Integrating BIM- and Cost-included Information Container with Blockchain for Construction Automated Payment using Billing Model and Smart Contracts

Xuling Ye, Katharina Sigalov and Markus König

Department of Civil and Environmental Engineering, Ruhr-University Bochum, Germany
E-mail: xuling.ye@rub.de, katharina.sigalov@rub.de, koenig@inf.bi.rub.de

Abstract –

Payment issues are necessary in the construction industry, often manifested by high levels of arrears and long-term payment delays. An automated payment could be a solution to speed-up the payment process after successful completions. In this paper, a framework is proposed for the automated payment via Building Information Modeling (BIM), Linked Data, Smart Contract and Blockchain technologies. Geometrical and payment-related data are stored as BIM-based linked data. Linked Data technology connects a BIM model with its related Bill of Quantities (BoQ) and Quantity Take-Off (QTO). Based on the transparency, traceability and collaboration of the Blockchain technology, an automated payment can be secured. To integrate such an information container with Blockchain for the automated payment, an additional information model called Billing Model (BM) is proposed. Thereby, each Billing Unit (BU) stores one or more construction work tasks with a related payment and a related due date. Together with the corresponding Smart Contract rules, the BM can automate payment via a Blockchain. Several aspects of improvement and further development are discussed in the end.

Keywords –

Billing Model; Building Information Modeling (BIM); Information Container; Blockchain; Smart Contracts

1 Introduction

Low margins and relatively low productivity have been the issues of the construction industry for decades [1]. The main reason for this lies in the low degree of digitization within the construction sector despite the increasing scale and complexity of construction projects. Relying mainly on drawings or paper-based documentation does no longer satisfy the increasing needs of construction projects. In addition, unsatisfactory software interoperability in the Architecture, Engineering and Construction (AEC) industry makes coordination between clients and contractors unnecessarily difficult. As a result, no single source can provide an integrated, real-time view of project design, costs or schedules. To make things worse, Caldas et al. [2] state that a large amount of construction documents, which necessarily contain graphical and non-graphical information that must be communicated between project team members, are needed to describe the characteristics, specifications and execution plans of building products, such as drawings, specifications, schedules and cost estimates.

These problems of contract management in construction can be reduced by applying newly available digital technologies and working methods. Building Information Modeling (BIM) has become a widely accepted solution for making the whole lifecycle of a construction project digital. By integrating “cost” into a BIM model, budget and cash flow planning and cost controlling can be enabled. After the implementation of smart contract applications by Ethereum [3], non-currency data can be stored in the Blockchain. Based on the transparency, traceability and collaboration of the Blockchain technology, it could be a good solution to enable digitalization of the contract management and the automation of the payment process.

However, because of the traceability of Blockchains, the data stored in a Blockchain will be repeatedly copied and stored in different blocks, even when modifications are only slight. As a result, the data storage of Blockchains is extremely expensive [4]. Moreover, because of the consensus mechanism and verification of Blockchains, the transaction speed is relatively slow compared to central network systems. It is time-consuming and arduous to store all the data of a construction project entirely in the Blockchain and therefore not worth the effort.

According to the points mentioned above and the available technologies, this paper presents a concept for the generation and application of an information container (BIM Contract Container or BCC for short), which is to be used as a basis for automatic payment transactions in the construction industry. For this purpose, a billing model is introduced, which is the basis for the creation of a smart contract.

2 Related Work

2.1 Digital Models in Construction

By integrating "cost" into the BIM model, a cost budget and the corresponding financial representations can be instantly generated. The purpose of using BIM for cost is to provide accurate and reliable cost estimation based on the digital 3D model, and eliminate errors caused by manual measurements or estimations [5]. To exchange digital building models for building constructions, the Industry Foundation Classes (IFC) were developed and standardized internationally by buildingSMART [6]. With the help of the IFC, geometric and semantic building information can be exchanged independently of the manufacturer. The Bill of Quantities (BoQ) are created for the tendering and awarding of construction works [7]. An international standard for the description of BoQ information does not yet exist. Nevertheless, there is a standardized format for data exchange, which was created in Germany by the Joint Committee Electronics in Construction (GAEB) with the GAEB XML standard [8]. The GAEB XML format has been proposed to serve the agreement of a German uniform exchange standard for service specification and thus to support all requirements for electronic processes concerning public notice, contracting and accounting in the execution of construction works [8]. The building model can be used to determine quantities that are assigned to the individual BoQ elements. The calculation of the individual quantities based on the components of the BIM model is called Quantity Take-Off (QTO).

