elements used in building goal-model diagrams. The first row identifies the goal element representing the functional requirements. The second and third rows identify the quality-goal and emotional-goals representing the non-functional requirements. The fourth row shows the agent representing the stakeholders. The fourth and fifth rows show decomposition and association connections describing the relationships that exist between elements. Each of the elements are described with their graphical notations, followed by the specific properties associated with the elements. The last row shows the on-chain element that is used in representing goal functions executed on a blockchain (on-chain).

The tokenized on-chain function represents our extension of the standard AOM goal model to capture the novel requirements of blockchain systems [19]. A main purpose of the DApp-modelling framework is to address the cost scalability issue on blockchain systems [12]. Some of the NFR for IOC listed in Table 1 such as interoperable, transparent and trusted requires a goal function to be executed on-chain to be realized. These on-chain goals are then enclosed by the on-chain element. Therefore, scalability is achieved by executing only specific goals of a DApp on blockchain based on the quality goals attached to them.

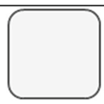
4 Goal Model of Decentralized Applications for Inter-Organizational Collaborations

This section shows the application of the requirement-modelling method developed in Sect. 3 to the running case provided in Sect. 2.3. A goal-model diagram developed from the running case depicts the requirements of DApps for IOC in KYC operations [28]. Several applications of the DAOM framework in modelling the requirements of blockchain DApps already exist in literature. Section 3 of the citation [41] shows the application of a DAOM-requirement diagram describing the functional, non-functional requirements and stakeholders in a DApp project that enables transparency and information symmetry in a collaborative construction project. Additionally, Section 3 of citation [42] shows the use of DAOM-framework concepts in describing the requirements of blockchain DApp for ensuring privacy and implementing user focused access control on user-generated data. Other applications of the DAOM requirement concepts can be found in [43,44]. Still, only the case in citation [28] is presented and syntactically evaluated in this paper.

4.1 Running Case Goal Models

The IdCredit platform is a blockchain system that seeks to address the inter-organizational collaborative issues identified in the running case described in Sect. 2.3. To outline and describe the requirements of the IdCredit platform, we employ the method described in Sect. 3 of this study. The main value proposition of the IdCredit system is to provide a decentralized blockchain system for the attestation and

Table 2 Descriptions of the DAOM graphical notation

Element	Description	Notation	Properties
Goal	A goal is a condition or state of affairs in the world that the stakeholders would like to achieve		1. Intended action words (or verbs) are used to textually notate a goal 2. A goal can have sub-goals showing the refinements 3. Additional elements such as quality goal, emotional goals and agents are used in describing the properties of a goal 4. A parallelogram is used in graphically representing the goal element
Quality goal	A quality goal defines specific criteria for achieving a goal		1. Descriptive terms (adjectives) are used to textually notate a quality goal 2. Several quality goals can be associated with a goal 3. Cloud symbol is used in graphically representing a quality goal
Emotional goal	An emotional goal defines the feelings of a user towards a particular function or attribute		1. Descriptive terms showing users feelings are used textually notate an emotional goal 2. Several emotional goals can be associated to a goal 3. An emotional goal is represented with a heart symbol
Agents	An agent is an actor that actively carries out actions to achieve goals by exercising its know-how		1. Identification words(nouns) are used in textually representing an agent 2. Several agents can be associated with a goal 3. Two classes of agents exist: Software and human agents 4. A UML actor symbol is used in graphically representing an agent
Decomposition	An agent is an actor that actively carries out actions to achieve goals by exercising its know-how		1. Decomposition shows the relationship between two goals: a parent goal and a sub-goal 2. A bold straight line is used in graphically representing a decomposition relationship
Association	An association relationship is used to link (associate) a goal with other elements that helps in describing the goal		1. Association shows the relationship between a goal and another goal modelling element 2. A broken straight line is used in graphically representing a decomposition relationship
On-chain	An on-chain element represents a goal function executed on the blockchain		1. Descriptive term token represents the digital asset for exchange of value in the decentralized network 2. The elements are attached to the on-chain goals 3. A tokenization element is symbolized by curved-edge square

authorization of digital assets. The main value of the system is refined into the first-level goal (FLG) of the system. These are as follows: manage key, on-board user, create a communication channel, store asset information, attest asset, manage transactions and manage blocks.

In Fig. 3, we show a reduced goal-model diagram of the IdCredit platform that comprises the root goal, first- and second-level goal refinements. The complete goal models of the system with further refinements of the FLGs are presented in citation [28]. The depicted quality goals of the main value proposition of the system are necessary for designing a system that satisfies the main value proposition of the IdCredit platform. These are as follows: scalability, security, transparency, trustworthiness, performance, usability, affordability, portability and high automation. In the context of the design of the IdCredit system, these quality goals are also defined in detail in [28].

The network stakeholders in the IdCredit system are represented as roles we identify as follows: donor, acceptor and identifier. The donors are the system users that provide personal data for KYC attestation. The acceptors are online merchants that require the donors' personal information to be attested before granting authorization for accessing a certain service and the identifiers validate and attest the donors' personal data for the acceptors. The role acceptor is associated with the store-asset information FLG, and the identifier is associated with the attest asset and manage blocks FLGs. All three roles are associated with the manage key and on-

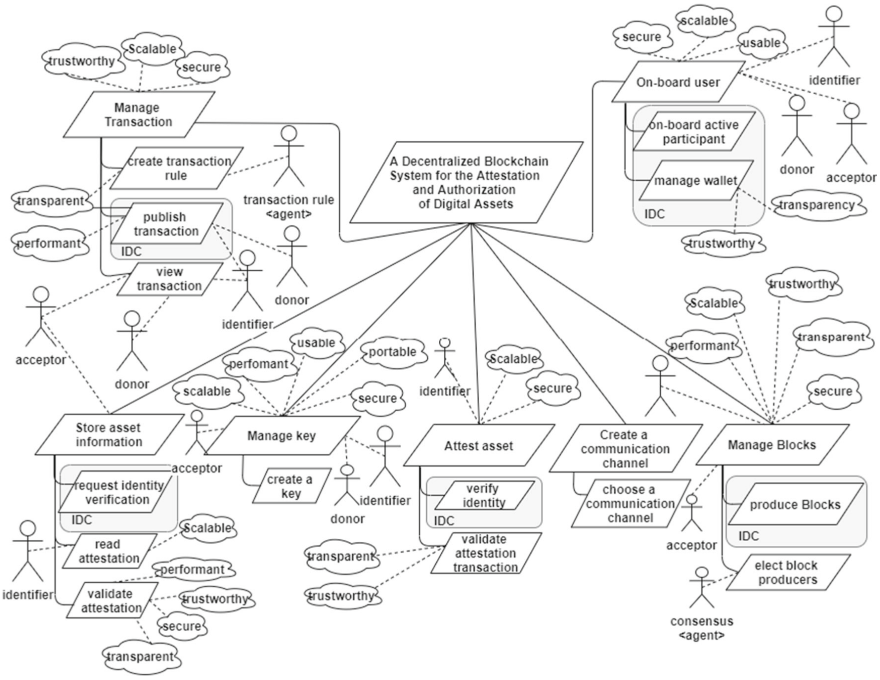


Fig. 3 IdCredit value proposition, first- and second-level goal model [28]

board user FLGs. Only active participants in the network are allowed to record attestation transactions on the blockchain. Active participants publish further service transactions such as identity attestation and request transactions in the last period.

The *manage key* function is a FLG that allow users of the IdCredit system to generate a key-pair comprising of the private and public keys for the identification purpose in using, or interacting with the IdCredit platform. To actively participate in the IdCredit system as an acceptor capable of submitting an identity-verification request, or as an identifier performing identity verification, the user needs to share a public key with other participants in the platform and set up a wallet that is enabled by the onboarding function. Generating the key pairs for the users does not require blockchain, and as a result, this goal is executed off-chain.

The *create communication channel* function allows the exchange of information between the participants of the IdCredit system to take place outside the blockchain system. There is no on-chain function associated with this element, showing that this DApp function is not executed on blockchain.

The function *store asset information* allows an acceptor to submit an identity verification request, or even reuse identity data that is already verified by the system. For an identity document to be accepted as a verified identity document by a service provider, the document must be attested, hashed and stored on the IdCredit blockchain that the function attest asset enables. The attestation, hashing and storing of the identity document are executed on-chain by the function 'request identity verification'.

The function *manage transactions* shows the transaction rule and possible transactions that are performed in the network. Once the blockchain is instantiated, a software agent creates the transaction rule that contains validation rules for all the transactions in the network. Once created, the transaction rules cannot be modified on the network. Transactions created on the platform are stored on-chain using the function 'publish transaction'. The published transactions can be viewed by all stakeholders in the network.

The function *manage block* shows the requirements for creating new blocks and selecting block producers on the network. Both the identifier and acceptor are capable of performing block-producing functions in the system. The blocks may contain any of the following transaction types, request verification, verify identity and token exchange transactions. The created blocks are stored on-chain using the function 'produce blocks'.

Table 3 Non-functional requirement mapping for blockchain DApps

	Model checking method	Notations	Model type	Formalization
[45]	If_else Conditional statement	Custom diagram	File system modelling	Set theory
[46]	XML Schema transformation	UML	Software requirement	–
[47]	PROMELA/LTL translation	UML	Sequence diagram	Set operators
[48]	PROMELA/LTL translation	Custom diagram	Security policy modelling	Set theory
[49]	LTL formulas	URM	User requirement diagram	Set theory
[50]	XML DTD Transformation	UML	Reverse engineering	Set theory
[51]	XML DTD Transformation	Telos	Data model	–
[52]	XML Schema Transformation	Custom diagram	Reverse engineering	Set theory

5 Syntax Description and Verification of Goal-Model Requirements of DApps for IOC

This section describes a method for verifying and validating the syntax of DAOM diagrams. First, we review existing model-validation methods in 5.1. The validation rules for the goal-diagram elements are presented in Sect. 5.2. The goal-model diagram developed in the previous section is transformed to XML elements in Sect. 5.3 and then formally verified.

5.1 Methods for Checking the Correctness of Models

Several methods exist in the literature for checking the correctness of models and their validations. We examined related articles to understand the existing model-checking methods and develop the most suitable approach for verifying the correctness of the goal model diagram developed in this paper. Table 3 shows the related articles for checking the correctness of models. The first column shows the article source and the second column shows the adopted correctness-checking method. The third column outlines the standard notations for the models. The fourth column shows the type of diagram the model represents and the last column identifies the mathematical method used to formally describe the models.

The result from Table 3 shows for XML transformations the PROMELA programming language and the linear temporal logic (LTL) are the common methods for checking and validating models. PROMELA is useful in checking concurrent systems to determine if a model produces the desired behaviour [47]. The LTL provides formulas for encoding model validation rules [53]. XML transformations provide structural elements for objects and their element compositions [54]. Both XML transformations and PROMELA are applied in validating custom diagrams as well as standard diagram notations such as UML. For the models that adopt XML transformations, either XML-schema or XML-DTD is used in the validation. All the articles formally describe their models using mathematical set theory, or set operators.

For the model-validation method developed in this paper, we adopt XML transformations because of being a common approach for model verification and it provides the possibility for manual transformations where a modelling-support tool is not yet available. Furthermore, we also adopt DTD over schema because it provides an additional layer of abstraction by separating the model-validation rule from the model object being validated. With such an approach, the validation rule developed in DTD can easily be exported, or implemented in a modelling-support tool. To formally describe the DAOM-requirement model, we adopt set theory in outlining the elements that comprise the DAOM diagrams and their structural compositions.

The goal model developed in this study follows the agent-oriented goal-modelling technique in study [19,40]. To the best of our knowledge, there exist no tools for verifying the correctness and consistency of AOM-based goal models. Therefore, to formally validate the artefact developed in Sect. 4.1, first we develop an ontology for describing goal-model diagrams and sets of constraints as validation rules for the diagram elements for which the element descriptions are provided in Sect. 3.4. We develop an XML document type definition (DTD) that incorporates the validation rules outlined in the AOM goal-modelling ontology. A goal-model diagram is transformed into respective XML elements using the developed ontology based on the validation rules in DTD. Finally, an XML application checks the correctness and consistency of the model.

5.2 AOM Goal Modelling Syntax Validation Rule Definitions

The XML DTD is a formal expression for defining structural components of an object [55]. We show in Table 4 a formal description of the goals for the modelling ontology using XML DTD and XPATH specifications. The XML DTD provides a legal building block of an XML document, and the

XPATH provides a query language for accessing elements in XML documents [56]. Similarly, study [46] shows the use of XML specifications for verifying software requirements. To formalize the goal-model diagram as well as the diagram elements, a set of validation rules are developed for each element represented as constraints in the second column of the table. A DTD-validation rule is derived for each set of the validation rule. An XPATH specification is also used for some constraints that cannot be verified using DTD. The last column of the table shows a formal description of each element.

For the first diagram element ’Diagram’ in the Table 4, five validation constraints are derived as shown in the second column of the table. All the validation constraints of the ’Diagram’ element can be verified using DTD specifications of the elements outlined in the third column of the table. The DTD constraint ’*’ attached to the diagram sub-elements *qualitygoal*, *emotionalgoal* and *agent* show that these types of elements can occur none, or multiple times in a valid goal-model diagram. The DTD constraint ’+’ attached to the *goal* sub-element shows that a valid diagram must have several occurrences of a goal element. The element diagram_value identifies the label of the diagram. The XPATH specification that starts with the bold part of the third column shows the location of the diagram sub-elements in the XML transformation. A formal description of the ’diagram’ element is derived in the last column of the table.

The second diagram element ’Goal’ comprises four constraints that are represented by the DTD sub-elements specifications *goal_value* showing the label of the goal and *refinement_level* showing if the goal element is first-, second-, third- or *n*th-level refinement. The *goal_id* attribute shows a unique identification of a goal element. The *decomposed_from* and *associated_to* attributes are straight lines and broken lines, respectively, that are connected to the goal elements and reference other elements in a diagram. The second constraint of the goal element: *a goal must have refinement, 0 refinement represents the root goal*, can only be verified using XPATH. Since only a single root goal must exist in a valid goal-model diagram, the XPATH query *//goal[refinement_level=0]* must return only one goal element. In addition, the XPATH specification of the goal element *//goal* shows the location of all goal elements in the transformed XML representation of the diagram. A formal description of the goal element is presented in the corresponding definition column.

The third and fourth diagram elements ’quality-goal’ and ’emotional-goal’ have single constraints that are represented by their values and unique identifications described using DTD specifications. The quality goal has a sub-element *qualitygoal_value* that identifies the label of the element and an ID attribute *qualitygoal_id* that provides a unique identification for quality-goal elements. The same applies to the emotional goal that has a sub-element *emotionalgoal_value* for identifying the label of the emotional-goal element and an ID attribute *emotionalgoal_id* provides a unique identification for emotional-goal elements. The XPATH specifications for the quality goals:*//qualitygoal* and emotional goals:*//emotionalgoal* show the locations of these elements in the transformed XML representation of the diagram. A formal description of the quality and emotional goals is also presented.

The last diagram element ’Agent’ in the Table 4 is defined by its label, unique identification and types. The DTD sub-element specification *agent_type* describes the type of an agent, and this can either be a software or human agent. The labels of the types are presented as readable data representing the actual name of the agent, and the ID attribute *agent_id* identifies each unique agent in the diagram. The XPATH specification *//agent* identifies all the agents in the transformed XML representation of the diagram. The formal description of the agent element is presented in the last column of the table.

Table 5 shows the relationship between goal elements and other elements in the goal-model diagram. The first column shows the element attributes that describe the relationship, followed by sets of constraints for each relationship attribute. The third column shows the DTD/XPATH specification for describing the relationship attributes, and the last column shows the formal description of the attributes.

The relationship attribute *decomposed_from*, representing a straight line, has two constraints that are realized using the ’ATTLIST’ and ’IDREFS’ properties of a goal element DTD specification. The first property ’ATTLIST’ shows that the decomposition attribute must be connected from a goal element. The second property ’IDREFS’ shows that the decomposition attribute must be connected to other IDs of corresponding elements. Since a goal must be decomposed from a parent goal, an XPATH query is used in verifying that all the IDs referenced by the attribute *decomposed_from* are goal-element IDs. Therefore, to verify that the goal elements are correctly connected in a goal-model diagram, the result of the XPATH query *//@decomposed_from*: element ID referenced by the decomposition attribute of goal elements must be a subset of the XPATH query result */diagram/goal/@goal_id*: goal IDs. A formal description of the relationship attribute *decomposed_from* is then derived.

The relationship attribute *associated_to*, representing a broken line, has two constraints that are also realized using the ’ATTLIST’ and ’IDREFS’ properties of a goal element DTD specification. The first property ’ATTLIST’ shows that the association attribute must be connected from a goal element. The second property ’IDREFS’ shows that the association attribute must be connected to another element’s ID. Since a goal must be associated with other element types, either to a quality goal or to an emotional goal, or to an agent,

Table 4 AOM goal-modelling syntax with element-validation rules

Element	Constraints	DTD and XPATH specification	Formal definition
Diagram	1. A goal model diagram must have a name	<!ELEMENT diagram (diagram_value,qualitygoal*, emotionalgoal*,agent*,goal+)> <!ELEMENT diagram'_value (#PCDATA)> ***diagram elements*** = *XPATH:* {'/*diagram*'}	$a, b, c, d, e \in A$ *where* $A = diagram$ $a = diagram\_value$ $b = quality\_goal$ $c = emotional\_goal$ $d = agent$ $e = goal$
	2. Must contain several goal elements		
	3. Can have none or several agents		
	4. Can have none or several quality goals		
	5. Can have none or several emotional goals		
Goal	1. A goal element must have a name and unique identification	<!ELEMENT goal (refinement_level,goal_value)> <!ELEMENT refinement_level (#PCDATA)> <!ELEMENT goal_value (#PCDATA)> <!ATTLIST goal goal_id ID #REQUIRED> <!ATTLIST goal decomposed_from IDREFS #IMPLIED> <!ATTLIST goal associated_to IDREFS #IMPLIED> **goal elements** = *XPATH* : {'//goal'} ***verify root goal exist*** = *XPATH:* {'//*goal[refinement_level=0]*'} ≡ *1*	$a, b \in A$ *where* $A = goal$ $a = refinement\_level$ $b = goal\_value$ $A = root\ goal \rightarrow a = 0$
	2. Must have a refinement level: 0 refinement represents the top level/root goal		
	3. Can be refined/decomposed from a parent goal		
	4. Can be associated with quality goals, emotional and agents		
Quality Goal	1. A quality goal must have a name and unique identification	<!ELEMENT qualitygoal (qualitygoal_value)> <!ELEMENT qualitygoal_value (#PCDATA)> <!ATTLIST qualitygoal qualitygoal_id ID #REQUIRED> ***qualitygoal elements*** = *XPATH:*{'//*qualitygoal*'}	$a \in A where\ A = qualitygoal$ $a = qualitygoal\_value$
Emotional goal	1. An emotional goal must have a name and unique identification	<!ELEMENT emotionalgoal (emotionalgoal_value)> <!ELEMENTemotionalgoal_value (#PCDATA)> <!ATTLIST emotionalgoal emotionalgoal_id ID #REQUIRED> ***emotionalgoal elements*** = *XPATH:*{'//*emotionalgoal*'}	$a \in A$ *where* $A = emotionalgoal$ $a = emotionalgoal\_value$
Agents	1. An agent must have a name and unique identification	<!ELEMENT agent (agent_type)> <!ELEMENT agent_type (software_agent|human_agent) > <!ELEMENT software_agent (#PCDATA)> <!ELEMENT human_agent (#PCDATA)> <!ATTLIST agent agent_id ID #REQUIRED> **agent elements** = XPATH:{'//*agent*'}	$a \in A$ *where* $A = agent$ $a = agent\_type$
	2. Can either be a software or human agent		$a = \{b \vee c\}$ *where* $b = software\ agent$ $c = human\ agent$

an XPATH query is used to verify that a goal element is correctly associated with the other element types. Therefore, to verify this, the result of the XPATH query *//@associated_to*: element ID referenced by the association attribute of goal elements must be a subset of the XPATH query results of either *//@qualitygoal_id*, or *//@emotionalgoal_id*, or *//@agent_id*. A formal description of the relationship attribute *associated_to* is then derived.

5.3 Running-Case Goal Model to XML Transformation

"Appendix A" provides the transformation of the goal model diagram into corresponding XML elements. The goal model diagram of the running case shown in Fig. 3 is transformed to XML elements using the XML DTD presented in Tables 4 and 5, respectively.

5.4 DAOM Goal Model Validation Results

This section provides the validation results of the XML elements generated in Sect. 5.3. First, a DTD-based validation is carried out, followed by XPATH validation, and the results are presented as follows.

5.4.1 DTD Validation of the Running-Case Diagram

Using an XML DTD validation-engine from http://xmlvalidator.new-studio.org/, the result shows that the XML representation of the running case diagram validates correctly.

