

Exploring the Systematic Design of Blockchain-based Applications Using Integrated Modeling Standards

Simon Curty^{1,*}, Hans-Georg Fill¹

¹University of Fribourg, Digitalization and Information Systems Group, Fribourg, Switzerland

Abstract

The systematic design of blockchain-based applications requires a holistic perspective covering business and IT aspects as well as their alignment. Further, multiple actors need to be involved in this process for achieving a mutual understanding of the design options and their effects on the organizational and technical requirements. In this context we explore an integrated set of state-of-the-art modeling methods for the business, business process, enterprise architecture and IT perspectives for supporting the prototypical realization of blockchain-based applications in interdisciplinary teams. We illustrate the application of the integrated methods with a use case in the area of decentralized auctioning and derive requirements for further research.

Keywords

Blockchain, Enterprise Application Design, Enterprise Modeling

1. Motivation

Blockchains offer a novel technological foundations for the digital transformation of traditional businesses as well as new opportunities due to their intrinsic qualities, such as secure, tamper-proof, and decentralized storage [1]. However, from the business perspective, the introduction of blockchain technologies in real-world scenarios is not only impeded by the comparatively low maturity of the technology in terms of scalability and interoperability with existing systems but also by concerns about costs, insufficient knowledge about the technology, architectural arrangements, and resulting benefits [2]. Thus, the major challenge in designing blockchain-based applications is to deal with the complexity in aligning institutional, market, and technology factors [3]. This includes for example the positioning on the market, the derivation of according business processes and implementation and engineering challenges unique to this technology [4]. Low maturity of the technological ecosystem – as evident by the few standards that have been established – further complicate adoption. Methodologies and software development practices for blockchain-based applications have been proposed [5], but these typically do not address the business perspective.


For dealing with the complexity in designing socio-technical systems, the field of *enterprise modeling* has developed a large number of methods for a systematic design. Many of these


PoEM'2022 Workshops and Models at Work Papers, November 23-25, 2022, London, UK

*Corresponding author.

✉ simon.curty@unifr.ch (S. Curty); hans-georg.fill@unifr.ch (H. Fill)

ORCID [0000-0002-2868-9001](https://orcid.org/0000-0002-2868-9001) (S. Curty); [0000-0001-5076-5341](https://orcid.org/0000-0001-5076-5341) (H. Fill)

 © 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

are extensively used by practitioners, e.g. for elaborating business models, analyzing business processes, or deriving IT architectures [6]. It therefore seems obvious to investigate how enterprise modeling methods may support the design of blockchain-based applications. This will be investigated in the following along two research questions: "Which existing enterprise modeling methods may be used for achieving a holistic design of blockchain-based applications?" and "How can such enterprise modeling methods be integrated?".

For answering these research questions we revert to an exploratory and experimental approach involving a recent literature analysis on modeling methods for blockchain-based application design [7] as well as a review of state-of-the-art enterprise modeling methods for the business, process, enterprise architecture and IT perspectives, cf. [8]. Thereby, we will focus on modeling methods that have proven successful in industry and are thus considered as state-of-the-art both in academia and practice. We will then identify possible integration points for the modeling methods and apply the approach to a use case in the area of decentralized auctions. From the insights gained thereby we will derive future requirements for further research in this area.

The remainder of this paper is structured as follows. In Section 2 we will introduce foundations on enterprise modeling and the model-driven development for blockchains. In Section 3 we will derive an integrated set of enterprise modeling methods, which will be applied to a use case for designing blockchain-based applications in Section 4. The paper will conclude with a discussion of the approach in Section 5 and an outlook to further research.

2. Foundations

In this section we briefly outline the foundations necessary for describing our approach. This includes previous contributions of the field of enterprise modeling, as well as recent findings on other approaches for the model-driven engineering of blockchain-based applications.

2.1. Enterprise Modeling

A multitude of languages have been proposed for either covering *particular aspects* in enterprise modeling, such as for example goals elicitation [9] or business process improvement [10]. Other directions follow a *holistic, integrated view*. For this latter case, approaches have been developed that are either inherently integrated, such as the Semantic Object Model [11] or ArchiMate [12], or that combine multiple sub-languages, such as in MEMO [13] or 4EM [14].

Whereas many approaches have been developed in academia, only some of these have been directly taken up by industry. As of today, practitioners often revert to languages that implement *international standards* such as BPMN¹, UML², or ArchiMate³. Using such standards has the immediate advantage of interoperability across organizations and the supporting tools. Frequently however, the integration of the languages is not considered, despite existing proposals in the literature, e.g. [8]. One reason may be the large complexity of each individual standard and thus the effort in determining suitable reference points between them.