In recent years, some formats have been developed in order to use and exchange information from various data sources (often files). For this purpose, the individual data sources are described and linked using metadata. The links are saved explicitly. Data sources, metadata and links are generally referred to as information containers. In the simplest case, an information container comprises only one data source without links. If the data sources are in the form of files, the data exchange of an information container is realized as an archive file (e.g. as a zip file). The BIM-LV container according to DIN SPEC 91350, for example, enables standardized data exchange between digital building models and digital service specifications [9]. DIN EN ISO 21597-1 defines a similar approach to building and exchanging linked information using linked data concepts [10]. An information container according to DIN EN ISO 21597-1 is described using the Resource Description Framework (RDF) [11]. ISO 21597-1 was developed for the standardized data exchange of linked data sources using a generic information container format as shown in Figure 1. A container includes a header file "index.rdf" in the top-level folder. A container shall have at least three folders. The "Ontology resources" folder may be used to store ontologies providing the object classes and properties that shall be used to specify the contents of and links between the documents within the container. The "Payload documents" folder and "Payload triples" folder may contain nested folders to allow groups of associated digital resources to be held together and referenced as a group (e.g. a building information model with its associated reference files or a set of linked spreadsheet documents). A container according to ISO 21597-1 can also be used to exchange data and forms the information basis for the approach presented here. Building models, BoQs and QTOs are linked with the help of a container and used in the creation of a billing model.

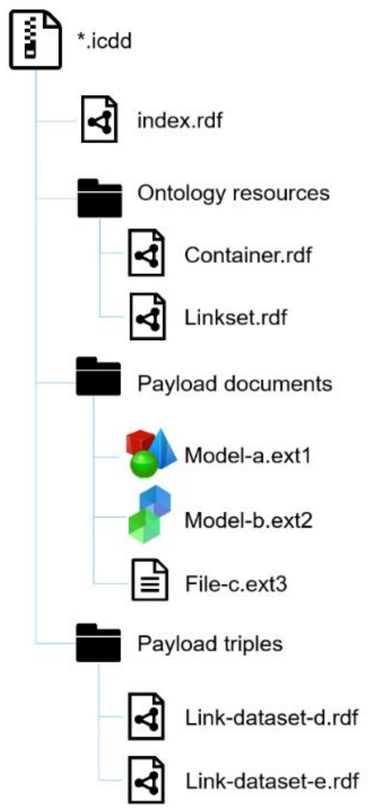

Figure 1. The structure of an information container according to DIN EN ISO 21597-1

2.2 Blockchain and Smart Contract

Blockchain is a distributed ledger technology, where "distributed" is managed by a decentralized peer-to-peer (P2P) network [12] and "ledger" is realized as an electronic database comprising a chain of blocks [13]. In this network, the transactions are verified, validated and combined into blocks. The blocks are then chained together. Thus, the procedure is based on a chained block structure that grows linearly. The data in the Blockchain is immutable and cannot be manipulated or deleted. Blockchain has the following key characteristics: decentralisation, persistency, anonymity and auditability [14].

The Blockchain concept was developed from the Bitcoin technology first proposed by the pseudonymous Satoshi Nakamoto in 2008 [15]. However, the Bitcoin (i.e. the origin of Blockchain) was a non-programmable system suitable only for cash transactions. The situation starts to change after smart contracts come into the stage of Blockchain. The smart contract, first proposed by Szabo in 1996 [16], is a digital agreement which self-executes itself. Since Buterin et al. [17] have embedded smart contracts in their Ethereum Blockchain system in 2014, Blockchain has become a platform with programmable applications. This way, more possibilities and more applications in more fields than just finance have come to the Blockchain stage. However, due to the limited Blockchain storage and relatively slow transaction speed, it is does not make sense to store all the project-related data in the Blockchain as mentioned above. Many researchers suggest the notion of off-chain data storage, which is a conventional or a distributed database to store the data outside of the Blockchain. For example, Xu et al. [18] propose an architecture using Blockchain for connecting application layer within off-chain data and off-chain control.

