Implementation and evaluation of the DAOM framework and support tool for designing blockchain decentralized applications

Chibuzor Udokwu¹ · Patrick Brandtner² · Alex Norta¹ · Alexandr Kormiltsyn¹ · Raimundas Matulevičius³

Received: 23 June 2021 / Accepted: 27 September 2021 / Published online: 21 October 2021
© The Author(s) 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

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

✉ 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

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 exists 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 Architect² 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
Relationship	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
	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 of the 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
Productivity	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, realism 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 free-hand tool, the modelling task is reported as being difficult to complete. The results confirm that the automations 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 realism 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.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Atkinson C, Kuhne T (2003) Model-driven development: a metamodelling 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