¹<https://www.bpmn.org/>

²<https://www.uml.org/>

³<https://publications.opengroup.org/standards/archimate>

2.2. Model-Driven Engineering for Blockchain-based Applications

For the domain of blockchains, enterprise modeling approaches have been proposed that focus specifically on the concepts of distributed ledgers and smart contracts [15]. In a recent survey of model-driven engineering and low-code approaches for blockchain-based applications [7], it could be shown that existing solutions predominantly focus on the technical level, e.g., the model-based generation of smart contract code. Holistic approaches integrating business, application and technological aspects are sparse and remain on an abstract level. However, the introduction of blockchain technologies affects all levels of an organization, e.g., when integrating business models with smart contract transactions on a technical level or for reaping the benefits of technical decentralization on the business level [16]. Thus, we find that for dealing with the added complexity, a systematic and holistic view is instrumental for the successful design of blockchain-based applications.

3. Deriving a Set of Integrated Modeling Standards

Enterprise modeling languages target various aspects of an organization, each providing a particular perspective on different levels of abstraction [13, 14]. Combining multiple enterprise modeling languages allows for deriving a more complete picture of business and IT aspects. This eases the understanding of the relations and inter-dependencies between low and high level concepts, e.g., how the implementation of a software artifact relates to a business process. This is especially useful for the development of blockchain-based applications as their technological complexity, low maturity, diversity of the ecosystem and largely absent standards impede successful adaption in business models.

As a means to deal with these issues we explore a procedure based on an integrated set of standard modeling methods which enables the collaboration of experts from different fields. It is inspired by common top-down methods used in enterprise modeling, e.g. [14, 11]. We illustrate for each step how it can be applied for the design of blockchain-based applications. We revert to business transaction models for creating business model canvases [17], BPMN, ArchiMate, and UML. Thus, the approach is based on up-to-date, proven and industry-accepted modeling methods. The procedure guides the design experts step-by-step through the tasks of assembling a comprehensive view of the business and technical levels - see Figure 1. In the following, we first elaborate the procedure and subsequently describe the used methods.

First, a business model canvas focusing on blockchain-specific properties is designed using business transaction models. Alternatively, an existing model is adapted for the integration of blockchain aspects. The business model describes mainly the value propositions, exchanges, and key activities an enterprise has to perform. From these, the main business processes can be derived and further detailed using BPMN. Supporting these processes requires appropriate application services, identified on the basis of the process model. We then proceed to designing the enterprise architecture to gain a holistic, high level view of the previously elaborated business processes, supporting applications and required IT infrastructure. Alternatively, the identified application services may be integrated into an existing enterprise architecture. For this we revert to the ArchiMate standard. The application services are then further detailed using various UML diagrams. These models subsequently serve as basis for the technical prototyping.

This procedure should be understood as inherently iterative as the adoption of novel technologies such as blockchains can affect all organizational levels and changes on one level can have implications for others. For example, the properties of a selected blockchain platform may render the business model non-viable due to its technical limitations. On the other hand, a business model may impose technological requirements, which lead to the selection of a particular blockchain system with appropriate properties. The procedure concludes upon the successful development of a viable prototype for a given business model. Following this, the development of the actual software application may continue with any state-of-the-art development methodology, e.g., the so-called "ABCDE" (sic!) method [5].

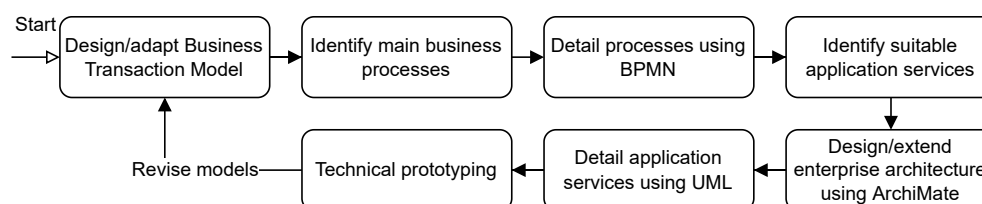


Figure 1: Procedure guiding designers from business modeling to prototyping.