In the construction industry, a great deal of research has been done on using Blockchains due to their enormous potential. For example, Li et al. [19] reviewed 73 papers from 2014 to 2018 on the applications of Blockchains in the construction industry. The reviewed applications lie in smart energy, smart cities and the sharing economy, smart government, smart homes, intelligent transport, building information modelling (BIM) and construction management, as well as business models and organizational structures. In 2020, Kasten [20] pointed out that research related to Blockchain applications in engineering or manufacturing has been growing fast after 2016, especially in data validity (e.g. BIM). In particular, Shojaei [21] states that smart contracts can provide a new type of work breakdown structure, which can automate the process of compliance and payment. Moreover, Li et al. [22] emphasize that smart contracts are one of the key complementary concepts to BIM, which can represent the construction project requirements in a computable way and automate contract clauses. However, Eschenbruch et al. [23] point out that using smart contracts in construction contract and payment automation still faces many legal issues.

3 Methodology

This paper proposes a framework for automated contract, invoice and billing management, as shown in Figure 2. The framework focuses on automating the payment between clients and contractors. The payment-related transactions are no longer defined in a paper-based construction contract, but rather mapped in a smart contract. Each payment-related transaction can be considered as a billing unit, and billing units must be defined in a digital way so that an automated payment can take place. For this purpose, billing units contain information about the construction work items and building elements and are based on the information from the BIM model, the BoQ and the QTO. All billing units together compose a billing plan, which together with the BIM model, the BoQ and the QTO is referred to as a billing model.

A billing model is generated accordingly, and serves as a data basis for smart contracts. The billing model, in collaboration with the smart contract, enables the automation of the delivery and acceptance process. If the client checked and confirmed a construction task reported by the contractor, the payment will be automatically transferred from the client bank to the contractor bank. A Common Data Environment (CDE), as an off-chain storage, handles all the payment-related files. For this purpose, a BIM Contract Container (BCC) is used, which contains all payment-relevant data (i.e. the BIM model, the BoQ with QTO and the billing model). For a secure and resource-efficient storage of the information in a Blockchain, only the URLs and hash values of the files stored in the CDE are used. The following three subsections detail how to filter payment-related transactions from a construction contract to a smart contract, how to generate a billing model, and how to automate the payment.

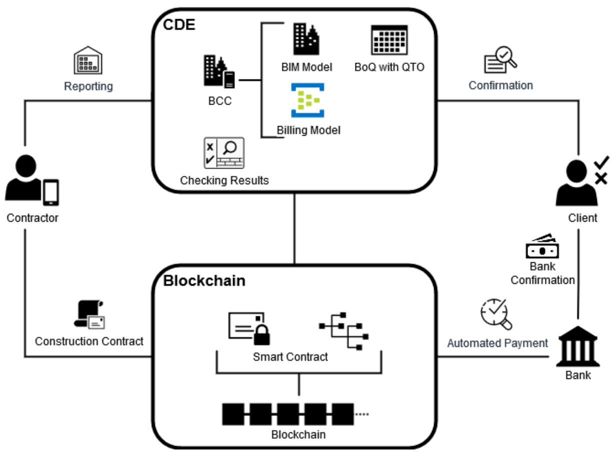

Figure 2. The overview of the proposed framework

3.1 Construction Contract to Smart Contract

Due to the complexity and the legality of a construction contract, a smart contract will not be able to replace a construction contract completely. The starting point is to filter out the payment-related transactions from the conventional construction contract. On the one hand, this is about the services and billing units to be provided and, on the other hand, the remuneration assigned to them. The payment-related data can be obtained via the BoQ and QTO. Based on a BIM model and a BoQ with QTO, a corresponding billing model is created, which is then automatically processed via smart contracts. Hash values of the smart contract, the BIM model and the BoQ with QTO are included in the paper contract to identify the billing basis of the contract.