5.4.2 XPATH Validation of the Running-Case Diagram

Table 6 shows the results of running XPATH queries on the XML-diagram transformation of the running case in Sect. 5.3. The first XPATH query shows that the diagram has only one root goal with the unique identity $g0$. The results of the second and third XPATH queries show that all the elements referenced by goal element attribute 'decomposed_from' are subsets of all the goal_ids in the diagram. The results of the fourth and fifth XPATH queries show that all the elements referenced by goal-element attribute 'associated_to' are the subsets of quality-goals, or emotional-goals, or agents.

5.4.3 Diagram Correctness Parameters and Results

By combining the XML-engine validation result and XML XPath-query results, we show that the goal model diagram developed in this study is consistent with the DAOM requirement-element syntax. Table 7 shows the correctness parameters used in validating the diagram. The rules check

Table 5 Syntactic validation for DAOM

Attributes	Constraints	DTD and XPATH specification	Formal definition
Decomposed_from	1. A decomposition attribute decomposes or refines a goal into sub-goal(s)	<!ATTLIST goal decomposed_from IDREFS #IMPLIED>	$A \subseteq B$ *where* $A =$ *decomposed elements id* $B =$ *goals id*
	2. A decomposition attribute must be connected from goal element to a parent goal element	verification of parent goals = XPATH : {//@decomposed_from} ⊆ {/diagram/goal/@goal_id}	
Associated_to	1. An association attribute associates or links a goal element to other element types	<!ATTLIST goal associated_to IDREFS #IMPLIED>	$A \subseteq B \vee C \vee D$ *where* $A =$ *associated elements id* $B =$ *quality goals id* $C =$ *agents id* $D =$ *emotional goals id*
	2. An association attribute must be connect from a goal to either a quality goal, emotional goal or agent	**verification of elements associated to goal** $elements = XPATH$: {//@$associated\_to$} ⊆ {//@qualitygoal_id | //@agent_id, of e | //@emotionalgoal_id }	

if the syntax is in accordance with the DAOM element syntax specified in Sect. 5 of this paper. The result shows that the diagram is composed of the correct elements where a root goal exits, goals are refined from parent goals, decomposition relationships are used in refining the goals, and association relationship is used in connecting the other elements to the goals. Since all the validation rules checked out correctly, we conclude that the goal model produced in this paper is a consistent and correct form of DAOM requirement diagram.

6 Discussions

This section provides a comparative discussion of the results from this study with other related research. The implications of the validation results are explained and compared with similar studies.

The DTD validation result of the idCredit goal-model XML transformation shows that the running case diagram in Fig. 3 contains valid AOM goal-modelling elements and the sub-elements of the main diagram elements are also valid AOM goals. The diagram elements and sub-elements are correctly used in the goal-model figure. The XPATH validation results show that the idCredit goal-model diagram contains valid root goal element. The element's relationship attribute 'decomposed_from' is correctly used in connecting goals with the parent goals. Also, the element's relationship attribute 'associated_to' is correctly used in connecting goal elements with other element types.

Although the goal model validates correctly with regard to the described modelling properties, the verification method outlined in this paper still has some weaknesses. The citation [19] shows one of the key properties of a goal-model diagram is the hierarchical-inheritance attribute of goal elements where a respective sub-goal automatically inherits the attributes (elements) associated with the parent goal. For instance, the quality goals, emotional goals and actors associated with the root goal are automatically assigned to the first-level goals. Still, it is not possible to verify this property with the method described in this study. Also, the transformation of goal diagrams to corresponding XML elements is manually carried out and results in several errors in a diagram with a high refinement level.

The study [57] describes a method for developing AOM goal-model diagrams. The result shows a modelling tool that can be accessed from AOM4STS[1]. The tool provides modelling elements and connections for building a goal-model diagram. Three connection types ensure that the goal element is correctly connected with other modelling elements. Still, it is not possible to verify the correctness of a goal-model diagram with the tool because of lacking verification means if a diagram contains the combinations of the correct elements. In our study, the correctness of element combinations in a goal diagram can be verified with the DTD specification of the diagram elements. The tool from the study [57] has additional limitations such as the inability to verify that a root goal exists, an inability to verify the uniqueness of diagram elements, and an inability to verify the two classes of agents such as software and human agents. In addition, an emotional diagram element is not captured by the tool. The results of this study show that all the verification limitations of the AOM4STS can be correctly verified with the AOM goal-model formalization method described in this study.

The study [46] shows how software requirements outlined in XML can be transformed into UML specifications since a framework exists for mapping XML documents to UML specifications and subsequently to software-requirement specifications. An XML-based schema is used in describing the elements and sub-element attributes and properties. Our study shows the mapping of a goal-model diagrams to XML to verify the correctness of the diagram using a DTD-based XML validation in describing the elements and sub-element attributes and properties. The main benefit of using DTD over XML schema in this research is that DTD provides an abstraction layer for separating validation rules and element transformation. In this case, we are able to separately define a validation rule and apply the same rule in any other XML document that contains the transformation of goal diagrams as XML elements. For this case, it is possible to validate the whole document (diagram), unlike in a schema where portions of the document can be specified and validated.

7 Conclusion

This study describes a model-driven method for developing requirements for blockchain-based inter-organizational collaborative systems and provides answer on how to model the requirements of a decentralized applications for inter-organizational collaboration. The first part of the study outlines a method for deriving functional, non-functional requirements and stakeholders of DApp for IOC. We also provide relevant non-functional requirements for describing DApps for IOC and furthermore provide the necessary classifications for actors and stakeholders involved in the inter-organizational collaborations.

In the second part of this study, we present the application of a requirement-derivation method by evaluating a running case and producing a goal-model diagram. The running case describes a blockchain-based system for the authorization and attestation of digital assets using blockchains. The result is a goal-model diagram showing the functional and non-functional requirements of the proposed system as well as the involved stakeholders and agents.

[1] AOM STS http://www.tud.ttu.ee/im/Msury.Mahunnah/AOM4STS/.

Table 6 XPATH query results

	XPATH Query	Result
1	diagram.xpath('//goal[refinement_level=0]')	{'goal_id': 'g0'}
2	diagram.xpath('//@decomposed_from')	['g0', 'g1 g0', 'g1 g0', 'g1 g0', 'g0', 'g2 g0', 'g2 g0', 'g2 g0', 'g0', 'g3 g0', 'g0', 'g4 g0', 'g4 g0', 'g0', 'g5 g0', 'g0', 'g6 g0', 'g6 g0', 'g0', 'g7 g0', 'g7 g0']
3	diagram.xpath('/diagram/goal/@goal_id')	['g0', 'g1', 'g1.1', 'g1.2', 'g1.3', 'g2', 'g2.1', 'g2.2', 'g2.3', 'g3', 'g3.1', 'g4', 'g4.1', 'g4.2', 'g5', 'g5.1', 'g6', 'g6.1', 'g6.2', 'g7', 'g7.1', 'g7.2']
4	diagram.xpath('//@associated_to')	['q1 q2 q3', 'a1 q4', 'q4 q5 a2 a3', 'a2 a3 a4', 'a4', 'q2 a3', 'q1 q3 q4 q5', 'q2 q3 q5 q6 q7 a2 a3 a4', 'q2 q3 a3', 'q1 q4', 'q1 q2 q3 q4 q5 a3 a4', 'a5', 'q2 q3 q6 a2 a3 a4', 'q1 q4']
5	diagram.xpath('//@qualitygoal_id | //@agent_id | //@emotionalgoal_id')	['q1', 'q2', 'q3', 'q4', 'q5', 'q6', 'q7', 'a1', 'a2', 'a3', 'a4', 'a5']

Table 7 Diagram correctness parameter and results

	DAOM requirement diagram Correctness parameter	Validation rule	Validation source	Result
1	Diagram contains more than one goal, may contain several quality goals, emotional goals and agents	<J ELEMENT diagram (diagram_value,qualitygoal*, emotionalgoal*,agent*,goal+)>	XML engine	TRUE
2	Only one root goal exists in the diagram	{'//goal[refinement_level=Q]'} = 1	Xpath query	TRUE
3	A goal is decomposed from a parent goal	(I//@decomposed_froml) = goal_id	XPath query/XML engine	TRUE
4	All elements connected with decomposition relationship are goal elements	{//@decomposed_from} 5 = {/diagram/goal/@goal_id}	Xpath query	TRUE
5	All elements connected with association relationship are quality goals, emotional goals or agents	{//@associated_to} C {//@qualitygoal_id// @agent_id | //@emotionalgoal_id }	Xpath query	TRUE

In the final part of this study, we show formal descriptions of the goal-model diagram elements as well as the verification of a goal-model diagram. The diagram generated in the running-case evaluation is formally verified for correctness and consistency of diagram elements. An XML-based ontology is developed for describing goal-model diagrams, and validation rules for the diagram elements are derived using XML DTD and XPATH specifications. The formal descriptions of the diagram elements are also derived. The goal-model diagram developed from the running case is transformed to XML elements and then validated for correctness and consistency using a XML DTD validation engine and XPATH queries. The results show that the artefact generated in using the requirement-elicitation method described in this study is correct and consistent.

One of the main limitations of this study is that the novel model-formalization technique developed in this study has been tested with a single case. In addition, the XML-transformation technique and the syntax validation described in this study is filled with manual processes that are prone to errors.

Based on the findings of this research as well as the acknowledged research limitations, we recommend the following future work. First, we aim to develop a DAOM support tool for modelling DApps that facilitates the modelling of DApps using the DAOM-diagram elements described in this paper and provides automatic syntax validation of the elements using the validation rules developed in this paper. The enterprise architect software development kit [2] provides clear documentation for the implementation of such tools. Furthermore, to address the single case issue, the developed support tool must be evaluated with multiple cases of DApps for IOC.

[2] Enterprise Architect SDK: https://sparxsystems.com/enterprise_architect_user_guide/14.0/modeling_tools/introduction_2.html.

Appendix

A idCredit Goal-Model XML


```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE diagram [
<!ELEMENT diagram (diagram_value,qualitygoal*,
emotionalgoal*,agent*,goal+)>
<!ELEMENT diagram_value (#PCDATA)>
<!ELEMENT qualitygoal (qualitygoal_value)>
<!ELEMENT qualitygoal_value (#PCDATA)>
<!ATTLIST qualitygoal qualitygoal_id ID #REQUIRED>
<!ELEMENT emotionalgoal (emotionalgoal_value)>
<!ELEMENT emotionalgoal_value (#PCDATA)>
<!ATTLIST emotionalgoal emotionalgoal_id ID #REQUIRED>
<!ELEMENT agent (agent_type)>
<!ELEMENT agent_type (software_agent|human_agent)>
<!ELEMENT software_agent (#PCDATA)>
<!ELEMENT human_agent (#PCDATA)>
<!ATTLIST agent agent_id ID #REQUIRED>
<!ELEMENT goal (refinement_level,goal_value)>
<!ELEMENT refinement_level (#PCDATA)>
<!ELEMENT goal_value (#PCDATA)>
<!ATTLIST goal goal_id ID #REQUIRED>
<!ATTLIST goal decomposed_from IDREFS #IMPLIED>
<!ATTLIST goal associated_to IDREFS #IMPLIED>
]>
<diagram>
<diagram_value>IdCredit value proposition,
first- and second level goal model</diagram_value>

<!--list of quality-goals in the diagram-->
<qualitygoal qualitygoal_id="q1">
<qualitygoal_value>trustworthy</qualitygoal_value>
</qualitygoal>
<qualitygoal qualitygoal_id="q2">
<qualitygoal_value>scalable</qualitygoal_value>
</qualitygoal>
<qualitygoal qualitygoal_id="q3">
<qualitygoal_value>secure</qualitygoal_value>
</qualitygoal>
<qualitygoal qualitygoal_id="q4">
<qualitygoal_value>transparent</qualitygoal_value>
</qualitygoal>
<qualitygoal qualitygoal_id="q5">
<qualitygoal_value>performant</qualitygoal_value>
</qualitygoal>
<qualitygoal qualitygoal_id="q6">
<qualitygoal_value>usable</qualitygoal_value>
</qualitygoal>
<qualitygoal qualitygoal_id="q7">
<qualitygoal_value>portable</qualitygoal_value>
</qualitygoal>

<!--list of agents in the diagram-->
<agent agent_id="a1">
<agent_type >
<software_agent>transaction rule</software_agent>
</agent_type>
</agent>
<agent agent_id="a2">
<agent_type >
<human_agent>donor</human_agent>
</agent_type>
</agent>
<agent agent_id="a3">
<agent_type >
<human_agent>identifier</human_agent>
</agent_type>
</agent>
<agent agent_id="a4">
<agent_type >
<human_agent>acceptor</human_agent>
</agent_type>
</agent>
<agent agent_id="a5">
<agent_type >
<software_agent>consensus</software_agent>
</agent_type>
</agent>

<!--root goal or value proposition
and associations-->
<goal goal_id="g0">
<refinement_level>0</refinement_level>
<goal_value>A Decentralized Blockchain System for
the Attestation and Authorization of
Digital Assets</goal_value>
</goal>

<!--No1 first level refinements and sub goals-->
<goal goal_id="g1" decomposed_from="g0"
associated_to="q1 q2 q3">
<refinement_level>1</refinement_level>
<goal_value>manage transaction</goal_value>
</goal>
<goal goal_id="g1.1" decomposed_from= "g1 g0"
associated_to="a1 q4">
<refinement_level>2</refinement_level>
<goal_value>create transaction rule</goal_value>
</goal>
<goal goal_id="g1.2" decomposed_from= "g1 g0"
associated_to="q4 q5 a2 a3">
<refinement_level>2</refinement_level>
<goal_value>publish transaction</goal_value>
</goal>
<goal goal_id="g1.3" decomposed_from= "g1 g0"
associated_to="a2 a3 a4">
<refinement_level>2</refinement_level>
<goal_value>view transaction</goal_value>
</goal>
<!--No2 first level refinements and sub goals-->
<goal goal_id="g2" decomposed_from="g0"
associated_to="a4">
<refinement_level>1</refinement_level>
<goal_value>store asset information</goal_value>
</goal>
<goal goal_id="g2.1" decomposed_from= "g2 g0">
<refinement_level>2</refinement_level>
<goal_value>request identity verification</goal_value>
</goal>
<goal goal_id="g2.2" decomposed_from= "g2 g0"
associated_to="q2 a3">
<refinement_level>2</refinement_level>
<goal_value>read attestation</goal_value>
</goal>

<goal goal_id="g2.3" decomposed_from= "g2 g0"
associated_to="q1 q3 q4 q5">
<refinement_level>2</refinement_level>
<goal_value>validate attestation</goal_value>
</goal>
<!--No3 first level refinements and sub goals-->
<goal goal_id="g3" decomposed_from="g0"
associated_to="q2 q3 q5 q6 q7 a2 a3 a4">
<refinement_level>1</refinement_level>
<goal_value>manage key</goal_value>
</goal>
<goal goal_id="g3.1" decomposed_from= "g3 g0">
<refinement_level>2</refinement_level>
<goal_value>create key</goal_value>
</goal>
<!--No4 first level refinements and sub goals-->
<goal goal_id="g4" decomposed_from="g0"
associated_to="q2 q3 a3">
<refinement_level>1</refinement_level>
<goal_value>attest asset</goal_value>
</goal>
<goal goal_id="g4.1" decomposed_from= "g4 g0">
<refinement_level>2</refinement_level>
<goal_value>verify identity</goal_value>
</goal>
<goal goal_id="g4.2" decomposed_from= "g4 g0"
associated_to="q1 q4">
<refinement_level>2</refinement_level>
<goal_value>validate attestation transaction
</goal_value>
</goal>

<!--No5 first level refinements and sub goals-->
```

```
<goal goal_id="g5" decomposed_from="g0">
<refinement_level>1</refinement_level>
<goal_value>create a communication channel</goal_value>
</goal>
<goal goal_id="g5.1" decomposed_from= "g5 g0">
<refinement_level>2</refinement_level>
<goal_value>choose a communication channel</goal_value>
</goal>
<!--No6 first level refinements and sub goals-->
<goal goal_id="g6" decomposed_from="g0"
associated_to="q1 q2 q3 q4 q5 a3 a4">
<refinement_level>1</refinement_level>
<goal_value>manage blocks</goal_value>
</goal>
<goal goal_id="g6.1" decomposed_from= "g6 g0">
<refinement_level>2</refinement_level>
<goal_value>produce blocks</goal_value>
</goal>
<goal goal_id="g6.2" decomposed_from= "g6 g0"
associated_to="a5">
<refinement_level>2</refinement_level>
<goal_value>elect block producers</goal_value>
</goal>
<!--No7 first level refinements and sub goals-->
<goal goal_id="g7" decomposed_from="g0"
associated_to="q2 q3 q6 a2 a3 a4">
<refinement_level>1</refinement_level>
<goal_value>onboard user</goal_value>
</goal>
<goal goal_id="g7.1" decomposed_from= "g7 g0">
<refinement_level>2</refinement_level>
<goal_value>onboard active participant</goal_value>
</goal>
<goal goal_id="g7.2" decomposed_from= "g7 g0"
associated_to="q1 q4">
<refinement_level>2</refinement_level>
<goal_value>manage wallet</goal_value>
</goal>
</diagram>
```


References

1. Bouchbout, K.; Alimazighi, Z.: Inter-organizational business processes modelling framework. In: ADBIS (2), pp. 45–54. Citeseer, (2011)
2. Matulevičius, R.; Norta, A.; Udokwu, C.; Nõukas, R.: Assessment of aviation security risk management for airline turnaround processes. In: Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXVI, pp. 109–141. Springer (2017)
3. Draheim, D.: Blockchain technology for enabling transparent and traceable government collaboration in public project processes of developing economies. In: Digital Transformation for a Sustainable Society in the 21st Century: 18th IFIP WG 6.11 Conference on e-Business, e-Services, and e-Society, I3E 2019, Trondheim, Norway, September 18–20, 2019, Proceedings, pp. 464. Springer Nature (2019)
4. Kormiltsyn, A.; Udokwu, C.; Karu, K.; Thangalimodzi, K.; Norta, A.: Improving healthcare processes with smart contracts. In: International Conference on Business Information Systems, pp. 500–513. Springer (2019)
5. Udokwu, C.; Kormiltsyn, A.; Thangalimodzi, K.; Norta, A.: The state of the art for blockchain-enabled smart-contract applications in the organization. In: 2018 Ivannikov Ispras Open Conference (ISPRAS), pp. 137–144. IEEE (2018)
6. Bashir, I.: Mastering Blockchain: Distributed Ledger Technology decentralization, and smart contracts explained. Packt Publishing Ltd, Birmingham (2018)
7. Swan, M.: Blockchain: Blueprint for a new economy. O'Reilly Media Inc, Newton (2015)
8. Cai, W.; Wang, Z.; Ernst, J.B.; Hong, Z.; Feng, C.; Leung, V.C.: Decentralized applications: the blockchain-empowered software system. IEEE Access **6**, 53019–53033 (2018)
9. Salah, K.; Nizamuddin, N.; Jayaraman, R.; Omar, M.: Blockchain-based soybean traceability in agricultural supply chain. IEEE Access **7**, 73295–73305 (2019)
10. Hasan, H.R.; Salah, K.: Proof of delivery of digital assets using blockchain and smart contracts. IEEE Access **6**, 65439–65448 (2018)
11. Hasan, H.R.; Salah, K.: Blockchain-based solution for proof of delivery of physical assets. In: International Conference on Blockchain, pp. 139–152. Springer (2018)
12. Udokwu, C.; Anyanka, H.; Norta, A.: Evaluation of approaches for designing and developing decentralized applications
13. Kim, H.M.; Laskowski, M.: Toward an ontology-driven blockchain design for supply-chain provenance. Intell. Syst. Account. Finance Manag. **25**(1), 18–27 (2018)
14. García-Bañuelos, L.; Ponomarev, A.; Dumas, M.; Weber, I.: Optimized execution of business processes on blockchain. In: International Conference on Business Process Management, pp. 130–146. Springer (2017)
15. Dagher, G.G.; Mohler, J.; Milojkovic, M.; Marella, P.B.: Ancile: privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. Sustain. Cities Soc. **39**, 283–297 (2018)
16. Giorgini, P.; Mylopoulos, J.; Sebastiani, R.: Goal-oriented requirements analysis and reasoning in the tropos methodology. Eng. Appl. Artif. Intell. **18**(2), 159–171 (2005)
17. Chopra, A.K.; Dalpiaz, F.; Giorgini, P.; Mylopoulos, J.: Modeling and reasoning about service-oriented applications via goals and commitments. In: International Conference on Advanced Information Systems Engineering, pp. 113–128. Springer, Berlin (2010)
18. Andersson, B.; Johannesson, P.; Zdravkovic, J.: Aligning goals and services through goal and business modelling. Inf. Syst. E-Bus. Manag. **7**(2), 143–169 (2009)
19. Sterling, L.; Taveter, K.: The art of agent-oriented modeling. MIT Press, Cambridge (2009)
20. Bartoletti, M.; Pompianu, L.: An empirical analysis of smart contracts: platforms, applications, and design patterns. In: International Conference on Financial Cryptography and Data Security, pp. 494–509. Springer, Berlin (2017)
21. Jain, A.; Arora, S.; Shukla, Y.; Patil, T.; Sawant-Patil, S.: Proof of stake with casper the friendly finality gadget protocol for fair validation consensus in ethereum. Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol. **3**(3), 291–298 (2018)
22. Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H.: An overview of blockchain technology: architecture, consensus, and future trends. In: 2017 IEEE International Congress on Big Data (BigData Congress), pp. 557–564. IEEE (2017)
23. Pilkington, M.: Blockchain Technology: Principles and Applications. Research handbook on digital transformations. Edward Elgar Publishing, Cheltenham (2016)
24. Croman, K.; Decker, C.; Eyal, I.; Gencer, A.E.; Juels, A.; Kosba, A.; Miller, A.; Saxena, P.; Shi, E.; Sirer, E.G.; et al.: On scaling decentralized blockchains. In: International Conference on Financial Cryptography and Data Security, pp. 106–125. Springer, Heidelberg (2016)
25. Dannen, C.: Introducing Ethereum and Solidity. Springer, Switzerland (2017)
26. López-Pintado, O.; García-Bañuelos, L.; Dumas, M.; Weber, I.: Caterpillar: a blockchain-based business process management system. In: BPM (Demos) (2017)
27. Von Alan, R.H.; March, S.T.; Park, J.; Ram, S.: Design science in information systems research. MIS Q. **28**(1), 75–105 (2004)
28. Balahontsev, V.; Tsikhilov, A.; Norta, A.; Udokwu, C.: A blockchain system for the attestation and authorization of digital