3.1. Business Perspective

Several modeling methods are available that target aspects such as strategies, goals, performance management, or the assessment of required and existing capabilities, e.g., *i** [9] or 4EM [14]. One of the potentially most important aspects on this level of abstraction is the definition of the business model, which specifies how value is generated by an organization and how it interacts with its stakeholders. In this context, a method that is frequently used in practice is the Business Model Canvas (BMC) [18]. It is used to identify the core business functions of companies without disregarding organizational complexities. A BMC is a uniform description of a business model in the form of strategic actions assigned to one of nine groups. A main goal of a BMC is to facilitate understanding, discussion and ultimately exploration of the underlying business model. The groups form the building blocks of the canvas, which are populated in a standardized order with fitting concepts relevant to the business model. We briefly summarize the characteristics of three groups that we will revert to later for the integration:

Key Activities describe the essential activities an organization must undertake to enable the business model. A *Channel* describes how the organization communicates with a customer segment to provide a value proposition. *Customer Segments* represent various groups the organization aims to serve. The description of the remaining groups can be found in the literature [18].

A limitation of the BMC is the lack of relations. Thus, it is not clear how identified elements relate to each other by means of a BMC alone. In order to specify the relations between elements, we use the technique of business transaction models (BTM) [17]. Thereby, a model is created in a fashion akin to a BMC, by first identifying concepts for each group. In a next step, the designer specifies the relations between the elements. No constraints are imposed on the allowed relations, they just add semantic information by means of their type. The introduced relation types are "causes", "delivers value", "is offered", "is part of", "is used" and "pays".

3.2. Business Process Perspective

For modeling the main business processes, we apply the Business Process Model and Notation (BPMN) in version 2.0⁴. While the discipline of business process management knows multiple modeling approaches [19], BPMN is widely accepted and a world-wide standard in business process management [20]. The expressiveness of BPMN allows it to be used for modeling any type of process, i.e., technology processes and execution specifications can be modeled as well as business processes. We apply BPMN for detailing the processes associated with the *key activities* of the business model. This may also include the interaction with application services. These services may be characterized by interactions between participants. E.g., a message exchange of users and a system involving different pools. Our methodology does not specify how a business process should be modeled in detail, rather we leave this decision to the process design expert.

3.3. Enterprise Architecture Perspective

Enterprise Architecture (EA) is the discipline of holistically analyzing and documenting the structure, operation and inter-dependencies of an organization across business domains [21]. We revert to the ArchiMate standard [12], which defines a visual modeling language for enterprise architecture where EA perspectives are represented as layers. The choice to revert to ArchiMate for EA modeling as opposed to alternative approaches is motivated by its relevance in industry with numerous practitioners⁵ and widespread tool support. The core specification⁶ defines three layers: (i) on the *business layer* lie technology-independent organizational concepts related to the business model. The *application layer* (ii) models the behavior, structure and relationships of the organization's applications. Lastly, the *technology layer* (iii) models the organization's underlying IT infrastructure. ArchiMate distinguishes between behavioral and structural elements, many of which are common across layers but with layer-dependent semantics. E.g., a process element represents a business process on the business layer but a software process on the application layer. Layers and elements for modeling strategic perspectives, organizational goals and motivations are available outside the core specification. We focus solely on the core parts as they contain the concepts frequently applied. In the context of the procedure, an existing EA maybe be extended with the new application services.

3.4. IT Perspective

For the specification and documentation of the application implementation we revert to modeling languages of the Unified Modeling Language (UML). In software development, UML is a widely known and recognized standard for the model-based description of systems. We make use of a subset of UML diagrams for the structural and behavioral modeling of the blockchain-based software application and the system boundaries. First, we revert to *UML use case* diagrams to specify the main functionalities and features of the software application from the perspective of the core actors. Second, *UML class diagrams* allow to draft, document and reason on structural details of the implementation. For the purpose of blockchain-based application development,