assets version 1.0 (2019). https://doi.org/10.13140/RG.2.2.25027.96807/1

29. Norta, A.; Grefen, P.; Narendra, N.C.: A reference architecture for managing dynamic inter-organizational business processes. Data Knowl. Eng. **91**, 52–89 (2014)
30. Cleland-Huang, J.; Settimi, R.; Zou, X.; Solc, P.: Automated classification of non-functional requirements. Requir. Eng. **12**(2), 103–120 (2007)
31. Cleland-Huang, J.; Settimi, R.; BenKhadra, O.; Berezhanskaya, E.; Christina, S.: Goal-centric traceability for managing non-functional requirements. In: Proceedings 27th International Conference on Software Engineering, 2005. ICSE 2005, pp. 362–371. IEEE (2005)
32. Korpela, K.; Mikkonen, K.; Hallikas, J.; Pynnönen, M.: Digital business ecosystem transformation–towards cloud integration. In: 2016 49th Hawaii International Conference on System Sciences (HICSS), pp. 3959–3968. IEEE (2016)
33. Irannezhad, E.: The architectural design requirements of a blockchain-based port community system. Logistics **4**(4), 30 (2020)
34. Zhang, P.; White, J.; Schmidt, D.C.; Lenz, G.: Design of blockchain-based apps using familiar software patterns with a healthcare focus. In: Proceedings of the 24th Conference on Pattern Languages of Programs, pp. 1–14 (2017)
35. Cysneiros, L.M.; do Prado Leite, J.C.S.: Nonfunctional requirements: from elicitation to conceptual models. IEEE Trans. Softw. Eng. **30**(5), 328–350 (2004)
36. Marchesi, M.; Marchesi, L.; Tonelli, R.: An agile software engineering method to design blockchain applications. In: Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia, pp. 1–8 (2018)
37. Kondo, Y.: Smart contract lifecycle management. US Patent App. 15/626,236 (2018)
38. Norta, A.: Designing a smart-contract application layer for transacting decentralized autonomous organizations. In: International Conference on Advances in Computing and Data Sciences, pp. 595–604. Springer (2016)
39. Ellis, S.; Juels, A.; Sergey.: A decentralized oracle network Nazarov (2017). https://www.chainwhy.com/upload/zb_users/upload/2018/05/201805071525668312734700.pdf. Accessed 22 Jan 2021
40. Tenso, T.; Norta, H.; Rootsi, H.; Taveter, K.; Vorontsova, I.: Enhancing requirements engineering in agile methodologies by agent-oriented goal models. In: 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW) (2017)
41. Norta, A.; Wenna, C.; Udokwu, C.: Designing a collaborative construction-project platform on blockchain technology for transparancy, traceablity and information symmetry (2020). https://doi.org/10.13140/RG.2.2.17356.64644
42. Norta, A.; Hawthorne, D.; Engel, S.L.: A privacy-protecting data-exchange wallet with ownership-and monetization capabilities. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2018)
43. Norta, A.; Rossar, R.; Parve, M.; Laas-Billson, L.: Achieving a high level of open market-information symmetry with decentralised insurance marketplaces on blockchains. In: Intelligent Computing-Proceedings of the Computing Conference, pp. 299–318. Springer (2019)
44. Dai, P.; Mahi, N.; Earls, J.; Norta, A.: Smart-contract value-transfer protocols on a distributed mobile application platform. Qtum Foundation, Singapore (2017). https://doi.org/10.13140/RG.2.2.35140.63365
45. Yang, J.; Twohey, P.; Engler, D.; Musuvathi, M.: Using model checking to find serious file system errors. ACM Trans. Comput. Syst. (TOCS) **24**(4), 393–423 (2006)
46. Aman, H.; Ibrahim, R.: Reverse engineering: from xml to uml for generation of software requirement specification. In: 2013 8th International Conference on Information Technology in Asia (CITA), pp. 1–6. IEEE (2013)
47. Lima, V.; Talhi, C.; Mouheb, D.; Debbabi, M.; Wang, L.; Pourzandi, M.: Formal verification and validation of uml 2.0 sequence diagrams using source and destination of messages. Electron. Notes Theor. Comput. Sci. **254**, 143–160 (2009)
48. Ma, J.; Zhang, D.; Xu, G.; Yang, Y.: Model checking based security policy verification and validation. In: 2010 2nd International Workshop on Intelligent Systems and Applications, pp. 1–4. IEEE (2010)
49. Souri, A.; Nourozi, M.; Rahmani, A.M.; Navimipour, N.J.: A model checking approach for user relationship management in the social network. Kybernetes **48**(3), 407–423 (2019). https://doi.org/10.1108/K-02-2018-0092
50. Weidong, Y.; Ning, G.; Baile, S.: Reverse engineering xml. In: First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06), vol. 2, pp. 447–454. IEEE (2006)
51. Rodríguez-Gianolli, P.; Mylopoulos, J.: A semantic approach to xml-based data integration. In: International Conference on Conceptual Modeling, pp. 117–132. Springer (2001)
52. Fong, J.; Cheung, S.K.; Shiu, H.: The xml tree model-toward an xml conceptual schema reversed from xml schema definition. Data Knowl. Eng. **64**(3), 624–661 (2008)
53. Huth, M.; Ryan, M.: Logic in Computer Science: Modelling and Reasoning About Systems. Cambridge University Press, Cambridge (2004)
54. Conrad, R.; Scheffner, D.; Freytag, J.C.: Xml conceptual modeling using uml. In: International Conference on Conceptual Modeling, pp. 558–571. Springer, Heidelberg (2000)
55. Calvanese, D.; De Giacomo, G.; Lenzerini, M.: Representing and reasoning on xml documents: a description logic approach. J. Logic Comput. **9**(3), 295–318 (1999)
56. Papakonstantinou, Y.; Vianu, V.: DTD inference for views of xml data. In: Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of database systems, pp. 35–46. ACM (2000)
57. Mahunnah, M.; Taveter, K.; Matulevičius, R.: An empirical evaluation of the requirements engineering tool for socio-technical systems. In: 2018 IEEE 7th International Workshop on Empirical Requirements Engineering (EmpiRE), pp. 8–15. IEEE (2018)

Publication IV

Udokwu, C., Norta, A., and Wenna, C.

Designing a Collaborative Construction-Project Platform on Blockchain Technology for Transparency, Traceability, and Information Symmetry

Reprinted with permission from
Asia service sciences and software engineering conference

Designing a Collaborative Construction-Project Platform on Blockchain Technology for Transparency, Traceability, and Information Symmetry

Chibuzor, j, Udokwu
Department of Software Science,
Tallinn University of Technology,
Estonia
chibuzor.udokwu@taltech.ee

Alexander Norta
Department of Software Science,
Tallinn University of Technology,
Estonia
alexander.norta@taltech.ee

Christoph Wenna
Wenna Consult GmbH, Linz, Austria
cw@wenna.com

ABSTRACT

The construction industry is a $6 trillion industry worldwide with a prediction to grow towards $10,3 trillion by 2023 and constitutes an essential part of the global economy. Nevertheless, the management of the construction effort is still very manual. The construction process from design, to sourcing of material, contract management, and so on, is a convoluted and intransparent process filled with risks the collaborating parties are exposed to. A need exists for management platforms that streamline and automate collaborative construction processes, establish transparency, traceability, and information symmetry between business parties. This paper presents the Construction Project Management (CoPM) platform that is based on blockchain- and smart-contract technologies for enabling peer-to-peer collaboration between construction parties that enhances the flow of information for reducing cost- and time expenditures while improving the quality of service. The CoPM system is based on diligent up-front requirement studies from which we derive a coherent system architecture and set of cooperation protocols. Thereby, the CoPM system overcomes the currently existing fractured value propositions for construction-management systems.

CCS CONCEPTS

• Software and its engineering; **• Software creation and management**; **• Designing software**;

KEYWORDS

Construction, project management, blockchain, smart contract, collaboration, business processes, inter-organisational, data logistics

ACM Reference Format:
Chibuzor, j, Udokwu, Alexander Norta, and Christoph Wenna. 2021. Designing a Collaborative Construction-Project Platform on Blockchain Technology for Transparency, Traceability, and Information Symmetry. In *2021 2nd Asia Service Sciences and Software Engineering Conference (ASSE '21), February 24–26, 2021, Macau, Macao.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3456126.3456134

ASSE '21, February 24–26, 2021, Macau, Macao

ACM ISBN 978-1-4503-8908-2/21/02…$15.00
https://doi.org/10.1145/3456126.3456134

1 INTRODUCTION

There is a strong demand for restructuring construction processes while in reality, the management of construction projects still lacks the required attention concerning meaningful automation and thus, this industry is one of the least digitised industries. More specifically, the construction industry suffers from a lack of investment in innovation that results in limited collaboration sophistication with structural fragmentation [13]. The lack of a collaboration-automation context results in hardly any of the available data in the construction industry to be utilized [35], thus, knowledge management is currently dysfunctional. The research and development investment is currently less than 1% of net sales [36], causing labour-productivity decline over the last decades [25].

Infrastructural development across nations involves massive construction- and building projects. Due to corruption and a lack of proper information management in the construction industry, many large-scale construction projects have failed, or performed poorly below expectations [1], [33]. The study [2] shows that digitizing the processes that support large-scale constructions project management, addresses some of the corruption issues experienced in the construction industry. Still, traditional information systems for project management do not provide the necessary data visibility, transparency and information security for managing complex inter-organizational collaborations that exist in large-scale construction projects. The blockchain-enabled smart contracts potentially address these problems by providing a decentralized network for trusted, tamper-proof, and secure exchanges of data in multi-party project executions experienced in large-scale constructions [29].

Smart contracts [7] are computer protocols that automate the enforcement of machine-readable contracts to allow for trackable- and irreversible transactions that render trusted third parties unnecessary. Blockchain technology [24] achieves these trackable- and irreversible transactions with the first use case being the crypto-currency bitcoin [19]. Still, smart contracts yield the option to establish agile collaboration configurations between decentralised autonomous organisations (DAO) [21] that comprise loosely coupled process-aware and inter-organisational exchange interfaces that protect internal process privacy. The digital assets that drive the internal processes of such autonomous organizations are referred to as tokens. Digital assets in the construction industry are building objects and models used for representing and communicating the requirements of a building project [18]. In building projects, the tender specification shows detailed requirements of the building project and these are represented as building models being digital assets [16].

The state-of-the-art shows a recognized need for construction-collaboration automation to inter-organizationally integrate processes along which the right type of information can flow from the right sources to the right sinks at the right time. However, we find a gap with respect to the available automation systems being not well studied with respect to the complete requirement sets that should be satisfied for construction-collaboration systems. Furthermore, the technology stack such construction-collaboration systems are based on, are outdated and inappropriately assembled, given the actual requirement sets. This paper fills the gap by posing the question of how to design automation systems that address the actual requirements for achieving high-quality construction-project collaborations without fractures and frictions causing unnecessary time delays and cost increases? To establish a separation of concerns, we deduce the following sub-questions: What are the sets of requirements that include and assign also the diverse stakeholders who collaborate? What is the architecture topology of a requirements satisfying construction-collaboration system? What are the dynamic exchange protocols between stakeholders and system components into which embedded on-chain transactions assure immutable traceability? Consequently, we introduce the conceptual model of the CoPM system that is a blockchain-technology employing distributed applications (DApp).

The remainder of this paper is structured as follows. Section 2 captures the AS-IS state of the construction industry and provides the method for this study. Section 3 arranges the requirement sets for a construction-collaboration system into a logical space that allows for a targeted association of key stakeholders. Next, Section 4 deduces a distributed architecture from the logical space of system requirements. Section 5 then detects key on-chain transactions that are system relevant for potential litigation situations and positions these on-chain transactions into dynamic exchange protocols that occur between stakeholders and the architecture of the construction-collaboration system. Section 6 specifies a blockchain-technology stack of pre-existing applications that can be assembled for rapid deployment of the CoPM construction-collaboration system and also discusses relevant projects. Finally, Section 7 concludes this paper and also discusses limitations, open issues and future work.

2 PRELIMINARIES

Section 2.1 gives a running case that captures the AS-IS state of deficiencies of the construction industry. Next, Section 2.2 describes the method used in designing a proposed blockchain application CoPM that addresses the problems presented in the running case.

2.1 Running Case

This article uses as a running case the construction disaster of the new Berlin Brandenburg international airport as described by [31]. Accordingly, the numerous problems that have arisen with the construction of the new Berlin Brandenburg international airport are representative examples within construction projects that can be overcome using the CoPM platform. The study [32] proposes a collective and participatory governance framework to address the complex inter-organizational problems experienced during large-scale construction. Such a legal framework can easily be outlined and enforced by smart-contracts on the CoPM. From a technical- and management perspective, the problems of the new Berlin Brandenburg international airport are mainly caused by a lack of collaboration-automation, distrust between collaborators and inefficient knowledge management among the project participants. The long planning period and the large number of subcontractors employed results in a high level of complexity. The complete representation of the running case is in [22].

2.2 Research method

This paper applies the methods and concepts in the decentralized agent-oriented modelling (DAOM) framework in describing a blockchain application that addresses the problem of information and project management in complex building construction. The DAOM framework provides a model-driven design framework for designing and developing blockchain applications [27, 28]. The DAOM framework provides a systematic approach for building a blockchain application by providing the models for the application requirements, -architecture and -usecases.

3 COPM-REQUIREMENTS

In this section, we present the stakeholders and their associated requirements in the decentralized CoPM system. The DAOM requirement diagram is used in outlining the functional requirements and the associated non-functional requirements and assigned roles in the CoPM platform.

3.1 Stakeholders Associated to CoPM-Requirements

The stakeholders in the CoPM platform are system participants that perform different functions. Based on expert knowledge, we identify the following classes of users for CoPM. *The designers, contractors, auditors, project owners (project managers), original equipment manufacturers (OEM), admin and software agents.* The designers are the architects, planners, engineers, and specifiers that perform goals that result in the creation of digital assets on CoPM. The contractors are the engineers responsible for the day-to-day construction activity of a building project. Auditors are certified individuals that perform checks to verify that a construction work meets the required-, or stated standards. The project owners and managers represent the individuals responsible for coordinating the building projects. The OEM produces physical building objects used by contractors. They also produce digital assets such as buidling information management (BIM) objects used by designers in creating corresponding building models. The admins are the platform administrators responsible for onboarding new users. The software agents represent the autonomous and semi-autonomous agents. They include *IoT agent, a know-your-customer (KCY) agent and auditor agent.* The IoT agents record and transmit the status of building projects. The KYC agents perform a verification of identity and other documents submitted by new users. The auditor agent

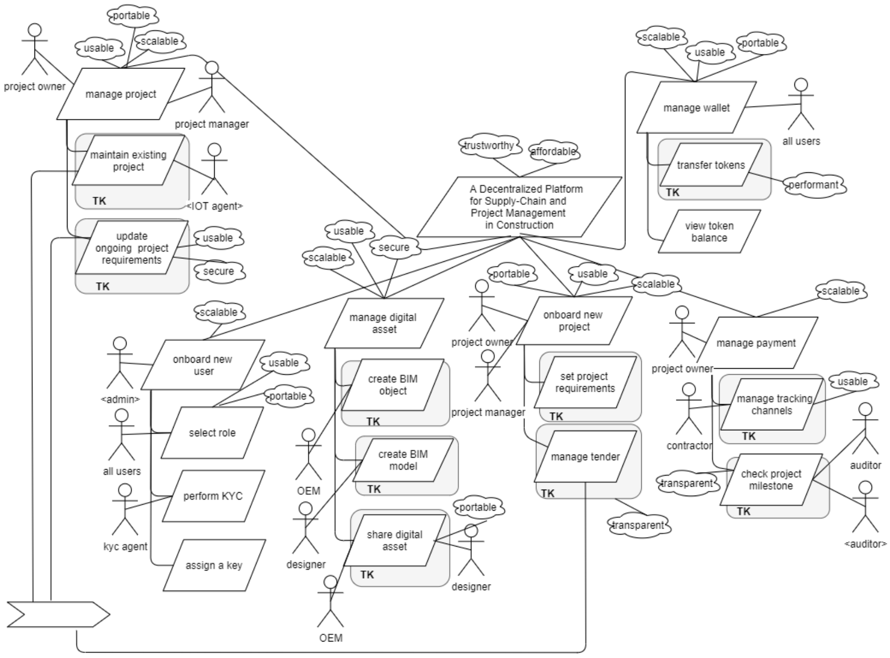


Figure 1: CoPM-system value proposition and the first-level-goal (FLG) model.

along with human auditor verify the status of building projects in a semi-autonomous way.

3.2 System-Value Proposition

The main value proposition of CoPM is to provide a decentralized platform for supply-chain- and project management in the construction sector. Figure 1 shows a reduced version of the goal model containing the requirements of the CoPM platform. The complete system requirement is available in [22]. The main value propositions of the platform are as follows, *onboard new user, manage digital asset, onboard new project, manage project, manage payment and manage wallet.* The *onboard new user* goal enables the execution of KYC-related goals and the assignment of roles to the users. The *manage digital asset* goal enables the users to create digital assets such as BIM objects and models and then assumes the management of the ownership and usage of such assets. Digital assets are design models used in deriving physical properties of a construction-, or building project. Therefore, properly managing these assets and rewards associated is a crucial property of the proposed platform. *Onboard new project* is a goal that enables users to initiate construction projects, manage technical-and business requirements associated to the project, manage the tender process and assign roles to relevant stakeholders. The *manage project goal* addresses the possibility to manage an ongoing-, or already completed project. For an ongoing project, the project managers and project owners can update, or modify requirements associated to a project. In the case of an already completed project, the user can manage tasks associated to building repairs and -maintenance. The *manage payment* goal provides the possibility of tracking milestones associated to a project and processing payments under clearly stated conditions. The *manage wallet* goal caters for allowing all involved stakeholder to manage their different token types and also for transferals between users of the platform.

3.3 Refinements of the CoPM value propositions

3.3.1 A detailed explanation for each of the FLGs describing the value propositions of the CoPM platform are presented as follows. Onboard New User: The FLG comprises the following goal refinements; select role, perform KYC and assign a key. The select role goal enables users to select the role they wish to perform on CoPM. The roles include *designer, contractor, auditor, project owner/manager and OEM.* A new user provides evidence about the capacity to function in a particular role. Such evidence can include skill certifications, educational qualifications etc. The perform KYC goal enables a KYC agent to verify data submitted by the user. If the KYC is successfully completed, the user is assigned a cryptographic key for the purpose of identification on the platform. The goal *select role* is expected to be usable by all the participants and portable. As a result, the users can easily perform tasks associated to onboarding a new user on mobile devices.