⁴<https://www.omg.org/spec/BPMN/2.0.2/>

⁵<https://archimate-cert.opengroup.org/certified-individuals>

⁶<https://pubs.opengroup.org/architecture/archimate3-doc/toc.html>

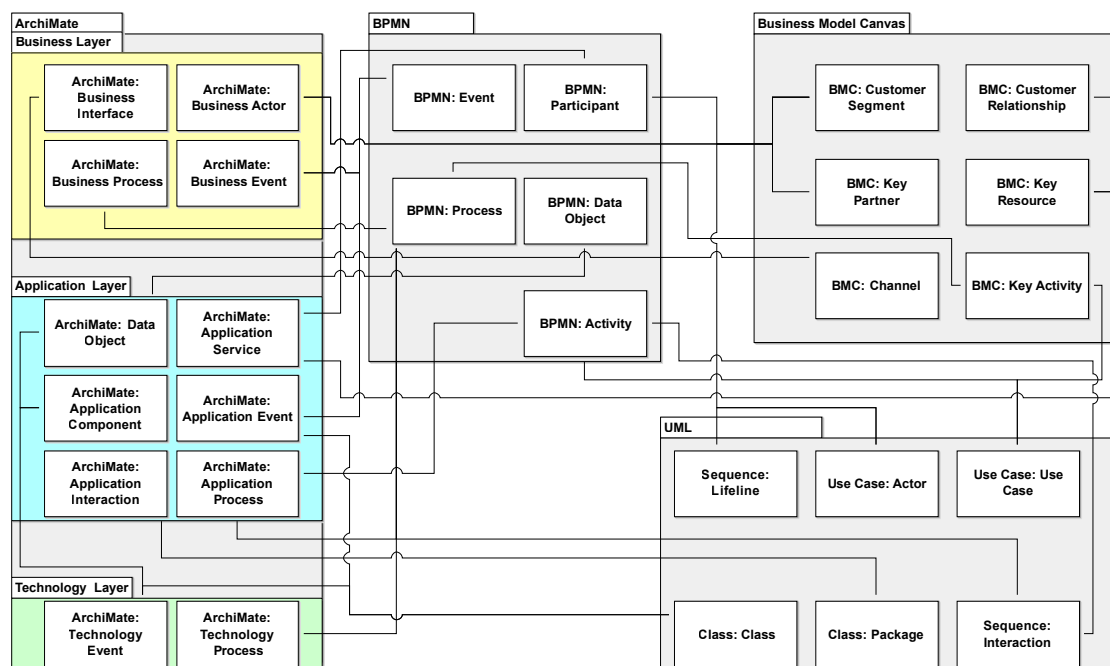


Figure 2: *Integration points as references between elements of the different models.*

a major concern is the reasoning about the system boundaries and the inter-connection, i.e., which software components are dependent or communicate with on-chain systems. As such, it lends itself to model smart contracts as classes alongside the off-chain components, however clearly separated by specifying packages. Lastly, interactions with and between on-chain smart contracts are a critical transactional part of a distributed application. *UML sequence diagrams* are well suited for detailing messages, such as function calls or requests and responses, and their ordering in such critical system interactions.

3.5. Model Architecture for Integration

For integrating the different modeling languages, we propose a loose coupling through inter-model references which we denote as *integration points* between model elements, excluding relation types - see Figure 2. An integration point is a reference between elements of different diagrams. Thereby, no restriction is placed on the number of references. Using these references one may navigate then between diagrams and elements. For example, a lifeline in a sequence diagram may reference a BPMN participant and/or a key partner defined in a BMC. This is however not an integrated meta-model or a one-to-one concept mapping of the different modeling languages. These integration points have been derived based on various sources. The ArchiMate specification addresses the relationship to other standards⁷, notably to BPMN and UML. Since ArchiMate is a holistic modeling language many of the concepts can be found in others. E.g., it is possible to model a business process in ArchiMate but not in such detail as

⁷https://pubs.opengroup.org/architecture/archimate3-doc/apdx.html#_Toc10045525

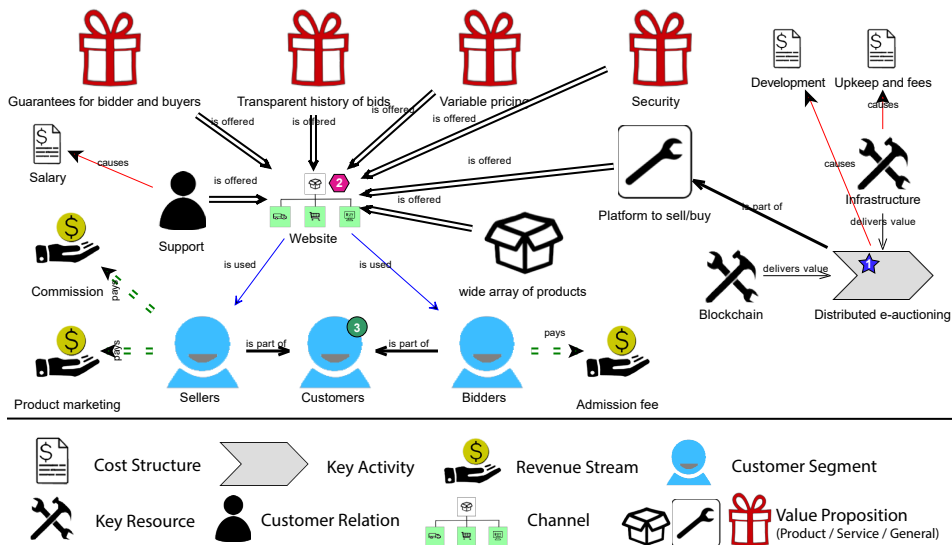


Figure 3: Business transaction model of a blockchain-based auction business model.

with BPMN. Many elements of the application and technology layer are derived directly from UML. Further, we consider the proposed mapping of business model canvas, BPMN and UML elements to ArchiMate by Lankhorst et al. [8]. Our integration points for business transaction models are similar to these mappings. However, we do not regard ArchiMate's strategy and motivation layer, since these are not part of the core specification. The proposed integration points for UML concepts are less restrictive than a mapping as suggested by Lankhorst et al. This is due to the fact, that there are many ways to model a UML diagram for parts of software systems. The designers should not be restricted in modeling by the proposed integration points. The benefit of such an integration is that it is thus defined how the concepts in the different perspectives relate to each other.

4. Use Case for Decentralized Auctions

For demonstrating our methodology, we examine a use case where a traditional auction house offers a blockchain-based e-auctioning system to its customers. Thus, we explore an evolution of a successful business model of web-auctioning as illustrative example. Elements referencing each other are marked with numbered icons in the text and figures. A prototype implementation of the application is available for the Ethereum blockchain⁸.

As a first step, the use case is modeled as a BTM as depicted in Figure 3. This provides a high level view and offers insights into the changes introduced through a new technology and how the traditional business model may be changed or disrupted. Here we consider the benefit of blockchains for offering transparent and provable auctioning to the customers, where each bid is transactionally recorded and verifiable. This results in new value propositions for the customers but poses the challenge of integrating blockchain technology into an existing system.

⁸Source code: <https://doi.org/10.5281/zenodo.7304990>

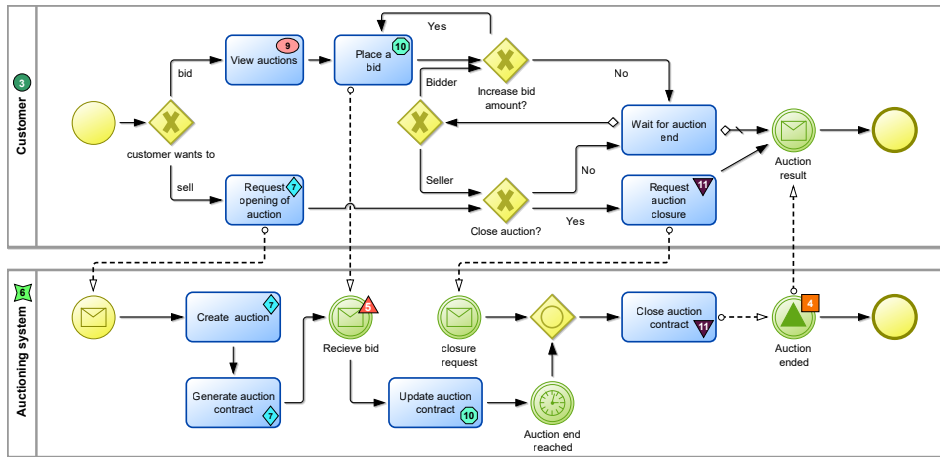


Figure 4: BPMN diagram of the auctioning process, corresponding to the key activity reference 1 (★).

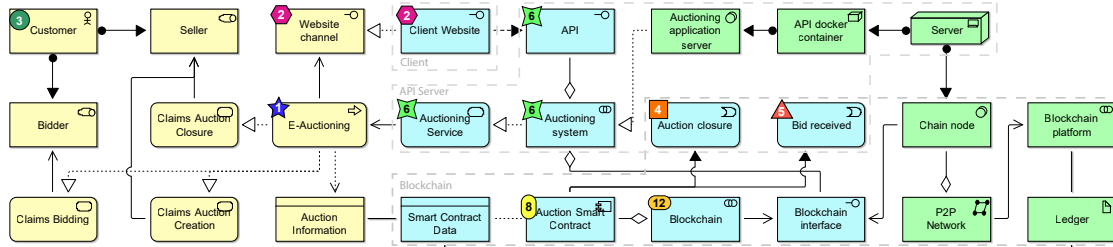


Figure 5: ArchiMate model of the enterprise architecture showing the business, application, and technology layers of the envisioned application.