Manage Asset: The FLG contains the following sub-goals; create building object, create building model and share digital asset. The digital assets that are managed on the CoPM platform are all defined based on the standardization defined in the BIM. The building objects are a clearly defined digital representation of a building object produced by an OEM. The objects serve as an input for designing in developing the building, or construction models used by engineers in translating to a physical building construction project. A designer can create a new BIM model by importing an already existing model (on the blockchain) and creating a new model from several BIM objects. These provisions are included with the goals *import BIM model* and *create new BIM model.* The utility token is used in processing model importation and thereby rewarding appropriately the original model creator. The BIM models created by the designer are stored on the blockchain for further use. This is enabled by the goal *store BIM model* and the goal is processed using the utility token. The goal *share digital asset* provides the possibility for OEMs and designers to provide access to their created digital assets for building contractors and project owners. This goal ensures that the designers and OEMs can efficiently and transparently track the use of their intellectual property (IP) and receive appropriate credits and payments for its use. The sub-goals share digital assets shows the created assets can be used by others in the platform by directly associating the asset a project, or a project owner. Finally, both the OEMs and designers can view the building objects and models stored on the CoPM. This is enabled by the goal *view digital asset.*

Onboard New Project: The FLG onboard new project contains the sub-goals, *set requirements,* and *manage tender*. Using the set requirement goal, a project owner can outline technical- and business requirements associated to a new project. To set technical requirements, the project owner imports BIM models containing building objects as well as their clearly defined properties. The BIM model is then associated to the project and published on the blockchain. A utility token is used in importing BIM models and publishing requirements on the blockchain. The business requirements contain a set of milestones that are fulfilled before payment is issued to the contractor. Tokens are set aside, locked in a smart contract and released to the contractor after the condition in the milestone is satisfied. A utility token is used in issuing payment to the contractor on reaching a project milestone. The *manage tender* goal enables the project owner to initiate a bid for a project and manage roles associated to a project. The following sub-goals are contained in manage tender; *submit bid, select project stakeholders and assign roles.* Project designers and contractors can submit a bid for a new project design (model), or for developing an already designed project. The *submit bid* goal is processed using a utility token to prevent users from spamming the platform with invalid bids. The goal *select stakeholders* provides the possibility for software-bid agents to identify the winning bid. The goal *assign roles* allow the project owner to manually assign roles to the stakeholders involved in the project and this is processed with a utility token. The goal that enables project owners to set requirements for a new project is expected to be secure ensuring the project requirements, cannot be manipulated by any malicious user. Furthermore, the goal that enables bidding by the contractors and designers is expected to be secure as well. Publishing project requirements to the blockchain as well as the bid submission goal are expected to be performant. The rules that guide the bidding process enabled by *manage tender* goal is expected to be transparent.

Manage Project: The FLG ensures that the project owner and project manager are capable of updating requirements for an ongoing project and also maintain existing projects. Maintaining existing projects using the collaborative construction platform, involves checking the lifespan of a building object to determine if repair, or replacement is needed. The status of a building object that needs repair, or replacement is published using the utility token. The goals *replace building object* and *repair building object* trigger a new bidding process and allow contractors to submit bids for the maintenance work. The goal *recycle building object* is a sub-goal of the replace building object, allowing a project owner to earn a token for using an environmentally friendly building maintenance method. The goal *update ongoing project requirement* allows the project owner to modify the requirements of a project that is already being executed by a contractor. The new requirements are then published to the blockchain using the goal *publish BIM requirements* and this is processed using the utility token. The goal *notify stakeholders* messages all parties involved in the project about the new requirements. An offline discussion can take place and the additional payment may be requested by project contractors. The relevant quality goals of the sub-goals of the manage project FLG are presented as follows. Checking the lifespan of a building project is semi-automated involving records from IoT-recording devices and manual checks carried out by project owners. The rule that defines how the checks are conducted is transparent and can be verified by all the stakeholders of CoPM. Publishing building status, BIM requirements as well as responses from contractors regarding changes to the project requirement are expected to be performant. The listed goals result in state changes on the blockchain. The update ongoing project requirement is expected to be secure and usable by the project owner.

Manage Payment: The FLG provides the possibility to track an ongoing project and unlock an escrow-type of payment when a specific project milestone has been reached. The FLG manage payment has the following sub-goals; *manage tracking channels* and *check project milestone.* The tracking channels are the artificial agents that a contractor uses to record the project status and publish them on a blockchain. These devices may include RFID readers, barcode scanners scanning barcodes of physical building objects delivered at a construction site and IoT-video cameras for recording the graphical/pictorial state of a project. The check project milestone goal provides the possibility for auditors on the platform to verify that the project meets the technical- and business requirements described during the project onboarding. The auditing is a two-staged task executed by both human auditor and software agents - auditor. The auditor agent verifies that the BIM-objects delivered at the construction project meets the requirements of the BIM-model requirements published in the project onboarding. The barcodes of the physical building objects contain information that outlines the BIM-objects therein. The human auditors perform checks to verify the visible properties of the completed project and confirm that it is in line with the original building requirements published by the project owner. Payment is issued to the contractor after checks are successful. The make payment goal is a process using the utility token and the manage project tracking channels must be usable by the contractors. The rules for checking project milestones must be transparent to all participants in the CoPM. Publishing of the project status and making payments must be performant.

Manage wallet: The FLG provides all the participants with the possibility to manage their tokens and transfer tokens between users in the platform. The FLG manage wallet has the sub-goals *transfer tokens* and *view tokens.* The goal *transfer tokens* allow all stakeholders to share and exchange tokens among each other. To view the token history and check the details of token exchanges, the goal *view token balanc*e is used to achieve that. The goal transfer tokens is expected to be performant as it results in a state change on the blockchain network.

4 COPM ARCHITECTURE

The static components and information-exchange interfaces are derived directly from the goal models above. The DAOM framework shows the heuristic mappings used in deriving the static architecture of a blockchain application model from the requirement diagram. Section 4.1 presents the methodology that is used in transforming the system requirements represented by the goal models into architectural components of the CoPM-platform and Section 4.2 presents the CoPM component diagram.

4.1 Mapping Goal Models to Component

We show the main components, sub-components, interfaces, and ports of a system and also outline relationships that exist between them according to the concepts of the DAOM framework. In deriving the component diagram of the CoPM platform from the goal models, the following heuristics are used. The FLGs shown in Figure 1 of this paper are used in deriving the main components of the system that contain themselves sub-components equally derived from the refinements of the first-level goals. Due to limited space, only the first- and second goal refinements are used in deriving the components and sub-components of the CoPM-architecture.

4.2 Component Diagram

As shown in Figure 6 of Citation [22], the component diagram of the CoPM-architecture is obtained by applying the heuristic mapping rule described in Section 4.1 of this paper. The architecture shows six main components and the sub-components of the main components. Each component contains interfaces for reading and writing operations and actor exchanges with components are shown with broken lines.

User manager: The user manager component has the following sub-components *KYC, Key manager,* and *Role manager*. The *KYC* component has an interface for a KYC-agent to verify the identity of the users of CoPM-platform. The *role manager* has two interfaces, one for assignment of roles by the admin actor and the other interface allows specified users to read the role's data that resides in the component. The following data types are available on the user-manager component, identity data, user roles and user keys.

Digital asset manager: The component enables the creation and exchange of digital assets such as building objects and models on the platform. The component has three sub-components, and they include *building object, building model creators* and *digital asset publisher.* The building-object components provide the interface for the OEM to create and store BIM-objects on the platform that serve as input for creating building models. The dedicated creator component is accessible to designers such as architects and engineers. The component provides an interface for creating building models using the building objects from the OEM as inputs. The building models created represent the technical requirements of construction- or building projects. The sub-component digital asset publisher enables the designers to share building models created with other users on the platform. The digital-asset publisher also has a read interface that provides the possibility for specified users to access the building models as building requirements. The following data types are available on the digital asset-manager component, building-object- and building- model data.

Project onboarder: The component enables the project owner to initiate a new project by outlining the technical requirements and business requirements of the project. Also, the component manages user-biding activities that result from the project tender. The project onboarder has the following sub-components, including *requirement logger* and *tender manager.* The requirement logger provides an interface for the project owner to write and store technical- and business requirements of a new project. The technical requirements are accessed from the building models shared from the digital-asset publisher. The business requirements contain milestones and deadlines for completing the outlined technical requirements of the project initiated. The tender-manager sub-component enables the project manager and -owner to initiate bidding for a new project. The tender component provides another interface for designers and contractors to submit bids for a project. There exists also another interface for assigning project roles based on the result of the bidding. The user-roles data used by the biding agent are accessed from the role-manager component. The following data types are available on the project onboarder, project-technical requirement, project-business requirement and user-bid data.

Completed project manager: The component enables the project owner to maintain existing building projects and update the requirements of an ongoing building project when necessary. The project manager has the sub-components *maintain project container* and *requirement updater.* The project container has logs about the status of a completed building project to identify when repairs are necessary. The sub-component provides interfaces for external IoT-agents to monitor the lifespan of building objects in a completed building project and to publish the status of the building objects. The project manager and project owner can as well read the status of building objects to initiate tenders for the repair, or replacement of a building object. The requirement updater provides an interface for the project owner to modify and update the requirements of an ongoing building project. The new requirement is updated on the requirement logger component. The data types available on the project manager are building-status data and updated building-requirement data.

Payment manager: The component enables the platform to automatically monitor ongoing projects and sends payments to the contractors and designers when both technical- and business criteria of the project are satisfied. The payment has two sub-components, *tracking logger* and *milestone logger*. The former provides an interface for IoT-agents to publish the status of a project. The contractor/designer handling the project ensures that the project status is correctly published by IoT-agents. The milestone logger contains information about the milestones that have been achieved during the execution of the project. The component provides an interface for an auditor agent to process payments once specific milestones have been reached. The general information about a milestone data relevant for the project is read from the requirement logger component while the information about completed milestones is read from the tracking logger as milestone status. The payment manager component contains the project status and milestone-status data types. The setup of the sub-components of the *project onboarder* and *payment manager* enables the implementation of a modular construction concept. This is because a single project can be split into several parts and bids are submitted separately for each part. The software-agent auditor monitors the status of each of the sub-projects and issues payments accordingly.

Wallet manager: The component provides interfaces for other components to access tokens and exchange tokens with other users. The wallet manager also allows users to access and view their token balances. The wallet manager has the sub-components, *token manager* and *token history component*. In the token manager component, an interface is provided for the auditor agent to access tokens and perform payments to project executors. There occurs an exchange of the token balance between the token manager and completed project manager sub-components for updating requirements and initiating tenders. Furthermore, the token balance is exchanged with the project-onboarder component for publishing requirements and initiating tenders for new projects. The token-history sub-component has a read interface for all users in the platform to view their token balances. The token-history component accesses the users' token-balance data from the token manager. The token-balance data is the only type available in the wallet manager.

5 DYNAMIC BLOCKCHAIN PROTOCOL

This section provides the exchange protocols of dynamic behaviour on the CoPM-platform showing the transactions that are executed on blockchain referred to as the on-chain transactions. The use-cases are also presented showing how the stakeholders interact with the platform while executing the on-chain transactions. The on-chain transactions are presented in a table showing the components, transaction event, description, and associated stakeholders. The use-cases are presented using UML-sequence diagrams showing the order of activities that occurs as the system stakeholders execute the on-chain transactions. Section 5.1 shows the on-chain transactions and Section 5.2 presents the platform use-cases.

5.1 Construction Platform On-Chain Transactions

The on-chain events show the CoPM generated transactions that are executed on the blockchain. They are as follows, creation of digital assets, initiating project tenders, issuing project payments, and monitoring, and maintaining completed building projects. This mapping of the on-chain transactions to their corresponding components in the CoPM architecture is presented in [22].

5.2 Construction Platform Use-Cases

The use-cases for the CoPM-platform show the set of message exchanges that are necessary for executing the on-chain events. The use-cases are presented as UML-sequence diagrams that depict interactions between actors and components of a system in sequential order [30].

Digital asset-creation operations: This operation shows the set of message exchanges necessary for creating digital assets such as building objects (BO) and models. First, an OEM-agent imports BO from external sources and then publish the same on the platform. A building designer accesses the BO as input for creating building models (BM). The latter is then published by the designer and becomes accessible to other users for creating technical requirements of building projects. These activities are shown in detail in Figure 8 of the Citation [22].

Tender-initiation operations: To set up a project tender and identify project stakeholders, the project first accesses existing building requirements from the digital asset-manager component. The component returns to the user building models containing the necessary technical requirements to start the tender process. The project owner outlines the necessary deadlines for the project milestones and then publishes the building requirements. The contractors and engineers can then submit their bids containing payment details and possible deadlines for executing the project. The bid agent identifies the winning bids and then allocates stakeholders to the project by returning the identification-details public key (PK) and project roles of the selected contractors to the project owner. These activities are shown in detail in Figure 9 of Citation [22].

Project-payment operations: On completing projects, or reaching particular project milestones, payments are performed to contractors. The activities necessary for executing payments are outlined as follows. First, the contractor uses an IoT-agent to publish the status of the project. The data associated with project status are logged on the tracking-logger component. Based on the information contained on the project-status data, the project-milestone status is then logged on the milestone-logger component. The auditor agent reads the milestone status and issues the necessary payment to the contractor. The payment data contains the contractor identification (PK), milestone completed (Mlst) and payment amount (amt). These activities are shown in detail in Figure 10 of Citation [22].

Building-maintenance operation: This operation contains activities necessary for monitoring completed projects and initiating a tender process for building-object repairs, or -replacements. First, the project owner sets an IoT-agent to read a building status to then log the status data to the project-container component. A check is performed to confirm if a repair is needed for the building objects, or if it is necessary to replace the object. In the first case, a repair-tender process is initiated and executed on the project-onboarder component for the damaged building object. In the second case, a replacement-tender process is initiated on the project-onboarder

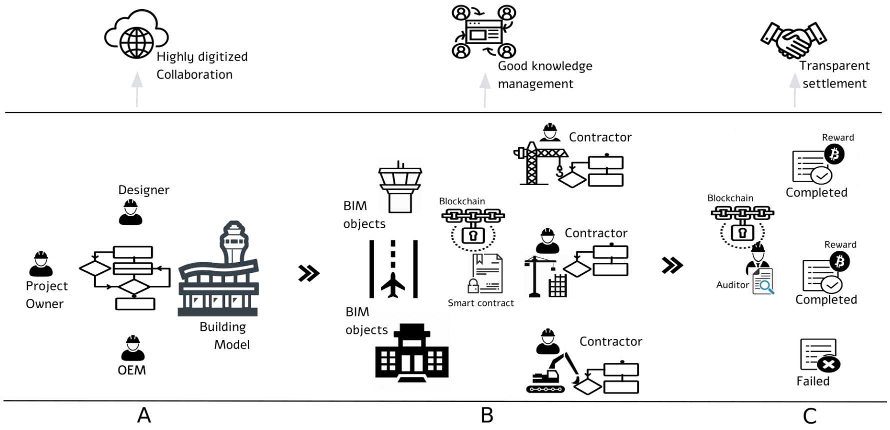


Figure 2: Proposed TO-BE state CoPM-platform ecosystem.

component. These activities are shown in detail in Figure 11 of Citation [22].

6 FEASIBILITY EVALUATION AND DISCUSSION

The TO-BE state of the modernized construction ecosystem we present based on the proposed CoPM-platform where the new running case in Figure 7 shows how the CoPM-platform transforms the low-level digitization in the construction industry into a highly automated collaboration-enabled community. Next, a paper-based evaluation of existing technologies described in Section 4 shows the feasibility of rapid development and deployment of the CoPM-architecture. The technologies outlined comprise of blockchain and non-blockchain projects that provide the possibility of quick deployment following the specific requirements of the proposed CoPM-platform. The rest of this section is structured as follows. Section 6.1 shows the digitized construction management of the running case in the resolved TO-BE state. Section 6.2 provides an evaluation of smart-contract platforms and Section 6.3 discusses decentralized storage platforms for blockchain systems. Next, Section 6.4 describes available standards for coding digital assets in the construction industry. Lastly, Section 6.5 shows available radio-frequency identifiers (RFID) and IoT-technologies for implementing scanners for building objects.

6.1 CoPM-Modernized Construction Running Case

In Figure 2 of this paper, we show the TO-BE state of the construction ecosystem after the deployment of the CoPM-platform. Unlike the current case where the complexity of projects results in project delays, rising cost and defects, the CoPM construction ecosystem enables properly managed collaboration for all the stakeholders in the construction project. First, as shown in Part A of the figure, the platform provides a collaboration system that supports the decoupling of complex project ideas into digitized building models that contain technical requirements of a construction project.

The running case of the proposed CoPM-platform ecosystem shows in Part B that digitized building models are further decomposed into subsequent building objects containing different parts of a building project. The final phase of the CoPM-ecosystem, outlined in Part C of Figure 7, shows a semi-autonomous auditor accessing building requirements from blockchains to examine whether the completed project milestones satisfy the outlined requirements. A detailed explanation of the TO-BE state of the construction management is presented in [22].

6.2 Smart-Contract Platforms

The largest blockchain network for smart contracts, Ethereum network [6], is a potential network for deploying smart-contracts for the CoPM-system. The Ethereum network still uses a proof-of-work (PoW) consensus method for the verification of transactions on the network. The PoW-consensus method requires considerable computational power for confirming transactions in the network, posing as a scalability bottleneck. This limits the number of on-chain transactions that can be executed on the CoPM-platform. Although there exist several blockchain systems that supports smart-contracts [34], we considered only platforms that have shown a high degree of stability and enterprise-wide adoption. Therefore, we identify Qtum [11], Cardano [4] and Hyperledger Fabric [8] as potential blockchains for executing smart contracts on CoPM. The Qtum network [11] is a highly scalable blockchain network for executing smart contracts and uses an incentivized proof-of-stake (PoS) consensus method for verifying transactions on the network. The Cardano blockchain network is another promising network for executing smart contracts for the CoPM-platform. Just as Qtum, the network uses PoS for verifying transactions, thereby rendering the network highly scalable when compared with the Ethereum network [15]. The Hyperledger Fabric is an example of a permissioned blockchain network for executing smart contracts. The network uses flexible consensus methods that are based on byzantine fault tolerance (BFT) systems. The BFT consensus family are highly scalable and secure [3].

6.3 Decentralized Storage

The proposed CoPM will generate a considerable amount of data objects such as digital assets that contain building requirements and data from scanners that contain building status. Due to corresponding blockchain limitations [29], decentralized storage provides extended addressable repositories for blockchain systems [10] and therefore, can be used in preserving digital assets in the proposed platform. The Interplanetary File System (IPFS) and BigchainDB are common decentralized storages used in blockchain applications [14], [9] and [26]. The IPFS is a peer-to-peer (p2p) system that provides the possibility for different computing devices to connect to the same file system for enabling access to large data sets with high efficiency over a non-trusting network [5]. BigchainDB provides a scalable and distributed database for storing large data sets generated from blockchain applications [17].

6.4 Digital Assets Standard

A universally acceptable and machine-readable standard is necessary for specifying digital assets created on the proposed decentralized CoPM-platform. The BIM-model provides an optimized system for designing building models ensuring the integration of software systems in building construction [20]. BIM is useful in construction tendering by providing a common environment for defining building data, specify technical requirements of a building project as BIM-objects and models, specify business requirements of a project as BIM-metadata. These are useful for storing digital assets created in CoPM.

6.5 Technologies for Building Scanners and –Sensors

The architecture of the proposed CoPM-platform shows significant involvement of IoT-devices in the broadcasting status of building objects when executing construction projects and when monitoring already completed projects. The project IOTA [23] provides a decentralized platform and real-time economy for processing data from IoT-devices and performing payments for services rendered. The project IOTA has the potential for providing a trusted environment for IoT-agents involved in the CoPM project. This implies that the building-status data from RFID sensors with IoT-capabilities can be transparently transmitted to the data-logging component of the proposed CoPM-platform. Still, the IOTA-project is not compatible with other major blockchain platforms for executing smart-contracts and cannot be effectively integrated into the proposed CoPM-platform. The new IBM microscopic chip[1] is a grain-sized computer with an independent processor, memory, storage and communication module that provides the possibility for transparent data transmission without the involvement of third parties. The integration of autonomous IoT-agents on the CoPM-platform and the interaction between these agents and human users outlines the social-technical aspects of the platform. The decentralized digital identity governance described in the machine-to-everything (M2X) economy [12], provides a formalized technique for identifying all the entities on the CoPM-platform.

[1]IBM Chip Computer https://cointelegraph.com/news/ibm-reveals-blockchain-computer-smaller-than-grain-of-salt-to-track-objects-devices

7 CONCLUSION, LIMITATION AND FUTURE

This paper presents the CoPM-platform, a decentralized platform for managing and executing construction projects. We describe a trusted and affordable system for outlining technical- and business requirements of construction projects and payments performed to the building contractors when specific requirements have been reached in the specified project milestone. The platform gives building-project owners and -managers the possibility of deploying a modular construction project and with smart contracts, ensures that the projects are executed according to the specified requirements. The main value proposition of the platform is to provide a decentralized system for supply-chain and project management in the construction industry. The main value proposition is further refined into the functional requirements of the system. The first-level functional requirement shows that the platform should be able to onboard new users, onboard new projects, manage digital assets, manage projects, manage payments, and manage wallets. The stakeholders that participate on the platform include human users and software agents. The latter automate the auditing and bidding functions, while the human users are mostly responsible for creating digital assets containing project requirements and executing projects. The static architecture of the proposed CoPM-platform is provided to further describe the system and a UML-component diagram. The main components of the static architecture are derived from the heuristics mapping of the first-level requirement of the platform. We use UML-sequence diagrams to describe the dynamic behaviour of the CoPM-platform. The UML-diagrams show the use-cases to outline the necessary activities that are required in executing on-chain events on the platform. Four use-cases are described, and they include digital-asset creation operations, tender-initiation operations, project-payment operations, and building-maintenance operations.