The major key activity is the provision, development and maintenance of the e-auctioning service, which is offered through a website to sellers and bidders. From this model we can determine the major business process, i.e., for conducting auctions, as well as the central actors. Taking into account the proposed integration points, we continue to detail the key activity (ref. 1 ★) in BPMN, as shown in Figure 4. A customer (ref. 3 3) may either take part in an auction as a bidder or open a new auction as the seller. Upon requesting the opening of a new auction the software system generates a contract representing this auction and its state. Valid bids on open auctions are recorded in the respective contract. An auction is closed either manually by the seller or after a set duration. Finally, the auctions outcome is reported. To realize this process, an auctioning software system as well as supporting IT infrastructure is required. Thus we identify the BPMN participant "Auctioning System" (ref. 6 6) as a main application service.

To draft a holistic high level view of these dependencies, we create an ArchiMate model in a top-down fashion. Thereby, we regard the BPMN model and follow integration points to capture relevant concepts. The auctioning service (ref. 6 6) is realized by a system consisting of an API server and a client web interface. The ArchiMate model in Figure 5 shows the relation of the business process and the supporting applications and infrastructure. On the business layer, a simplified view of the business process is shown (ref. 1 ★). This process is enabled by UI functionality in the form of a website – the main channel as drafted in the BTM (ref. 2 2).

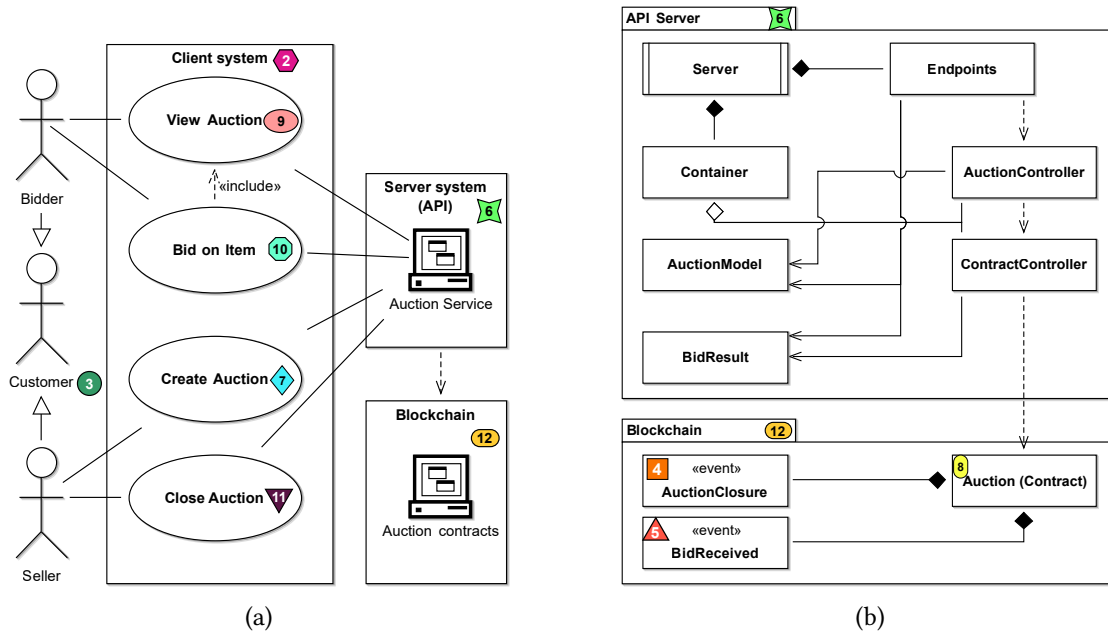


Figure 6: UML use case (a) and class diagram (b). The use case diagram depicts the client system as realization of the channel (2) and the API server (6). The API server and the smart contract are further detailed by the class diagram.

The website communicates with the auctioning system as represented on the application layer. Here, the auctioning system is run on an application server as a container deployed on a physical or virtual server. This server also operates a blockchain node, thereby participating in a blockchain network and serving as interface between on-and off-chain application components. The smart contract (ref. 8 (8)) realizing the auctioning logic is modeled on the application layer as a component together with the related (persistent) auction data that is stored on the ledger. This is a direct representation of the information required to support the business process as modeled by the business object ("Auction Information") on the business layer.

Following the methodology, the next step is to detail the application specifications by the use of UML. First, the requirements and features of the envisioned auctioning system from a customer perspective are captured in a use case diagram, as shown in Figure 6a. In This shows the supported users actions and how these relate to the major application components. The application structure is further detailed by means of a UML class diagram as shown in Figure 6b. For brevity, only the API server (ref. 6 (6)) and the smart contract (ref. 8 (8)) are shown in a conceptual view. Here, another integration point is followed through the models: The BPMN model has events for receiving a bid (ref. 5 (5)) and the auction end (ref. 4 (4)). These are represented as application events in the ArchiMate model triggered by the smart contract. Key application functionality can then further be specified by a UML sequence diagram. However, this is omitted for brevity. On basis of these models, one can then proceed to implement a technical prototype. This provides further insight on consequences for the business model and architecture caused by technical limitations.