A paper-based evaluation is carried out to identify existing blockchain-and non-blockchain technologies as candidates for rapid development and deployment of the CoPM platform. Potential blockchains for executing smart contracts, the decentralized database for storing digital assets, and smart sensing devices are analysed for project suitability. A lack of case-study based empirical evaluations is the main limitation of the current paper. This is due to the not yet implemented prototype of the current design of the CoPM platform. Therefore, to address this, we propose the following future work. First, a goal is to develop a prototype implementation of the CoPM-platform. An actual construction project-management process is executed using the platform and user feedback is collected. Then a proper case-study evaluation is conducted by comparing user feedback with respect to the new CoPM system and the manual process of construction-project management.

REFERENCES

[1] E. Abodei, A. Norta, I. Azogu, C. Udokwu, and D. Draheim. Blockchain technology for enabling transparent and traceable government collaboration in public project processes of developing economies. In Conference on e-Business, e-Services and e-Society, pages 464–475. Springer, 2019.

[2] A. Afolabi, E. Ibem, E. Aduwo, and P. Tunji-Olayeni. Digitizing the grey areas in the Nigerian public procurement system using e-procurement technologies. International Journal of Construction Management, pages 1–10, 2020.

[3] E. Androulaki, *et al.* Hyperledger fabric: a distributed operating system for permissioned blockchains. In Proceedings of the thirteenth EuroSys Conference, page 30. ACM, 2018.

[4] C. Badertscher, P. Gazi, A. Kiayias, A. Russell, and V. Zikas. Ouroboros Genesis: Composable proof-of-stake blockchains with dynamic availability. InProceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pages 913–930, 2018.

[5] J. Benet. IPFS-content addressed, versioned, p2p file system.arXiv preprint-arXiv:1407.3561, 2014.

[6] V. Buterin *et al.* Ethereum white paper.GitHub repository, pages 22–23, 2013.

[7] V. Buterin *et al.* A next-generation smart contract and decentralized application platform, (2014). URl: https://github. com/ethereum/wiki/wiki/White-Paper (visited on 10/09/2016), 2014.

[8] C. Cachin *et al.* Architecture of the hyperledger blockchain fabric. In Workshop on distributed cryptocurrencies and consensus ledgers, volume 310, page 4, 2016.

[9] J. L. B. Cisneros, F. M. Aarestrup, and O. Lund. Public health surveillance using decentralized technologies. Blockchain in Healthcare Today, 2018.

[10] K. Croman, *et al.* On scaling decentralized blockchains. In International Conference on Financial Cryptography and Data Security, pages106–125. Springer, 2016.

[11] P. Dai, N. Mahi, J. Earls, and A. Norta. Smart-contract value-transfer protocols on a distributed mobile application platform.URL: https://qtum. org/uploads/-files/cf6d69348ca50dd985b60425ccf282f3. pdf, page 10, 2017.

[12] A. S. Devices. The M2X economy.

[13] M. Farmer. The farmer review of the UK construction labour model. Construction Leadership Council, 2016.

[14] E. Garc ıa-Barriocanal, S. Sanchez-Alonso, and M.-A. Sicilia. Deploying metadata on blockchain technologies. In Research Conference on Metadata and Semantics Research pages 38–49. Springer, 2017.

[15] A. Kiayias, A. Russell, B. David, and R. Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In Annual International Cryptology Conference, pages 357–388. Springer, 2017.

[16] Z. Ma, Z. Wei, and X. Zhang. Semi-automatic and specification-compliant cost estimation for tendering of building projects based on IFC data of design model.Automation in Construction, 30:126–135, 2013.

[17] T. McConaghy, R. Marques, A. M üller, D. De Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare, and A. Granzotto. Bigchaindb: a scalable blockchain database. white paper, BigChainDB, 2016.

[18] S. Mihindu and Y. Arayici. Digital construction through BIM systems will drive there-engineering of construction business practices. In2008 international conference visualisation, pages 29–34. IEEE, 2008.

[19] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[20] N. O. Nawari. Bim standard in off-site construction. Journal of Architectural Engineering, 18(2):107–113, 2012.

[21] A. Norta. Creation of smart-contracting collaborations for decentralized autonomous organizations. In International Conference on Business Informatics Research, pages3–17. Springer, 2015.

[22] A. Norta, C. Wenna, and C. Udokwu. Designing a collaborative construction-project platform on blockchain technology for transparency, traceability and information symmetry – Whitepaper, DOI: 10.13140/RG.2.2.17356.64644

[23] S. Popov. The tangle, iota whitepaper, 2018.

[24] D. Tapscott and A. Tapscott. Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world. Penguin, 2016.

[25] P. Teicholz. Labour-productivity declines in the construction industry: causes and remedies (a second look).AECbytes Viewpoint, 2013.

[26] C. D. Thimmaiah, S. Disha, D. Nayak, B. Diya, and H. Gururaj. Decentralized electronic medical records. 2019.

[27] C. Udokwu, A. Norta. Deriving and Formalizing Requirements of Decentralized Applications for Inter-Organizational Collaborations on Blockchain, Arabian Journal for Science and Engineering, DOI: doi.org/10.1007/s13369-020-05245-4 Springer 2021.

[28] C. Udokwu, H. Anyanka, and A. Norta. Evaluation of approaches for designing and developing decentralized applications on blockchain. In Proceedings of the 20204th International Conference on Algorithms, Computing and Systems, pages 55–62,2020.

[29] C. Udokwu, A. Kormiltsyn, K. Thangalimodzi, and A. Norta. The state of the art for blockchain-enabled smart-contract applications in the organization. In 2018 Ivannikov Ispras Open Conference (ISPRAS), pages 137–144. IEEE, 2018.

[30] O. UML. Omg (2017) unified modeling language®(omg uml®) version 2.5. 1https: //www. omg. org/spec. UML, 2017.

[31] A. Wulf. Der rechtsformenmangel der öffentlichen hand. eine ökonomische fallstudie zu den rechtlich-institutionellen hindernissen in öffentlichen unternehmen am beispiel des flughafens berlin brandenburg (ber). In Bedeutung des Wirtschaftsrechts für die volkswirtschaftliche Entwicklung, pages 127–174. Springer, 2018.

[32] A. J. Wulf. Participatory legal forms for public enterprises. how a group of affiliated public service organisations can help to overcome legal and institutional obstacles. Journal of Participation and Employee Ownership, 2020.

[33] J. B. H. Yap, K. Y. Lee, T. Rose, and M. Skitmore. Corruption in the Malaysian construction industry: investigating effects, causes, and preventive measures. International Journal of Construction Management, pages 1–12, 2020.

[34] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran. An overview on smart contracts: Challenges, advances and platforms.Future GenerationComputer Systems, 105:475–491, 2020.

[35] K. Anstatsios. Go Cloud. But Keep the Data! Published 2017. Accessed Dec., 2020. Url: https://www.letsbuild.com/blog/go-mobile-go-cloud-but-keep-the-data

[36] Sriram Changali, Azam Mohammad, and Mark van Nieuwland. The construction productivity imperative. Published 2015. Accessed: Dec., 2020. Url: https://www.mckinsey.com/business-functions/operations/our-insights/the-construction-productivity-imperative

Publication V

Udokwu, C., Brandtner, P., Norta, A., Kormiltsyn, A., and Matulevicius, R.
Implementation and evaluation of the DAOM framework and support tool for designing blockchain decentralized applications

correctional note

Int. j. inf. tecnol. (December 2021) 13(6):2245–2263
https://doi.org/10.1007/s41870-021-00816-6

ORIGINAL RESEARCH

Implementation and evaluation of the DAOM framework and support tool for designing blockchain decentralized applications

Chibuzor Udokwu[1] · Patrick Brandtner[2] · Alex Norta[1] · Alexandr Kormiltsyn[1] · Raimundas Matulevičius[3]

Received: 23 June 2021 / Accepted: 27 September 2021 / Published online: 21 October 2021

Abstract Inter-organizational collaboration is an important aspect of organizational operations. Traditional systems that support organizations in executing these collaborations are inefficient, not inter-operable and insecure. Novel functions provided by blockchain technology yields the potential for addressing problems that affect organizational collaborations by enabling tamper-proof, transparent, and secure systems for the exchange of information between organizations. Still, a proper approach for building blockchain-decentralized applications (DApps) that support inter-organizational collaborations is missing. The DAOM framework addresses this gap by providing a model-driven design approach for building DApps. This paper shows the development of the semantics of the DAOM framework, implementation of the support tool, and the evaluation of the DAOM framework and support tool. We conducted an evaluation to understand the usefulness of the DAOM framework in developing blockchain DApps and the effectiveness of the support tool in producing DAOM diagram models. The evaluation result shows that the framework is useful and applicable for developing DApps for inter-organizational collaborations. Furthermore, evaluation of the tool support shows that DApps can be modelled efficiently and correctly with the implemented enterprise-modelling software.

Keywords DApps · Model-driven design · Evaluation · Tool-supported implementation · Modelling language

✉ Chibuzor Udokwu
Chibuzor.udokwu@gmail.com

Patrick Brandtner
Patrick.brandtner@fh-steyr.at

Alex Norta
alexander.norta@taltech.ee

Raimundas Matulevičius
raimundas.matulevicius@ut.ee

[1] Blockchain Technology Group, Department of Software Science, Tallinn University of Technology, Tallinn, Estonia

[2] FHOOE, Steyr, Austria

[3] University of Tartu, Tartu, Estonia

1 Introduction

Inter-organizational collaborations (IOC) represent important aspects of organizational operations since companies are usually not self-sufficient and they outsource functions for which they lack expertise [8, 12]. Additionally, services that are not part of an organization's core function are procured from third-parties [29]. These types of multi-party collaborations to achieve specific organizational goals, involve the exchange of data and shared processes between the organizations. Such multi-dimension information exchange between different organizations requires security and transparency of data exchanged. Current systems that support inter-organizational collaborations are centralized, not interoperable and insecure [17, 22]. Due to insecurity, access control cannot be enforced on shared data and due to centralization, activities of collaborating parties cannot be independently verified. Additionally, a lack of interoperability leads to a manual, inefficient exchange of information among the collaborating parties. As a result, traditional information systems (IS) are not suitable for executing cross-organizational processes.

Blockchain is considered as one of the emerging technologies that will shape the future [36]. Blockchain technology provides the potential to address the challenges limiting inter-organizational collaborations. In [33], the authors show that the main purposes of applying blockchain in organizations are to improve transparency, security and interoperability of data. The study [3] shows the use of blockchain application in addressing traceability issues that exits in food supply chains, thereby improving the general safety of consumed food products. The study [31] shows the use of blockchain in enabling secure and interoperable sharing of information between users, software systems and internet of things (IoT) objects in a smart-city design. Blockchain technology has also been applied in secure and transparent sharing of patients data across service providers in the healthcare sector [4].

Notwithstanding the opportunities blockchain technology delivers with regards to addressing the problems in inter-organizational collaborations, there exists no aligned design framework for developing blockchain-decentralized applications (DApps) [32]. The study [21] proposes a software-engineering focused method for developing blockchain DApps. The proposed method involves the use of UML models including use-case diagrams, class diagrams and activity diagrams in deriving the requirement and architecture of a DApp. The proposed method does not address the transaction-cost scalability and usability problems in the technology. The study [11] describes a process-reduction method for developing blockchain DApps that addresses the transaction-cost scalability problems in blockchain technology. However, the process-reduction method for executing business processes on the blockchain does not capture the complete stages of software modelling. The study [16] proposes an ontology-driven data-modelling method for building blockchain systems. Still, the method does not consider the usability issues associated with blockchain systems. The privacy-preserving techniques for building DApps referred to as Ancile provides an approach for ensuring a secure management of healthcare data using blockchain [9]. Although the Ancile framework enables the efficient- and interoperable exchange of patient data among healthcare-service providers, it does not show general applicability in developing other systems that support organizational collaboration, except in the healthcare domain. The Caterpillar is a tool for executing- and managing business processes on blockchain [18]. The tool provides the possibility for transforming the business process modelling notations (BPMN) into smart contracts that are executed on the blockchain using its compiler. The tool is designed to run on a specific blockchain and compiles only BPMN notations. The tool is therefore not blockchain-technology agnostic and also not suitable for modelling complex business operations that cannot easily be represented in BPMN.

To address these gaps, the studies [32, 34] propose a decentralized agent-oriented modelling (DAOM) framework for building DApps and formalize the modelling syntax [32, 34]. The main goal of this study is to implement the DAOM framework described in [32, 34], develop a support tool for the framework, then evaluate the DAOM framework and the support tool. Although the syntax of the framework has previously been evaluated for correctness and consistency, we still lack an evaluation to determine the usefulness of the modelling language in describing the DApps. There exists also a need to evaluate the effectiveness of the tool support for modelling DApps using our modelling framework. Therefore, we propose the main research question of "How to implement and evaluate the DAOM framework and support tool?" The DAOM consists of three different diagram types. The first diagram outlines the main requirements of a DApp, the second diagram shows the architecture and information exchanged between components and the last diagram shows sequences of information exchange in executing a use-case of a given DApp. Thus, it is necessary to have a complete representation of the diagram elements and relationship with each element. It is also necessary to qualitatively evaluate the modelling language and the tool support separately. Therefore, we derive the following sub-questions from the main research questions:

- What is the model representation of the DAOM showing relationships between all DAOM modelling elements?
- What are the modelling constructs for implementing the DAOM diagram models?
- What is the usefulness of the DAOM framework in designing DApps and the effectiveness of the tool support in producing DAOM diagram models?

The rest of the paper is structured as follows. Section 2 provides the background of this study by summarizing the main aspects of blockchain technology and the running case for the evaluation. Furthermore, the evaluation methods used in this study are presented. Section 3 provides a complete picture of the three model types in the DAOM framework using a metal-model to describe the modelling elements and their relationships. Next, Sect. 4 describes the implementation of tool support by describing the UML profile diagrams, stereotypes and tag values used in realizing an enterprise-modelling tool for DAOM. Section 5 provides a complete evaluation of the DAOM framework by assessing the modelling language and support tool separately. The language is assessed for the usefulness in modelling DApps for inter-organizational collaborations. The support tool is examined for its effectiveness in producing models that conform with the syntax of DAOM. Finally, the conclusion of the study is presented in Sect. 6 together with future work.

2 Presuppositions

This section provides the background of this study by first summarizing the main concepts and technologies that support the blockchain in Sect. 2.1. The research method applied in this paper and running case adopted in this study are presented in Sect. 2.2. Lastly, the running case of an inter-organizational collaboration is presented in Sect. 2.3.

2.1 Main concepts in blockchain technology

Blockchain is a distributed network of peers where cryptographically linked data are stored on blocks and all the participants have the same copy of the data [30]. The cryptography ensures that each of the participants can independently verify the correctness of the stored data. In describing technologies that support blockchains, this paper considers the concepts such as consensus mechanism, public-key infrastructure (PKI) and smart contracts.

The blockchain *consensus methods* define rules for adding new records of data to the network. Some of the common consensus methods include proof of work (PoW), proof of stake (PoS), delegated PoS (DPoS) and byzantine fault-tolerant (BFT) consensus models. In PoS, a difficult mathematical puzzle is presented and the first participant to correctly solve the puzzle is rewarded to add the next block of data. In PoS, the participant to create the next block is randomly selected and the probability is based on wealth distribution in the network. DPoS is an advanced form of PoS, however, specific participants are selected to participate in block creation. In BFT, consensus systems are generally voting-based consensus methods [38].

The *public-key infrastructure* is an asymmetric cryptographic system that uses a key pair being public- and private keys for the encryption and decryption of data. The public key is used for the identification of parties in a network as well as data encryption. The encrypted data can only be decrypted by the associated private key and data stored on the network is digitally signed with the private key of the publisher. The public-key encryption and digital signatures contribute to the verifiability of information stored on a blockchain network [28].

Smart contracts are computer programs that are executed on a blockchain. A smart-contract contains a set of conditions that define the agreements between several parties. When the corresponding conditions are satisfied, the execution of smart-contract logics results in a permanent change of a blockchain state [7].

2.2 Research method

The research method adopted for this paper is the design-science research (DSR) method. The DSR provides a framework for the creation and evaluation of new artefacts [14]. Three key pillars form the foundation of the DSR and they include environment, DSR evaluation and knowledge base. The environment represents problems in organizations and application domains which the research addresses. The knowledge base provides the scientific foundation for creating new artefacts that address the detected organizational problems. The DSR evaluation involves the assessment of the created artefact [14].

The environment pillar for this research is represented by the previous works conducted by the authors of this paper in identifying problems associated with creating blockchain DApps that support IOC. These problems are summarized by the absence of framework and support tools for creating DApps for IOC [32]. Although the study [34] already describes the DAOM framework and its syntax for developing DApps that address issues in IOC, still, a proper semantic description and implementation of the support tool are missing.

The knowledge base represents existing approaches, models, techniques that form the foundation for developing the modelling framework semantics and implementing the support tool. Meta-modelling concepts are applied in describing the semantics of DAOM while, the UML profile-diagram concepts are applied in realizing the support tool of the framework [1, 5].

The DAOM framework and the support tool are the artefacts evaluated in this paper. The focus of the evaluation conducted in this study is to assess the modelling language and corresponding tool support. The evaluation approach used in this paper is based on similar studies that apply the DSR method in evaluating other software-related artefacts. To select the appropriate evaluation methods suitable for artefacts produced in this research, as per Sect. 5.1, we assess recent studies in which similar evaluations are conducted.

2.3 Inter-organizational collaboration problems: attestation and verification of identity information

During the initial coin offering (ICO) related projects in 2017, most of the DApps created are hobby projects focused on the financial application of blockchain technology [37]. Thereafter, research conducted on the application of blockchain in inter-organizational collaboration identifies the main objectives for applying blockchain organizations. These include transparency, trust, data security/privacy, resource management, tamper-proof, and interoperability [33].

Considering the IOC problems associated to Know Your Customer (KYC) identity attestation and authorization, the traditional supporting systems do not guarantee on-time verification. Additionally, the verification processes are repeatable multiple times for the same user of different services and thus, the verification result cannot be guaranteed. Still, with the tamper-proof recording that blockchain provides, repeated verification can be eliminated. An increased level of transparency is introduced in the KYC processes with the introduction of blockchain technology.

Figure 1 shows a diagrammatic description of the described case. The figure illustrates that in verifying the same identity document for different internet services, there is a possibility for failed attestation even if the document has previously been positively attested. These repeated verification processes result in delays and increased service costs.

3 The DAOM meta-model

We present the meta-model of the DAOM framework. The first part in Sect. 3.1 provides a general concept of meta-modelling in model-driven designs (MDD) and the second part in Sect. 3.2 shows the design of the DAOM metamodel using UML class diagrams.

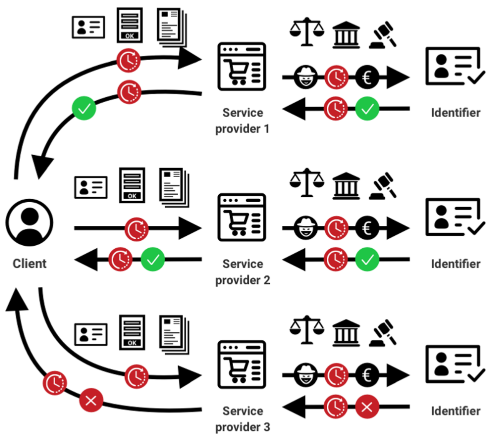


Fig. 1 Running case about the status quo in document authentication for the financial industry [34]

3.1 Meta-model concepts in MDD

To clearly understand the DAOM framework, it is necessary to outline all the elements and systematically describe the relationships between the elements and their diagram types. The meta-modelling provides a foundation for developing and implementing a model-driven design approach with the application of rules and constraints for modelling specific problem domains [1]. In this study, we apply meta-modelling for developing DAOM-element relationship rules in modelling problems in the DApp-development domain. A similar study [23] applies UML class diagrams in outlining the meta-model of a modelling framework.

3.2 DAOM meta-model design

The implemented meta-model of the DAOM framework depicted in Fig. 2 uses UML class diagrams. Three diagram types are contained in the DAOM framework for requirement, static architecture and use-case diagrams [32]. The requirement diagram consists of the elements, goals and its refinements, quality goals, emotional goal, user story and agents. The goal represents a state that is to be achieved, or a specific function performed in the DApp and they are refined into sub-goals. The quality goal defines a given software-engineering criterium for achieving a goal. It simplifies how a particular goal is to be achieved. The agent represents the software and human participants that execute a particular goal, or function in the DApp. The emotional goal describes the feeling of the human agent when executing a particular goal.