5. Discussion

The application of the integrated modeling methods in a blockchain use case provided us with insights into advantages and limitations of the proposed approach. The procedure results in a holistic view on blockchain design in an enterprise, starting from business models, over business processes, the enterprise architecture and finally the requirements specification and implementation of the software artifacts. This allows for an alignment of the business and IT aspects represented by the different models. By use of wide-spread standard modeling languages, the approach can be adopted easily by teams of interdisciplinary design experts. Furthermore, using standard modeling languages ensures interoperability between organizations and tools and thus eases the exchange of information. Each language has a dedicated focus and provides a unique perspective. This offers the designers to represent every aspect of blockchain application design in detail. Thereby, the procedure and proposed integration points do not restrict how specific concepts of the application can be modeled – but offer a guided process informing the various designers on how to apply the modeling languages and assist in structured reasoning on the application at hand. Thereby, the references guide from one model type to another. Navigating along suitable model elements highlights potential issues such as the absence of certain concepts. While the proposed approach is suitable for blockchain application design, it still requires detailed knowledge on distributed ledger technologies. However, this modeling approach can facilitate discussion and knowledge exchange between experts of different fields, e.g., blockchain, business and EA experts.

The chosen languages are expressive and offer flexibility in modeling. However, the details of each of the languages also contribute to the complexity of the approach. The learning curve can be steep if designers are not yet familiar with a language. This is somewhat mediated by the use of standardized languages as many designers may have existing experience with one or several of the languages. Moreover, each model is typically created by a domain expert. In this case, the integration points help to align the various models and facilitate the understanding of how the perspective of other experts relates to one's own. E.g., an EA expert needs to react to, communicate on, and decide on architecture design decisions, resulting from design decisions and changes made by a business process expert. To reduce complexity of the overall modeling approach, simpler languages with fewer elements could have been chosen, e.g., e^3 value for business models.

There is no fully integrated tool support yet available for the proposed approach. Currently, several different tools are required. However, this permits the designers to chose their preferred tools. Other approaches such as MEMO, 4EM that do not revert to standardized modeling languages can offer a full integration at the expense of limited interoperability.

Concepts specific to blockchain scenarios and use cases are not yet present in the chosen modeling languages. Extending these languages with such concepts would further ease the derivation of blockchain-specific business, enterprise architecture and software models. Such domain-specific modeling would further profit from a unified metamodel of the blockchain domain.

6. Conclusion and Further Research

In this paper we proposed an iterative procedure for the holistic prototypical design of applications based on an integrated set of standard modeling languages. Further, we presented so-called integration points for linking the different model types. This facilitates reasoning about inter-dependencies between concepts present across models and eases communication between domain experts. We demonstrated the applicability of this procedure to the field of blockchain application design by means of a use case in the context of electronic decentralized auctions. In this way we achieve an alignment along the perspectives of the models.

Future work will focus on the development of domain-specific languages or profiles for modeling blockchain-related aspects. For this, we are in the process of evaluating the results of a course where interdisciplinary teams have applied the proposed method to design and implement prototypes for various blockchain use cases. Integration of existing domain-specific profiles and language extensions could additionally enrich the approach, e.g., by including e³value modeling supporting distributed actors [22] or UML profiles for smart contract modeling [5].

7. Acknowledgments

This research was partially funded by the Swiss National Science Fund grant number 196889.