The static architecture diagram contains components, sub-components and associated agents. The static architecture of DAOM also shows the interfaces and information between the components. The use-case diagram contains a sequence of objects. The sequence outlines the interactions between objects in a DApp when executing a particular use of the DApp. The numbered sequences are also captured to identify interactions that are stored on the blockchain. [32]. Furthermore, the study [34] describes the semantics of the requirement diagram and its syntax validation. The syntax validation of the DAOM-requirement diagram is achieved by defining the elements composition of the DAOM model using an XML document type definition (DTD) object. A given DAOM-requirement diagram is then validated by transforming the model elements to XML objects and validated using the standardized XML DTD representation of the requirement model. The semantics and syntax of the static architecture and use-case diagrams follow the standard ontology of UML component- and sequence diagrams [5]. Still, the extensions of the

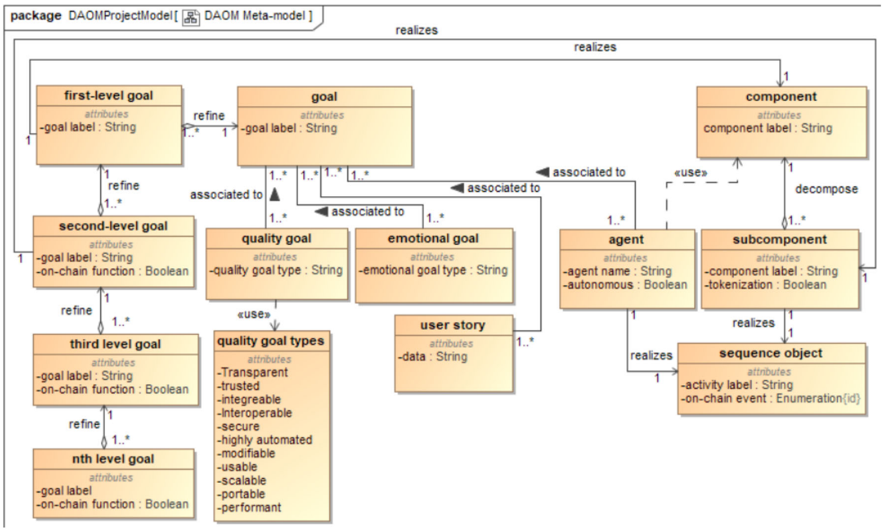


Fig. 2 DAOM meta-model class diagram

standard UML elements in the DAOM implementation of Sect. 4 are used in realizing the DAOM models.

The summary of the relationship between DAOM modelling elements is presented as follows. The goal elements are *refine* into sub-goals, up to an nth-level goal. The level of the requirements captures the depth of the requirement details described for a DApp. The quality goals, emotional goals, agents and user-stories are *associated to* the goal elements. The quality goal is defined by the element's attributes in the quality-goal-type class. The quality-goal types include transparency, trusted, integrable, interoperable, secure, highly automated, modifiable, usable, scalable, portable and performant. A detailed description of these quality-goal types is presented in [34]. The goal attribute *on-chain function* has a boolean value that determines if a given goal is executed on a blockchain. The agent has an attribute *autonomous* with a boolean value that determines if a given goal is executed by a software-, or a human agent. The first-level goal refinements are used in *realizing* the main components of the static architecture, while the second-level goal refinements *realizes* sub-components. An agent *uses* the components it is associated with. The sub-components have an attribute *tokenization* that shows if an agent interaction with a given component results in exchanging digital assets (tokens). A respective sub-component and agent *realizes* the sequence objects of the use-case description.

With the DAOM elements clearly defined and the relationship between elements and diagrams well established, the DAOM framework can easily be applied to various enterprise-modelling software that supports the creation of custom diagrams. The DAOM elements and relations can be mapped to the standard modelling notations that exist in the modelling software.

4 The DAOM tool-support implementation

This section describes the implementation of tool support for modelling DApps using the DAOM framework. The main concept for implementing the DAOM-support tool is based on the possibility of using extended UML elements with profile diagrams to realize the DAOM modelling notations on traditional system-modelling software. The modelling software used is the MagicDraw system modelling tool.[1] Still, the DAOM-diagram types implemented using the software can easily be exported to other system-modelling software such as Enterprise Architect[2] and Visual Paradigm.[3] All these cases of modelling software support the implementation of new modelling constructs by extending standard UML elements using profile diagrams.

In Sect. 4.1, we present the UML base elements used in realizing DAOM elements. In Sect. 4.2, we show the UML profile diagrams used in realizing the diagram types of the

[1] MagicDraw Software: https://www.nomagic.com/products/magicdraw.

[2] Enterprise architect software: https://sparxsystems.com/.

[3] Visual Paradigm software: https://www.visual-paradigm.com/.

DAOM framework. Lastly, in Sect. 4.3, we show a running case of a DApp for IOC modelled with the DAOM framework and software-modelling support tool.

4.1 UML base elements for DAOM

The UML base elements and their mapping to the elements of DAOM we show in Table 1. The first column with the property *type* classifies the UML base elements into element items and relationship items. The second column shows the list of all the UML elements that are mapped to the DAOM elements. The third column describes the base elements and justifies the mapping to the DAOM elements. The column with the property *extended* outlines if the given UML is extended in any form in realizing a DAOM element. The fifth column shows the DAOM element the UML base element is mapped to. The last column shows the DAOM diagram type the UML base element is used in.

In general, seven UML-base elements and six UML-relationship items are used in realizing the DAOM elements. The base elements include information item, artifact, actor, comment, component, interface and the sequence object. The relationship elements are as follows, abstraction, dependency, interface realization, interface usage, send message and reply message.

4.2 Profile diagrams for DAOM-diagram types

A UML profile diagram contains meta-classes and stereotypes for extending UML and implementing a new modelling language [5]. The meta-class shows the base element that is extended while the stereotypes show the extended attributes in the base element. For the implementation of DAOM tool support, base elements extended are UML elements in Table 1 with the extended property marked as *yes*.

Figure 3a shows a UML profile of the DAOM-requirement diagram. The *Information item* meta-class is extended to the stereotypes-, quality- and emotional goal. The goal-stereotype comprises attributes for the *refinement level* and *on-chain function*. The actor metaclass is extended by the agent stereotype with the attribute *autonomous agent*. The

Table 1 Mapping of UML elements to DAOM elements, adapted from [5]

Type	UML base elements	Detail	Extended	DAOM element	DAOM diagram type
Element	Information Item	Represents abstraction of information that can be exchanged between elements	Yes	Goal	Requirement
			Yes	Quality goal	Requirement
			Yes	Emotional goal	Requirement
	Artifact	An information produced or used by a system	Yes	On-chain goal	Requirement
			Yes	On-chain transaction	Use-case
	Actor	A use that interacts with a system	Yes	Agent	Requirement
			Yes	Agent	Static architecture
	Comment	Information note about the associated element	Yes	User-story	Requirement
	Component	A modular part of a given system	No	Component	Static architecture
			Yes	Tokenized component	Static architecture
	Interface	Support point for interaction in a system	No	Interface	Static architecture
	Sequence object/ lifeline	A connectable that participates in a system interaction	No	Sequence object	Use-case
Relationship	Abstraction	Level of relationship between elements	Yes	Decomposition	Requirement
	Dependency	Depent relationship between elements	Yes	Association	Requirement
	Interface realization	Realization of a given interface as output.	No	Interface realization	Static architecture
	Interface usage	Usage of a given interface as input	No	Interface usage	Static architecture
	Send message	Depicts a given sequential activity	No	Send message	Use-case
	Reply message	Return value of a given sequential activity	No	Reply message	Use-case

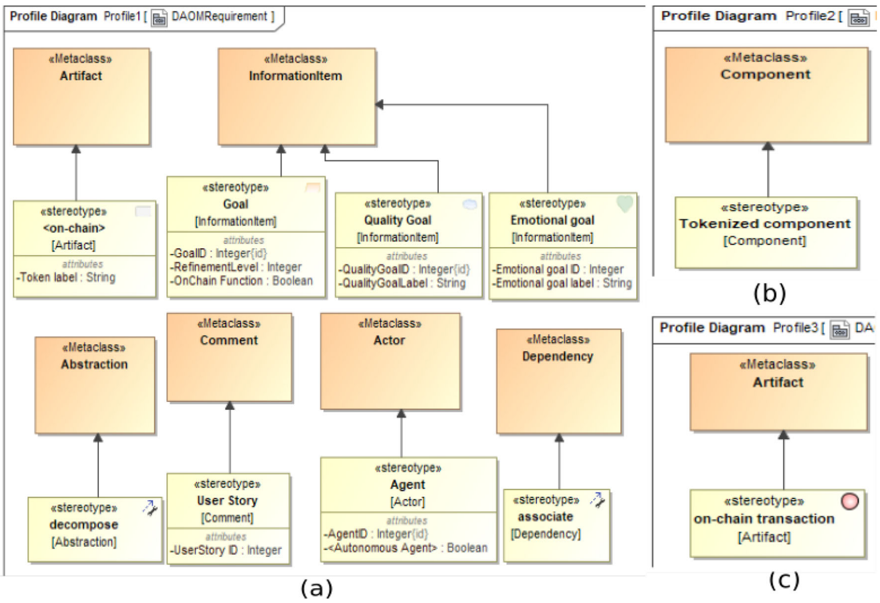


Fig. 3 DAOM profile diagrams: **a** DAOM requirement UML profile. **b** DAOM static architecture UML component-diagram extension profile. **c** DAOM use-case UML sequence-diagram extension profile

comment metaclass is extended by the user-story stereotype. The artifact metaclass is extended by the on-chain stereotype with the attribute *token label*. The relationship metaclass abstraction and -dependency are extended by stereotypes to decompose and associate respectively. For all the meta-classes, the visual representation of each DAOM requirement is captured in the stereotype icon attribute.

Figure 3b, c show the profile diagrams that extend the UML component and sequence diagram. The extended diagrams are used in implementing the DAOM static architecture and use-case diagram types. The metaclass component is extended by the stereotype tokenized component that is represented by a grey-shaded UML component element. The artifact metaclass is extended by the stereotype on-chain transaction. The visual representation is captured in the stereotype icon attribute.

4.3 Running case dapp design with a DAOM-support toolbox

The implementation of the DAOM-support tool using the UML profile diagrams developed in this study is presented as follows. For each of the DAOM-diagram types, i.e., requirement, architecture and behaviour, we present a figure showing the elements in the toolbox of MagicDraw enterprise-modelling software and a DApp modelled using the toolbox. The modelled DApp diagrams are adapted from [2], showing an *IdCredit* blockchain solution that addresses IOC problems in identity verification and -attestation. The complete system description of the IdCredit DApp is presented in the whitepaper [2]. In this paper, we only show a summary of the DAOM models used in developing the IdCredit system. Other examples showing the DAOM framework application in developing DApps are presented in the business cases of Datawallet [25] and Black insurance [26].

The current implementation of the DAOM support tool does not automatically verify the correctness of DAOM models. Still, the DAOM models demonstrated in the support tool use-case have previously been verified for correctness and consistency in the study [34]. To achieve this, the syntax of DAOM-diagram models are described in an XML document type definition (DTD) document. The developed DAOM models are then transformed to XML elements and validated.

4.3.1 DAOM-framework requirement-toolbox implementation

The DAOM requirement toolbox and a modelled IdCredit DApp requirement diagram we show in Fig. 4. The left

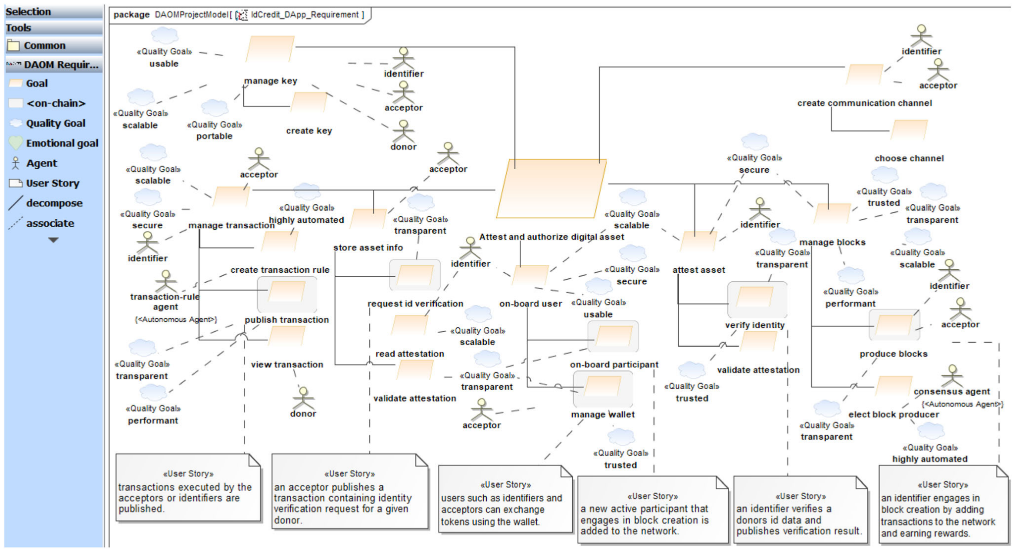


Fig. 4 DAOM-requirement toolbox and requirements of the IdCredit blockchain DApp

pane of the figure shows the DAOM-requirement toolbox elements and relationship (connectors) icons. The element icons are as follows goal, on-chain, quality goal, emotional goal, agent and user story. The connector icons are decompose- and associate icons. The IdCredit requirement diagram presented in the main part of the figure only shows the second-level goal refinements.

The main value proposition of the IdCredit DApp is to attest and authorize digital assets. The main value of the system is further refined into the seven first-level goals (FLG). The FLGs are; manage key, on-board user, create a communication channel, store asset information, attest asset, manage transactions, and manage blocks. The relevant quality goals for developing the IdCredit DApp are scalable, secure, transparent, trusted, performant, usable, portable and highly automated. Scalable implies that an increased number of users represented by their digital identity can execute the associated goal. Transparent implies the output generated after executing a given goal can be verified by all the network stakeholders. Trusted implies that the smart-contract code can enforce the conditions associated with a given goal. Usable implies that the given goal is easily understandable by the users and easy to execute. Performant implies that an increased number of a given goal can be executed over a period of time. Portable implies that a given goal can be executed from multiple ranges of devices, e.g., mobile devices. Lastly, highly automated implies that a given goal is executed by a software agent. The user-stories are attached to any goal that is executed on-chain and the content of the user-stories are self-explanatory as per the corresponding figure.

The network stakeholders in the IdCredit DApp are human agents and they are donor, acceptor and identifier. The donors are the online service users that provide personal information for KYC attestation. The acceptors are online merchants that require the donors' personal information to be attested before granting access to specific services offered. The identifiers verify, validate and attest the donors' personal information for the acceptors. The software agents in the DApp are a transaction-rule agent and consensus agent.

The *manage key* goal allows the DApp users to generate a key-pair comprising of the private- and public keys for the identification purpose in using, or interacting with the IdCredit platform. To actively participate in the IdCredit DApp platform as an acceptor, or as an identifier, the user needs to share a public key with other participants on the platform and set up a wallet that is enabled by the *on-board participant* goal. These two types of users are considered active participants on the DApp platform. On-boarding users are expected to be scalable and usable. The *create communication channel* function allows communication- and information exchange between the participants of the IdCredit system to take place outside the blockchain system. All the participants have access to the manage-key goal while only the identifier and acceptor can execute the goal to create a communication channel. The manage-key goal is expected to be usable, scalable, and portable. The communication channel protocol is external to the DApp, therefore, no quality goal is associated with it.

The goal *store asset information*, allows an acceptor to submit an identity-verification request and reuse an identity-verification status that already exists on the blockchain network and validates an attestation-request transactions. Consequently, the verification request is published on-chain. Viewing and reusing existing attestation results is expected to be scalable. For an identity document to be considered as verified, the document must be attested, hashed and stored on a blockchain. This is achieved with the goal *attest asset*. The verified identity status is stored on-chain and the steps used during the verification are expected to be trusted. The published verification status is transparently available to all participants. Only the identifiers have access to the attest-asset goal.

The goal *manage transactions* shows the transaction rule and possible transactions that are performed in the IdCredit-DApp platform. Once the blockchain is instantiated, the transaction-rule software agent creates a rule that validates the transactions recorded on the blockchain. Publishing any kind of transaction on-chain is denoted by the goal and expected to be transparent and performant. Only the active participants represented by the identifiers and acceptors can publish transactions. The donor can view published transactions to check the status of the identity attestation.

The goal *manage block* captures the requirements for creating new blocks and adding new transactions to the IdCredit blockchain DApp. Only the active participants represented by the acceptors and identifiers engage in block production. The consensus software agent automatically selects the next block producer using a defined set of rules. The produced blocks are expected to be transparent and verifiable by all the network participants. The rule for block creation is also expected to be trusted.

4.3.2 DAOM framework architecture-toolbox implementation

The DAOM architecture toolbox and a modelled IdCredit-DApp static architecture diagram we show in Fig. 5. The left pane of the figure shows the standard UML component elements and connectors as well as the extended DAOM element. The standard component-element icons used in creating a DAOM static-architecture diagram are component, interface and interface usage while the DAOM specific element is the tokenized component. The standard UML actor element is imported into the DAOM specific part of the toolbox. The IdCredit static architecture diagram is the main part of the figure to show the components and data exchanged between components in realizing the IdCredit-DApp platform.

The *communication channels* in the IdCredit architecture is an external communication tool that allows acceptors and identifiers to share data. The *key manager* sub-component allows the users to generate and share their identification key on the platform. It contains a sub-component termed a key-creating component. The *onboard manager* sub-component is used in performing the active user onboarding. This allows active users to share their public key with other participants when a new active participant is added. The wallet manager sub-component provides interfaces for executing payment transactions such as token transfers. The public key is the data generated and shared between the key-manager component and onboard manager. All stakeholders have access to both components. The on-boarding component is tokenized because tokens are spent when adding an active participant on-chain. The wallet manager is tokenized since tokens can either be gained, or spent from the component.

The *id-verification request* component provides the possibility for acceptors to submit attestation requests and reuse already existing attestations that are recorded on-chain. The identity-verification request manager sub-component provides an interface for acceptors to share a donor's identity information with the identifiers using their preferred communication protocol. The attestation-viewer sub-component provides the possibility for acceptors to view the status of attestation, or reuse an already existing attestation. The validation manager sub-component provides the possibility for acceptors to verify the transactions created by the request-manager component. The id-verification request is tokenized since tokens are spent when publishing the verification-request transaction.

The *attestation-processing* component provides an interface for an identifier to handle identification-verification requests submitted by the acceptor. The identity documents are received via the communication channel, then hashed and stored on the digital asset-manager component. The identifier conducts a manual check to verify the identity of the document received. Once the manual verification process completes, the identifier publishes an attestation transaction through the identity-verifier manager component. The validation-manager component enables users to verify transactions created by the verifier-manager component. The identity-verifier component is tokenized since the identifier gains tokens publishing attestation transaction.

The *transaction manager* component handles all transactions executed in the IdCredit DApp. All the transactions originating from the other components such as a verification-request transaction, attestation transaction and token-

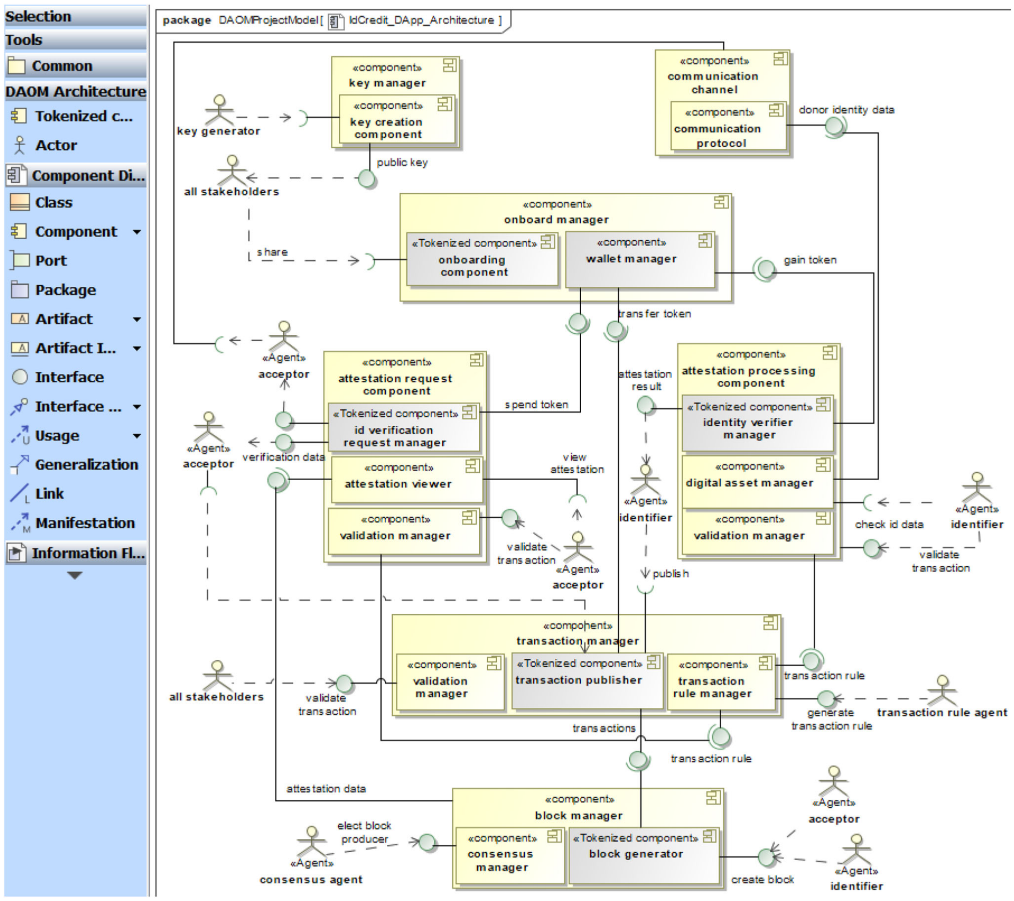


Fig. 5 DAOM architecture toolbox of the IdCredit DApp

transfer transactions are gathered and published from the transaction-publisher sub-component. The transaction-rule manager sub-component contains the rules that are used to validate all other transactions. Upon instantiating the IdCredit blockchain, the validation rule is created by the transaction rule agent. Lastly, the validation manager provides an interface for verifying all types of transactions published on-chain. The publisher component is tokenized because a publishing transaction results either token gain, or a loss from the wallet of the executor of the transaction.