References

- [1] H. Fill, Enterprise modeling: From digital transformation to digital ubiquity, in: Proceedings of the 2020 Federated Conference on Computer Science and Information Systems, volume 21 of *Annals of Computer Science and Information Systems*, 2020, pp. 1–4. doi:10.15439/2020F001.
- [2] S. Flovik, R. A. Moudnib, P. Vassilakopoulou, Determinants of blockchain technology introduction in organizations: an empirical study among experienced practitioners, *Procedia Computer Science* 181 (2021) 664–670. doi:10.1016/J.PROCS.2021.01.216.
- [3] M. Janssen, V. Weerakkody, E. Ismagilova, U. Sivarajah, Z. Irani, A framework for analysing blockchain technology adoption: Integrating institutional, market and technical factors, *International Journal of Information Management* 50 (2020) 302–309. doi:10.1016/j.ijinfomgt.2019.08.012.
- [4] N. Kannengiesser, S. Lins, C. Sander, K. Winter, H. Frey, A. Sunyaev, Challenges and common solutions in smart contract development, *IEEE Transactions on Software Engineering* (2021). doi:10.1109/TSE.2021.3116808.
- [5] L. Marchesi, M. Marchesi, R. Tonelli, ABCDE—agile block chain DApp engineering, *Blockchain: Research and Applications* 1 (2020). doi:10.1016/j.bcra.2020.100002.
- [6] F. Wolff, M. A. Jeusfeld, A. Persson, Explorative survey into goals, focus, experiences and extent of enterprise-wide process modelling in practice, in: *The Practice of Enterprise Modeling*, Springer, 2016, pp. 272–285. doi:10.1007/978-3-319-48393-1_19.
- [7] S. Curty, F. Härer, H. Fill, Blockchain Application Development Using Model-Driven Engineering and Low-Code Platforms: A Survey, in: *Enterprise, Business-*

- Process and Information Systems Modeling, Springer, 2022, pp. 205–220. doi:10.1007/978-3-031-07475-2_14.
- [8] M. M. Lankhorst, A. Aldea, J. Niehof, Combining archimate with other standards and approaches, in: *Enterprise Architecture at Work: Modelling, Communication and Analysis*, Springer, 2017, pp. 123–140. doi:10.1007/978-3-662-53933-0_6.
- [9] E. S. Yu, Social modeling and i*, in: *Conceptual Modeling: Foundations and Applications: Essays in Honor of John Mylopoulos*, Springer, 2009. doi:10.1007/978-3-642-02463-4_7.
- [10] F. Johannsen, H. Fill, Meta modeling for business process improvement, *Bus. Inf. Syst. Eng.* 59 (2017) 251–275. doi:10.1007/s12599-017-0477-1.
- [11] O. K. Ferstl, E. J. Sinz, D. Bork, Tool support for the semantic object model, in: *Domain-Specific Conceptual Modeling, Concepts, Methods and Tools*, Springer, 2016, pp. 291–310. doi:10.1007/978-3-319-39417-6_13.
- [12] M. M. Lankhorst, H. A. Proper, H. Jonkers, The architecture of the archimate language, in: *Enterprise, Business-Process and Information Systems Modeling*, Springer, 2009, pp. 367–380. doi:10.1007/978-3-642-01862-6_30.
- [13] U. Frank, Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges, *Software and Systems Modeling* 13 (2014) 941–962. doi:10.1007/s10270-012-0273-9.
- [14] K. Sandkuhl, J. Stirna, A. Persson, M. Wißotzki, *Enterprise Modeling - Tackling Business Challenges with the 4EM Method*, The Enterprise Engineering Series, Springer, 2014.
- [15] H. Fill, P. Fettke, S. Rinderle-Ma, Catchword: Blockchains and enterprise modeling, *Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model.* 15 (2020) 16:1–16:8. doi:10.18417/EMISA.15.16.
- [16] F. Härer, Process Modeling in Decentralized Organizations Utilizing Blockchain Consensus, *Enterprise Modelling and Information Systems Architectures* 15 (2020) 13:1–17. doi:10.18417/emisa.15.13.
- [17] M. Wieland, H. Fill, A Domain-Specific Modeling Method for Supporting the Generation of Business Plans, in: *Modellierung 2020*, volume P-302 of *LNI*, Gesellschaft für Informatik e.V., 2020, pp. 45–60. doi:10.500.12116/31846.
- [18] A. Osterwalder, Y. Pigneur, *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*, John Wiley and Sons, 2010.
- [19] J. Recker, M. Rosemann, M. Indulska, P. Green, Business Process Modeling- A Comparative Analysis, *Journal of the Association for Information Systems* 10 (2009) 333–363. doi:10.17705/1jais.00193.
- [20] M. Chinosi, A. Trombetta, BPMN: An introduction to the standard, *Computer Standards & Interfaces* 34 (2012) 124–134. doi:10.1016/j.csi.2011.06.002.
- [21] M. M. Lankhorst, Introduction to Enterprise Architecture, in: *Enterprise Architecture at Work: Modelling, Communication and Analysis*, The Enterprise Engineering Series, Springer, 2009, pp. 1–11. doi:10.1007/978-3-642-01310-2_1.
- [22] S. Perrelet, H. Fill, J. Dibbern, A Modeling Approach for Blockchain-inspired Business Models: An Extension of the e3 -Value Method, in: *55th Hawaii International Conference on System Sciences*, 2022. doi:10.24251/hicss.2022.558.