The *block manager* component handles all the tasks associated with the creation of blocks in the IdCredit DApp platform. The consensus manager sub-component provides an interface for a software actor named consensus agent to generate rules that guide the creation of blocks in the network. The block-generator sub-component enables active users to create new blocks. Each block contains several transactions that are retrieved from the transaction-publisher component. The block generator component is tokenized because block creation results in tokens gain for the active participant that created the block.

4.3.3 DAOM framework behaviour-toolbox implementation

Figure 6 shows the DAOM-behaviour toolbox and modelled diagrams of the IdCredit-DApp use-cases. The left pane of the figure shows the standard UML sequence-diagram elements and connectors as well as the extended DAOM element. The standard component-element icons

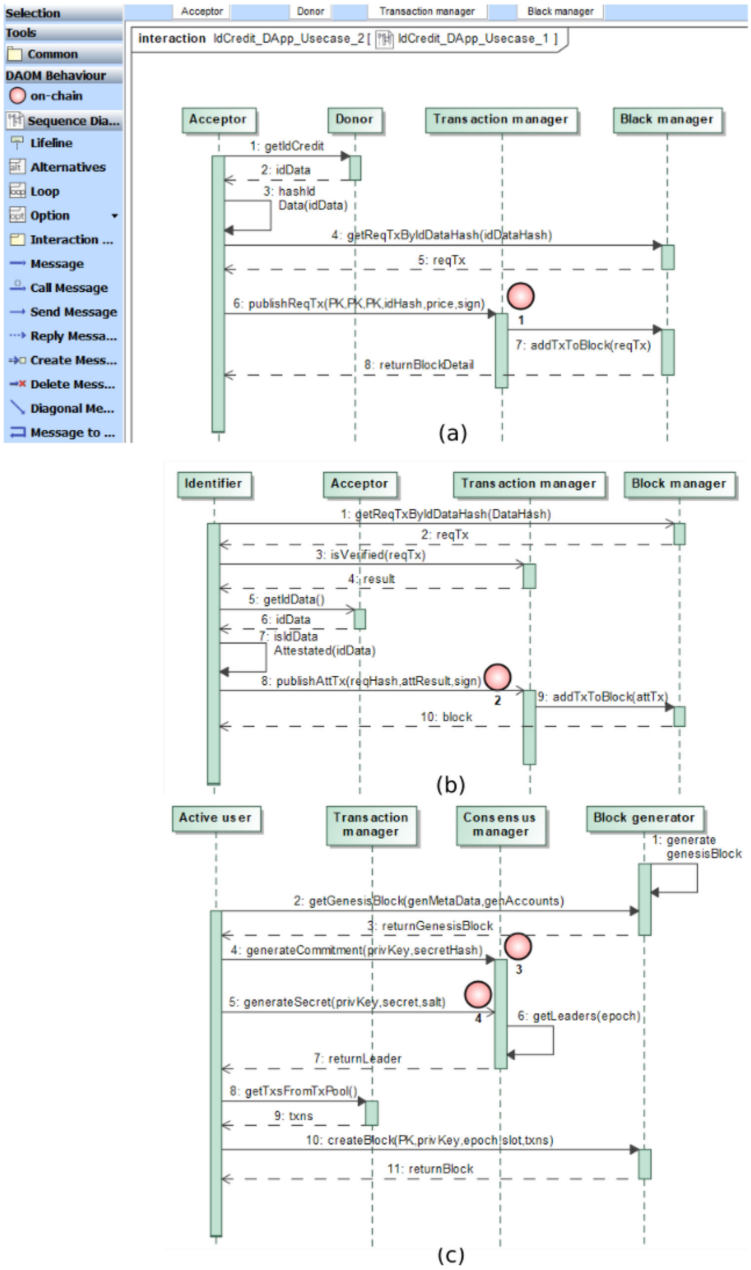


Fig. 6 DAOM behaviour toolbox and use-cases of the IdCredit-blockchain DApp. **a**) Identification-request operations. **b** Attestation operations. **c** Block-formation operations

used in creating DAOM use-case diagram are as follows, lifeline (sequence object), call message and reply. The DAOM specific element is the on-chain event counter. The IdCredit behaviour diagram in the main part of the figure shows the use-cases outlining the main operations carried out on the IdCredit-DApp platform. These operations include token-transfer operations, identification request operation, attestation operations and block-formation operations. The token-transfer operation is basic and simply represent the exchange of tokens between users on the platform. The rest of the operations are presented as follows.

The activities involved in the identification-request operation are captured in Fig. 6a. The sequence of the user activities is outlined as follows. First, the acceptor requests and receives the identity document of a donor. The id document is hashed and a check is performed to determine if an attestation result already exists for the data. If no attestation exists, the acceptor publishes a new verification request using the transaction-manager component.

The verification request transaction as well as other transactions are then gathered in a block and updated in the IdCredit blockchain. The acceptor is returned the block information containing the published transaction.

User activities for the attestation operation we show in Fig. 6b. This operation results in the publishing of identity attestation for a given donor data. The sequence of the activities is presented as follows. The identifier receives an identity document from the acceptor, then locates the associated verification request published on a blockchain. The identity-verification request published by the acceptor is verified by the identifier to confirm the correctness of the transaction. The identifier performs a manual check of the donor's ID document and publishes the result of the attestation. For a positive check, the identifier writes a positive attestation transaction and otherwise, a negative attestation result is published. The published attestation result, as well as other transactions such as verification request and token transfers, are then gathered in a block and updated in the IdCredit blockchain. The identifier is returned the block information containing the published transaction.

In Fig. 6c, we show the user activities involved in a block formation. The sequence of the activities is presented as follows. First, the genesis block is created in a block-generator component during the instantiation of a blockchain. The details of the genesis block provide input for the creation of the subsequent blocks. Two transactions, commitment and secret transactions are used as inputs for the algorithm that generates the randomness used in selecting the next block producer and these are referred to as commitment and secret transactions. These two transactions do not involve a token gain, or -loss for the executor of the transaction. The selected active participant gathers the transactions in the transaction manager component, and once the number of transactions required for block formation is reached, the new block is published by the active participant. Subsequently, the details of the newly created block are returned. In [2], the full description is presented of the consensus algorithm used in the block creation, as well as the governance method for the resolution of different block versions.

5 The DAOM evaluation

In this research, two main artifacts are produced and they are the DAOM framework and corresponding tool support. The framework provides a model-driven approach for building DApps and the support tool provides enterprise-modelling software for building DApps using the DAOM framework. These two artifacts are evaluated in this section. First, we identify the appropriate evaluation method for artifacts produced in this research in Sect. 5.1. Then, the DAOM framework is evaluated in Sect. 5.2 and in Sect. 5.3, the support tool is evaluated.

5.1 Assessment of evaluation methods for modelling languages and their support tools

We present relevant studies that have evaluated modelling languages, -frameworks, -techniques and support tools for implementing the models. Their strengths and weaknesses are captured and assessed to determine the most suitable approach for evaluating the DAOM framework and its support tool. Furthermore, the selected evaluation approaches and their adaptations for application in the current study are presented in detail.

5.1.1 Evaluation approaches

Table 2 shows recent literature that have evaluated either a modelling language, or the support tool. The column *study* shows the article assessed, the column *artefact* shows the research output produced in the study, the column *domain* shows where the modelling concept is applied to. The *notation* column captures the type of elements used in implementing the modelling language. Lastly, the *evaluated* column captures the aspects of the modelling-language evaluated.

The first article, study [6] describes a systematic technique for evaluating the syntax, semantic and usefulness of a modelling language. The developed evaluation technique is applied in assessing the qualitative aspects of a modelling language in innovation management in organizations. The second article, study [19] presents a quantitative approach for evaluating a support tool for an agent-modelling language for software development. The third article, study [13] develops a modelling language for customer-journey mappings to present an evaluation of the support

Table 2 Evaluation-method assessments for modelling languages and support tools

ID	Study	Artifact (ML/ST)	Domain	Notation	Evaluated		
					Syntax correctness	Semantic correctness	Usefulness
1	Multi-Media and Web-based Evaluation of Design Artifacts-Syntactic, Semantic and Pragmatic Quality of Process Models	ML	Innovation management	BPMN	++	++	++
2	An empirical evaluation ofthe requirements engineering tool	ST	Software development	AOM	+	−	+
3	Evaluation of a modelling language for customer journeys	ST	Customer journey	CJML	+	−	+
4	Evaluation of the E3 Process Modelling Language and Tool for the Purpose of Model Creation	ST	Software development	E3 Process	−	−	+
5	Empirical Evaluation of Tropos4AS Modelling	ML	Software development	Tropos4AS	−	+	−
6	Ontological evaluation of the UML using the Bunge-Wand-Weber model	ML	Information systems	UML	−	++	−
7	Evaluation of StudentUML: an Educational Tool for Consistent Modelling with UML	ST	Object-oriented analysis	UML	+	−	+

(−) not applicable, (+) applicable evaluation method, (++) systematic development of applicable evaluation method.

(ML) Modelling language (ST) Support tool

tool by assessing the correctness and usefulness of the models produced. As the fourth article, study [15] evaluates a new modelling technique for software development by comparing the result obtained using the support tool with the standard tool. The fifth article, study [24] evaluates the semantic aspects of a modelling language that extends standard agent-oriented software-engineering techniques by assessing the meaningfulness of the models produced. The sixth article, study [27] describes a systematic technique for evaluating semantic aspects of UML. Finally, study [10] develops a support tool for UML and evaluates the consistency of models produced using the tool as well as the usefulness of the tool when compared with standard modelling technique.

The main findings from the data presented in Table 2 is that most studies evaluate the support tool of a modelling language by comparing the usefulness of the tool with a standard modelling language. Only study [6] provides a detailed description of how the evaluation method is developed. Further, the study also captures all the evaluation aspects such as syntactic correctness, semantic correctness and general usefulness of the modelling language to the domain of application. For the study [19], although the empirical evaluations of the support tool conducted is similar in objective with other studies, the study is more relevant since the notations evaluated are similar to this case. The DAOM is an extension of the standard AOM and applicable in the blockchain domain. Therefore, the two studies, [6] and [19] we adapt for the current study.

5.1.2 Modelling-language evaluation aspects

The study [6] presents an evaluation approach for assessing syntactic-, semantic- and pragmatic qualities of a business-modelling framework for implementing innovations in organizations. The approach is adapted and applied for the blockchain DApp-development domain. The syntactic aspects of DAOM models are already described and evaluated in [32]. Only the semantic- and pragmatic aspects are considered in this study in that the semantic quality measures how the new models capture information that domain experts deem necessary in representing the domain. Table 3 shows the properties for measuring the semantic aspects of a new modelling language and its adaptation to this study. These properties are as follows, the correctness of the modelling method, relevance to the problem domain, completeness in representing the domain and authenticity of the models.

The pragmatic aspect measures the perceived usefulness of the modelling language in facilitating the design and implementation of blockchain DApps for inter-organizational collaborations. Table 4 shows the properties for measuring the pragmatic aspects of a new modelling language and the adaptation to this study. These properties are as follows, subjective norm, image, job relevance, output

Table 3 Semantic qualities for assessing models, adapted from [6]

Title	Description	Adaptation
Correctness	All statements in the representation	The DAOM framework represents the process and elements of building Dapps correctly
Relevance	All statements in the representation are relevant to the problem	All the elements in the DAOM framework are relevant for building Dapps
Completeness	The representation contains all statements about the domain that are correct and relevant	The DAOM framework gives a complete representation of the elements and process of building Dapps
Authenticity	The representation gives a true account of the domain	The DAOM framework is realistic representation of the elements and process of building Dapps

Table 4 Pragmatic qualities for assessing models, adapted from [6]

Title	Explanation
Subjective Norm	People who are important would support using DAOM framework in building Dapps
Image	People in my organisation who use DAOM framework would have a high profile
Job Relevance	In my job, usage or application of DAOM framework would be relevant
Output Quality	The quality of the output I get from using DAOM framework will be high
Results Demonstrability	I believe I could communicate to others the opportunities in using DAOM framework
Performance	Using the DAOM framework would improve my performance in my job
Productivitiy	Using the DAOM framework in my job would increase my productivity
Perceived Usefulness	I find the DAOM framework to be useful in my job

quality, result demonstrability, performance, productivity and perceived usefulness. In both the semantic- and pragmatic evaluations, experts in the blockchain system-design domain constitute the experiment participants.

5.1.3 Modelling-language tool support evaluation aspects

The study [19] develops and evaluates a support tool for agent-oriented modelling designs. The evaluation approach is based on comparative analyses of results gathered when modelling with the support tool against free-hand sketch. The goal is to determine the benefits of the support tool by assessing the correctness of the models produced and time spent in producing the models.

Table 5 shows the properties for assessing the performance of a modelling-framework support tool adapted from [19] for this study. The items measure the difficulties, time spent and efforts in producing the diagram types of the DAOM models. Other properties measured include the understanding of the case modelled, understanding of the DAOM concepts and application of DAOM in practice. For this type of evaluation, novices in the blockchain domain constitute the experiment participants.

5.2 DAOM framework evaluation

To evaluate the framework, we conduct a webinar event involving domain experts in the blockchain field and present the DAOM framework. The experts are drawn from different industry backgrounds such as supply chain, healthcare, finance, education and research. The webinar has a total of 15 participants. Figure 7 provides detailed information on the background of domain experts that evaluated the DAOM framework. The figure shows the industry background, job positions, domain field and experience level of the participants. About 37% of the participants are from research and education background, 25% are from supply chain and fintech and 13% from healthcare. Most of the participants are highly experienced as 80% have an experience level of 2 years and above. With regards to the field domain, 60% of the participants are from academics, while 40% are from the industry. A scale of 1 to 5 is used to record the feedback of the domain experts for a given semantic property of the DAOM framework measured. A record of 1 represents the lowest score, while a record of 5 represents the highest score for the measured property.

Using the modelling language evaluation method presented in the Tables 3 and 4 in Sect. 5.1, the semantic- and

Table 5 Properties for assessing modelling-support tool, adapted from [19]

Item	Description
q1	The description of the case study was clear to me
q2	Difficulties in modelling the requirement diagram of the DAOM
q3	Difficulties in modelling the static architecture diagram of the DAOM
q4	Difficulties in modelling the use-case diagram of the DAOM
q5	Short time is required for accomplishing the modelling task
q6	Goal decomposition was very useful in the Dapp design
q7	The concepts of the DAOM framework were detailed enough to model the requirements of the blockchain system
q8	The effort of modelling seems too high for an efficient use of the methodology in practice

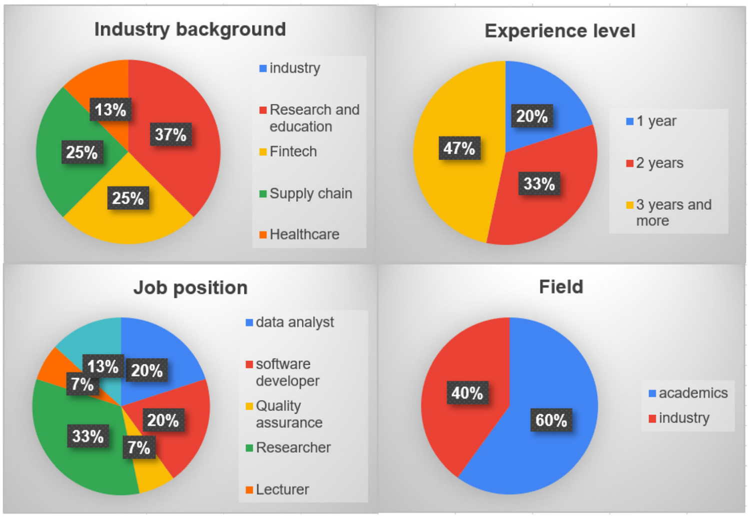


Fig. 7 Background of domain experts

pragmatic qualities of the DAOM are examined. Figure 8 shows the evaluation results of the DAOM framework. The left part of the figure shows the semantic qualities while the right side shows the pragmatic qualities represented as perceived usefulness.

For the semantic quality evaluation, blockchain domain experts all agree that the DAOM framework is highly relevant and realistic in representing the design process of DApps. They also agree that the framework is very correct and consistent in representing elements for designing

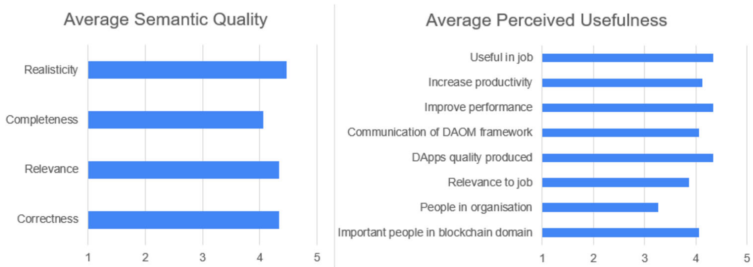


Fig. 8 DAOM framework evaluation result

DApps. The properties are rated with the average scores as follows, realisticity 4.3, completeness 4.3, relevance 4.1 and correctness 4.5. Applying the same scale in pragmatic quality, the DAOM-framework usefulness in jobs in blockchain domain, producing quality DApps output and improved performance in DApp implementation are rated highly (above 4.33) by the participants. Other pragmatic properties of the DAOM framework with good scores include, increase in productivity and communication of the DAOM framework, with the scores of 4.1 respectively. Also rated with the good score 4.0, is the subjective norm property of how important people in the blockchain domain perceive the DAOM framework. The result of this evaluation show that blockchain domain experts in agree that the DAOM framework is semantically correct and pragmatically useful in designing and developing DApps.

In providing additional details of the semantic evaluation based on the background of the domain experts, similar scores are recorded by experts from academic backgrounds and ones from the industry. The average semantic quality of the framework provided by experts from academics and industry are 4.28 and 4.27 respectively. Similar scores are recorded for the pragmatic qualities of the framework by domain experts in academics and industry. The experts from academics ranked the average pragmatic quality of the framework 4.06, while the experts from industry ranked the pragmatic quality 4.07. These show that people from academics and industry have a similar perception of the semantic properties and pragmatic usefulness of the framework.

Considering the three major job positions that participated in the workshop, researchers, software developers and data analysts and their perceptions about the semantic correctness and practical usefulness of the framework. For the semantic properties, data analysts, software developers and researchers ranked the framework 4.5, 4.3 and 4.5 respectively. The similar and higher scores recorded by researchers and analysts could imply a better understanding of the framework since experts in this role generally involve in the initial stages of blockchain DApps design, unlike the software developers that are involved in the development and prototyping stage. The DAOM framework mainly supports the design phase of software development. For the pragmatic usefulness of the framework in jobs, the scores recorded are 4.12, 4.2 and 4.07 respectively for data analysts, software developers and researchers. Also, similar scores are recorded for the analysts and researchers, however, lower than the score for software developers. This shows that although data analysts and researchers have a better understanding of the semantics of the framework, in terms of application of the framework to their job roles, the software developers consider the framework more useful. Generally, all these job roles ranked the framework high in terms of semantic qualities and practical usefulness.

5.3 DAOM support-tool evaluation

To evaluate the DAOM framework support, a workshop was conducted and participants drawn from seven (7) masters student conducting various related projects in blockchain domain. The workshop participants are divided into two groups, to model a sample DApp using the DAOM framework. The first group consists of 4 students and the second group consists of 3 students. The students are given a task to recreate a sample DAOM-requirement diagram of a given case of DApp design. The first group of students uses the DAOM-framework support tool and the second group uses a free-hand sketch tool. The correctness of the models produced, time spent and easiness of the task are captured using a feedback form.

The freehand tool selected for the comparative analyses of the DAOM support tool is the draw.io tool. Currently, no other standardized tool exists for producing diagrams of the DAOM framework. Furthermore, the draw.io tool has previously been used by the authors of this publication for developing DAOM diagram models before the implementation of the support tool. Some of the DAOM models previously produced with draw.io can be found in [25, 26, 35]

By applying the modelling-language support-tool evaluation method presented in Table 5 in Sect. 5.1, we check the easiness (usability) of producing DApp designs using the DAOM support tool. This is assessed by comparing feedback data from students that created DApp designs using the DAOM support tool versus students that use the freehand sketch tool. Figure 9 shows the evaluation results of the DAOM support tool. The red-coloured bar charts are average scores from the students with the freehand tool while the blue represents the average scores from the students that use the DAOM support tool. A scale of 1 to 5 is used to record the feedback of the students for a given

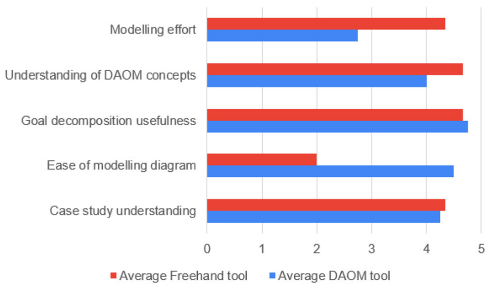


Fig. 9 DAOM support tool evaluation result

property measured. A record of 1 represents the lowest score, while a record of 5 represents the highest score for the measured property.

The results in Fig. 9 show that all the students that participate in the workshop have relatively the same understanding of DAOM concepts and the running case presented. Yet, the effort required to re-create the modelling task and the easiness of using the drawing tools differs among the students. The students that complete the design task using the support tool find the effort required in re-creating the DAOM requirement diagram relatively low, unlike the students that use a freehand drawing tool that requires a high effort in completing the modelling task. The ease of modelling the diagram is high for the students that uses the support tool. For the students that uses the freehand tool, the modelling task is reported as being difficult to complete. The results confirm that the automatons available in the DAOM support tool help in designing DAOM DApp models with ease and little effort.

6 Conclusion, limitations and future work

6.1 Discussions

To highlight the novelty of the research work conducted in this paper, we compare the findings of this paper with other similar studies. Two similar studies that provide systematic approaches for building blockchain applications are compared with the results of the current paper. The study [20] presents a UML based software engineering method for developing blockchain applications that provide an approach for capturing applications requirements and communicating such requirements to the project stakeholders. The study [11] provides an approach for developing scalable blockchain applications by using the process reduction method for transforming and reducing standard business processes into reduce Petri-nets and then generating the resulting application smart-contract codes.

The DApp modelling framework presented in [20] uses the traditional UML diagrams and elements in representing the DApp models and does not extend the UML in any way to capture the novel properties of the blockchain technology such as on-chain functions and component tokenizations. In our approach, the DAOM framework extends the notations of UML and agent-oriented model (AOM) notations to capture these important attributes of the blockchain. To evaluate the modelling approach developed in [20], feedback was collected from DApp developers who have previously applied the framework in building various DApps. The purpose of the gathered feedback is to evaluate the practical usefulness of the framework by measuring its ease of use, ability to properly integrate on-chain and off-chain functions, requirements elicitation of DApps, and security analyses of the designed DApp. However, in our paper, in addition to assessing the practical usefulness of the developed modelling framework, we also examined the semantic properties. The semantic evaluation of the DAOM framework helps in understanding the correctness, completeness, relevance and realisticity of models produced when the framework is used in building blockchain DApps.

To address scalability issues in the blockchain, the study [11] applies a Petri-net based process transformation and reduction method of a given business process that is to be executed on the blockchain. The main weakness of this approach is the assumption that all the functions(tasks) in a given business process have to be executed on blockchain to achieve transparency, traceability and trustability. However, in the DAOM approach, the specific functions that enables transparency and trustability in organizational collaborations are identified in the initial stages of the software requirement elicitation. Such functions that are executed on blockchains and are referred to as the on-chain functions. The rest of the functions are considered off-chain functions. Thus, scalability is achieved in the DAOM approach by identifying and optimizing these on-chain functions in the early stages of the DApp model design.

6.2 Conclusion

The main objective of this paper is to assess the DAOM framework and this involves evaluating the modelling framework semantics and the effectiveness of the support tool. To achieve this, we first described the DAOM diagram types and their element relationships systematically using the meta-modelling concept. Then we show the DAOM-diagram constructs by outlining the UML profile diagrams for implementing the DAOM framework as well as DApps models realised using the DAOM support tool. Lastly, we evaluate the semantics of the DAOM framework to identify its usefulness and then evaluate the support tool to determine the effectiveness in producing DAOM diagram models.

The UML class diagram is used in developing the DAOM framework meta-model used in this study. The meta-model of the framework shows the relationship between the elements and diagram types being requirement, static architecture and behaviour diagrams. For the requirement aspect, the meta-model shows the goal-element refinements to sub-goals, their association with other elements such as quality- and emotional goals, agents and user-stories. The meta-model also shows the heuristic mapping of the goal elements to the static architecture elements. The first- and second level goals are used in realizing the main components and sub-components of the static architecture diagram. While the agents associated with specific architecture components and sub-component elements are used in realizing the sequence objects that represent the behaviour diagram of the framework.

To develop actual DOAM diagrams, we implement a support tool, apply the DAOM framework to the running case presented in this paper and realize a model diagram for DApp design. The tool support is implemented using the UML profile diagrams on a standard enterprise-modelling software. The profile diagrams are developed from base UML elements that share similar properties with the DAOM-diagram elements. As a result, the DAOM requirement-diagram toolbox is developed by implementing a new diagram type, while the architecture and behaviour-diagrams toolboxes are developed by extending the UML component- and sequence diagram respectively. The implemented DAOM support tool is then used in designing a sample DApp for the attestation and verification of identity data on a blockchain.

To assess the DAOM framework, first, we evaluate its usefulness in blockchain DApp development. This is done by examining the semantic- and pragmatic qualities of the modelling framework. The semantic qualities are analyzed by examining how experts in the blockchain domain perceive the correctness, completeness and relevance of the DAOM framework in representing DApps design and -development. The pragmatic qualities assess the usefulness of blockchain-related jobs and applicability of the framework by people in organizations in productively realizing quality DApp designs. The results from the semantic evaluation shows that the DAOM framework provides a realistic and correct representation of steps required in creating DApps. The pragmatic evaluation results show that the framework is perceived as very useful in producing quality DApp designs. For the support-tool evaluation, the effectiveness is determined by comparing results obtained when modelling DAOM diagrams with a free-hand sketch tool and the DAOM tool. The results show that the support tool requires less modelling effort and provides more usability in producing DAOM-diagram models.

6.3 Limitation

The main limitation of this paper is the risk of generalization in interpreting the evaluation results. Considering the limited number of experts that participated in the workshop used in evaluating the artefact produced in this work, there is the possible impact of risk of generalizations for the results generated in this research. Still, blockchain technology is a new innovation space and the number of experts is limited with the required knowledge to participate in webinar/ workshop conducted in this research. As a result, gathering a large number of experts to participate in an evaluation event conducted for this paper is a noteworthy limitation. Also, specifically for the support tool evaluation, the participants are unequally grouped into two due to the odd number of the participating students. This unequal distribution can potentially affect the experiment results of the support tool evaluation.

6.4 Future work

The future work for this study is to extend the DAOM framework by providing additional support in the DApp development stage. The current implementation of the framework mostly focuses on DApp design with little support for software development and code generation. The essence of model-driven software engineering is to automate software-development processes by using graphical models in code generation. Therefore, as future work, we propose the extension of the DAOM framework to capture the automatic generation of smart-contract codes for common DApp functions.

Funding Open access funding provided by University of Applied Sciences Upper Austria.

References

1. Atkinson C, Kuhne T (2003) Model-driven development: a metamodeling foundation. IEEE Softw 20(5):36–41
2. Balahontsev V, Tsikhilov A, Norta A, Udokwu C (2021) A blockchain system for the attestation and authorization of digital assets version 1.0
3. Balamurugan S, Ayyasamy A, Joseph KS (2021) Iot-blockchain driven traceability techniques for improved safety measures in food supply chain. Int J Inf Technol pp 1–12
4. Batchu S, Henry OS, Hakim AA (2021) A novel decentralized model for storing and sharing neuroimaging data using ethereum blockchain and the interplanetary file system. Int J Inf Technol
5. Booch G (2005) The unified modeling language user guide. Pearson Education India, London
6. Brandtner P, Helfert M (2018) Multi-media and web-based evaluation of design artifacts-syntactic, semantic and pragmatic quality of process models. Int J Inf Technol Action Commun Workpract 11(1):54–78

7. Buterin V et al (2014) A next-generation smart contract and decentralized application platform. White Pap 3(37)
8. Cullen S, Seddon PB, Willcocks L (2006) Managing outsourcing: the lifecycle imperative, vol 139. London School of Economics and Political Science, London
9. Dagher GG, Mohler J, Milojkovic M, Marella PB (2018) Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. Sustain Cities Soc 39:283–297
10. Dranidis D (2007) Evaluation of studentuml: an educational tool for consistent modelling with uml. In *Proceedings of the Informatics Education Europe II Conference*
11. García-Bañuelos L, Ponomarev A, Dumas M, Weber I (2017) Optimized execution of business processes on blockchain. In International conference on business process management, pp 130–146. Springer
12. Gray B (2000) Assessing inter-organizational collaboration: Multiple conceptions and multiple methods. Cooperative strategy: economic, business, and organizational issues, pp 243–260
13. Halvorsrud R, Haugstveit IM, Pultier A (2016) Evaluation of a modelling language for customer journeys. In 2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pages 40–48. IEEE
14. Hevner AR, March ST, Park J, Ram S (2004) Design science in information systems research. *MIS Q* pp 75–105
15. Jaccheri ML, Stålhane T (2001) Evaluation of the e3 process modelling language and tool for the purpose of model creation. In: International Conference on Product Focused Software Process Improvement, pp 271–281. Springer, New York
16. Kim HM, Laskowski M (2018) Toward an ontology-driven blockchain design for supply-chain provenance. Intell Syst Account Financ Manag 25(1):18–27
17. Kormiltsyn A, Udokwu C, Karu K, Thangalimodzi K, Norta A (2019) Improving healthcare processes with smart contracts. In: International conference on business information systems, pp 500–513. Springer, New York
18. López-Pintado O, García-Bañuelos L, Dumas M, Weber I (2017) Caterpillar: a blockchain-based business process management system. In: BPM (Demos)
19. Mahunnah M, Taveter K, Matulevičius R (2018) An empirical evaluation of the requirements engineering tool for socio-technical systems. In: 2018 IEEE 7th International Workshop on Empirical Requirements Engineering (EmpiRE), pp 8–15. IEEE
20. Marchesi L, Marchesi M, Tonelli R (2020) Abcde–agile block chain dapp engineering. Blockchain: Res Appl 1(1–2):100002
21. Marchesi M, Marchesi L, Tonelli R (2018) An agile software engineering method to design blockchain applications. In: Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia, pp 1–8
22. Matulevičius R, Norta A, Udokwu C, Nõukas R (2017) Assessment of aviation security risk management for airline turnaround processes. In: Transactions on large-scale data-and knowledge-centered systems XXXVI, pp 109–141. Springer
23. Mayer N (2009) Model-based management of information system security risk. PhD thesis, University of Namur
24. Morandini M, Perini A, Marchetto A (2011) Empirical evaluation of tropos4as modelling. iStar 766:14–19
25. Norta A, Hawthorne D, Engel SL (2018) A privacy-protecting data-exchange wallet with ownership-and monetization capabilities. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp 1–8, IEEE
26. Norta A, Rossar R, Parve M, Laas-Billson L (2019) Achieving a high level of open market-information symmetry with decentralised insurance marketplaces on blockchains. In: Intelligent computing-proceedings of the computing conference, pp 299–318, Springer, New York
27. Opdahl AL, Henderson-Sellers B (2002) Ontological evaluation of the uml using the bunge-wand-weber model. Softw Syst Model 1(1):43–67
28. Pilkington M (2016) Blockchain technology: principles and applications. In: Research handbook on digital transformations, Edward Elgar Publishing
29. Soliman KS, Janz BD, Puschmann T, Alt R (2005) Successful use of e-procurement in supply chains. Supply Chain Manag Int J
30. Swan M (2015) Blockchain: Blueprint for a new economy. O'Reilly Media, Inc.
31. Tiwari A, Batra U (2021) Ipfs enabled blockchain for smart cities. Int J Inf Technol 13(1):201–211
32. Udokwu C, Anyanka H, Norta A (2020) Evaluation of approaches for designing and developing decentralized applications on blockchain. In: Proceedings of the 2020 4th international conference on algorithms, computing and systems, pp 55–62
33. Udokwu C, Kormiltsyn A, Thangalimodzi K, Norta A (2018) The state of the art for blockchain-enabled smart-contract applications in the organization. In: 2018 Ivannikov Ispras Open Conference (ISPRAS), pp 137–144. IEEE
34. Udokwu C, Norta A (2021) Deriving and formalizing requirements of decentralized applications for inter-organizational collaborations on blockchain. Arab J Sci Eng pp 1–18
35. Udokwu C, Norta A, Wenna C (2021) Designing a collaborative construction-project platform on blockchain technology for transparency, traceability, and information symmetry. In: 2021 2nd Asia service sciences and software engineering conference, pp 1–9
36. Yamin M (2019) Information technologies of 21st century and their impact on the society. Int J Inf Technol 11(4):759–766
37. Zetzsche DA, Buckley RP, Arner DW, Föhr L (2017) The ico gold rush: It's a scam, it's a bubble, it's a super challenge for regulators. University of Luxembourg Law Working Paper (11):17–83
38. Zheng Z, Xie S, Dai H, Chen X, Wang H (2017) An overview of blockchain technology: Architecture, consensus, and future trends. In: 2017 IEEE international congress on big data (BigData congress), pp 557–564, IEEE

Int. j. inf. tecnol.
https://doi.org/10.1007/s41870-022-01026-4

CORRECTION

Correction to: Implementation and evaluation of the DAOM framework and support tool for designing blockchain decentralized applications

Chibuzor Udokwu[1] · Patrick Brandtner[2] · Alex Norta[1] · Alexandr Kormiltsyn[1] · Raimundas Matulevičius[3]

Correction to:
Int. j. inf. tecnol. (December 2021) 13(6):2245–2263
https://doi.org/10.1007/s41870-021-00816-6

In the original Publication, the below references were missed, and they are inserted as below:

There are 2 locations in the paper for these 3 references to be inserted. The first insertion location for all these 3 references listed above is at the end of the first paragraph of Sect. 3.2 and I add a screenshot for clarification:

Location 1.

Location 2.

The second insertion location of the same 3 references is at the end of the caption for Fig. 2 and again, I insert for you a screenshot below:

[A] Leon Sterling and Kuldar Taveter. (2009). The Art of Agent-Oriented Modeling. Cambridge, MA, and London, England: MIT Press.

[B] Miller, T.; Lu, B.; Sterling, L.; Beydoun, G.; Taveter, K. (2014). Requirements Elicitation and Specification Using the Agent Paradigm: The Case Study of an Aircraft Turnaround Simulator. IEEE Transactions on Software Engineering, 40, 1007–1024, https://doi.org/10.1109/TSE.2014.2339827.

[C] Miller, T., Pedell, S., Lopez-Lorca, A. A., Mendoza, A., Sterling, L., & Keirnan, A. (2015). Emotion-led modelling for people-oriented requirements engineering: the case study of emergency systems. Journal of Systems and Software, 105, 54–71.

The original article can be found online at https://doi.org/10.1007/s41870-021-00816-6.

✉ Raimundas Matulevičius
raimundas.matulevicius@ut.ee

Chibuzor Udokwu
Chibuzor.udokwu@gmail.com

Patrick Brandtner
Patrick.brandtner@fh-steyr.at

Alex Norta
alexander.norta@taltech.ee

[1] Blockchain Technology Group, Department of Software Science, Tallinn University of Technology, Tallinn, Estonia

[2] FHOOE, Steyr, Austria

[3] University of Tartu, Tartu, Estonia

ACTA UNIVERSITATIS LAPPEENRANTAENSIS

999. VIITALA, MIRKA. The heterogeneous nature of microplastics and the subsequent impacts on reported microplastic concentrations. 2021. Diss.

1000. ASEMOKHA, AGNES. Understanding business model change in international entrepreneurial firms. 2021. Diss.

1001. MUSTO, JIRI. Improving the quality of user-generated content. 2021. Diss.

1002. INKERI, EERO. Modelling of component dynamics and system integration in power-to-gas process. 2021. Diss.

1003. GARIFULLIN, AZAT. Deep Bayesian approach to eye fundus image segmentation. 2021. Diss.

1004. ELFVING, JERE. Direct capture of CO2 from air using amine-functionalized resin - Effect of humidity in modelling and evaluation of process concepts. 2021. Diss.

1005. KOMLEV, ANTON. Magnetism of metal-free graphene-based materials. 2021. Diss.

1006. RISSANEN, MATTI. EcoGame and Ecosystem Profiler: solutions for business ecosystem management. 2021. Diss.

1007. VANHAMÄKI, SUSANNA. Implementation of circular economy in regional strategies. 2021. Diss.

1008. LEHTINEN, VESA. Organisaation emergentti itseohjautuvuus, case sinfoniaorkesteri: ”Miksi orkesteri soittaa hyvin, vaikka sitä johdettaisiin huonosti?”. 2022. Diss.

1009. KÄHKÖNEN, TIINA. Employee trust repair in the context of organizational change – identification and measurement of active trust repair practices. 2022. Diss.

1010. AHONEN, AILA. Challenges in sport entrepreneurship: cases in team sport business. 2022. Diss.

1011. LEVIKARI, SAKU. Acoustic emission testing of multilayer ceramic capacitors. 2022. Diss.

1012. ZAHEER, MINHAJ. Evaluation of open-source FEM software performance in analysing converter-fed induction machine losses. 2022. Diss.

1013. HAAPANIEMI, JOUNI. Power-based electricity distribution tariffs providing an incentive to enhance the capacity effectiveness of electricity distribution grids. 2022. Diss.

1014. BUAH, ERIC. Artificial intelligence technology acceptance framework for energy systems analysis. 2022. Diss.

1015. GIVIROVSKIY, GEORGY. In situ hydrogen production in power-to-food applications. 2022. Diss.

1016. SOMMARSTRÖM, KAARINA. Teachers’ practices of entrepreneurship education in cooperation with companies. 2022. Diss.

1017. KAN, YELENA. Coherent anti-stokes raman scattering spectromicroscopy in biomedical and climate research. 2022. Diss.

1018. MÄNDMAA, SIRLI. Financial literacy in perspective – evidence from Estonian and Finnish students. 2022. Diss.

1019. QORRI, ARDIAN. Measuring and managing sustainable development in supply chains. 2022. Diss.

1020. MARTIKAINEN, SUVI-JONNA. Meaningful work and eudaimonia: contributing to social sustainability in the workplace. 2022. Diss.

1021. MANNINEN, KAISA. Conducting sustainability target-driven business. 2022. Diss.

1022. LI, CHANGYAN. Design, development, and multi-objective optimization of robotic systems in a fusion reactor. 2022. Diss.

1023. CHOUDHURY, TUHIN. Simulation-based methods for fault estimation and parameter identification of rotating machines. 2022. Diss.

1024. DUKEOV, IGOR. On antecedents of organizational innovation: How the organizational learning, age and size of a firm impact its organizational innovation. 2022. Diss.

1025. BREIER, MATTHIAS. Business model innovation as crisis response strategy. 2022. Diss.

1026. FADEEV, EGOR. Magnetotransport properties of nanocomposites close to the percolation threshold. 2022. Diss.

1027. KEPSU, DARIA. Technology analysis of magnetically supported rotors applied to a centrifugal compressor of a high-temperature heat pump. 2022. Diss.

1028. CHAUHAN, VARDAAN. Optimizing design and process parameters for recycled thermoplastic natural fiber composites in automotive applications. 2022. Diss.

1029. RAM, MANISH. Socioeconomic impacts of cost optimised and climate compliant energy transitions across the world. 2022. Diss.

1030. AMADI, MIRACLE. Hybrid modelling methods for epidemiological studies. 2022. Diss.

1031. RAMÍREZ ANGEL, YENDERY. Water-energy nexus for waste minimisation in the mining industry. 2022. Diss.

1032. ZOLOTAREV, FEDOR. Computer vision for virtual sawing and timber tracing. 2022. Diss.

1033. NEPOVINNYKH, EKATERINA. Automatic image-based re-identification of ringed seals. 2022. Diss.

1034. ARAYA GÓMEZ, Natalia Andrea. Sustainable management of water and tailings in the mining industry. 2022. Diss.

1035. YAHUA, MANAL. Augmented reality based on human needs. 2022. Diss.

1036. KARUPPANNAN GOPALRAJ, SANKAR. Impacts of recycling carbon fibre and glass fibre as sustainable raw materials for thermosetting composites. 2022. Diss.

ISBN 978-952-335-848-5
ISBN 978-952-335-849-2 (PDF)
ISSN 1456-4491 (Print)
ISSN 2814-5518 (Online)
Lappeenranta 2022