Once confirmed, any contractually relevant data cannot be changed unnoticed. All subsequent modifications or extensions must be agreed between the contracting parties, and be stored as an update of all the former versions. The relationships from a construction contract to a smart contract can be viewed in Figure 3. With the help of completion notifications of the contractor and confirmations of the client, payment instructions are automatically sent to the involved banks and payments are performed out immediately. Such processing modalities or processes are expressed in the smart contract with if-then relationships.

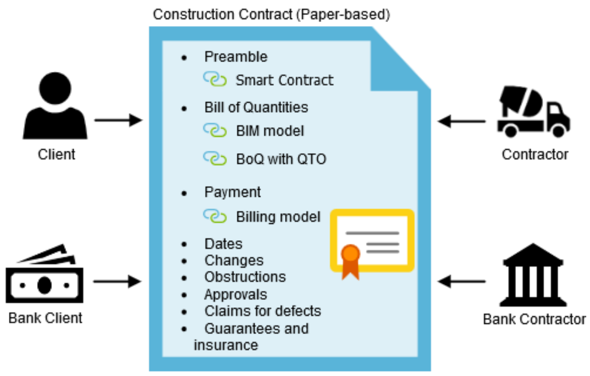

Figure 3. Integration of a smart contract into a construction contract

3.2 Billing Model

In order to automate payment transactions, a billing model has been developed based on the BIM model. Depending on the type of contract used and the underlying service description, the billing model can have different contents. In this context, a billing model is equivalent to a digital payment plan. The basis for the preparation of the payment plan is a service specification (i.e. a BoQ with QTO), which is linked to the BIM model. A service specification contains hierarchically structured elements, whereby the smallest element is a quantity split. The payment plan consists of billing units that can be clearly identified by a generated unique ID. Each billing unit contains one or more construction work items from the underlying service description with a total amount to be paid and a completion date. If the billing unit is completed and accepted on schedule, the contractor will get the agreed payment of the billing unit.

The used XML Schema Definition (XSD) of a billing model is visualized in Figure 4. A billing model consists of general billing model information (*BMInfo*) and several billing unit(s). Each billing unit contains one or more items, and the items may contain sub-items. Each item could be a single element or a group of service specification elements. In addition to the items, each billing unit stores short and long descriptions, the total quantities, the unit and the total price. The *CompletionDate* of a billing unit indicates the latest completion date of all the construction work tasks in this billing unit.

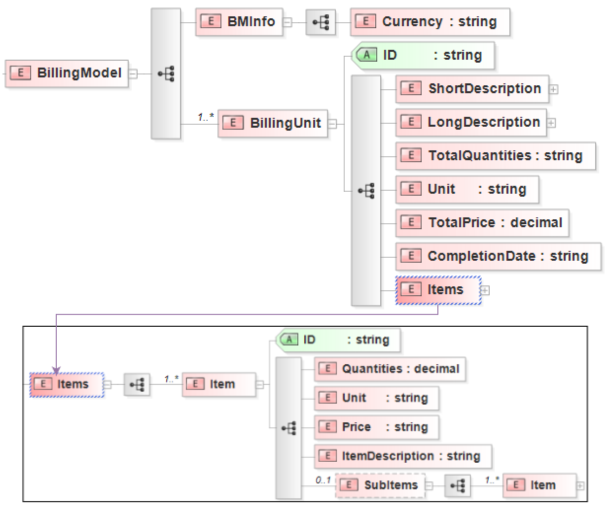

Figure 4. Visualization of the XML Schema Definition (XSD) of a billing model

To generate a Billing Model, a software tool as shown in Figure 5 has been implemented. A BIM model can be imported in the form of a BIM-LV container (Figure 5a), which contains IFC files (i.e. BIM models), GAEB files for service specification (i.e. BoQs with QTOs) and the LinkSet files to build up the links between IFC files and GAEB files. In this way, by selecting an element of the BIM model (Figure 5b), the corresponding linked BoQ item (Figure 5c) will also be selected. Multiple BoQ items in the GAEB Parser can be selected and formed as the items of a billing unit in the billing model (Figure 5d). The overall quantities, unit and total price of a billing unit are automatically generated based on the information of selected items. To successfully generate a billing unit, the

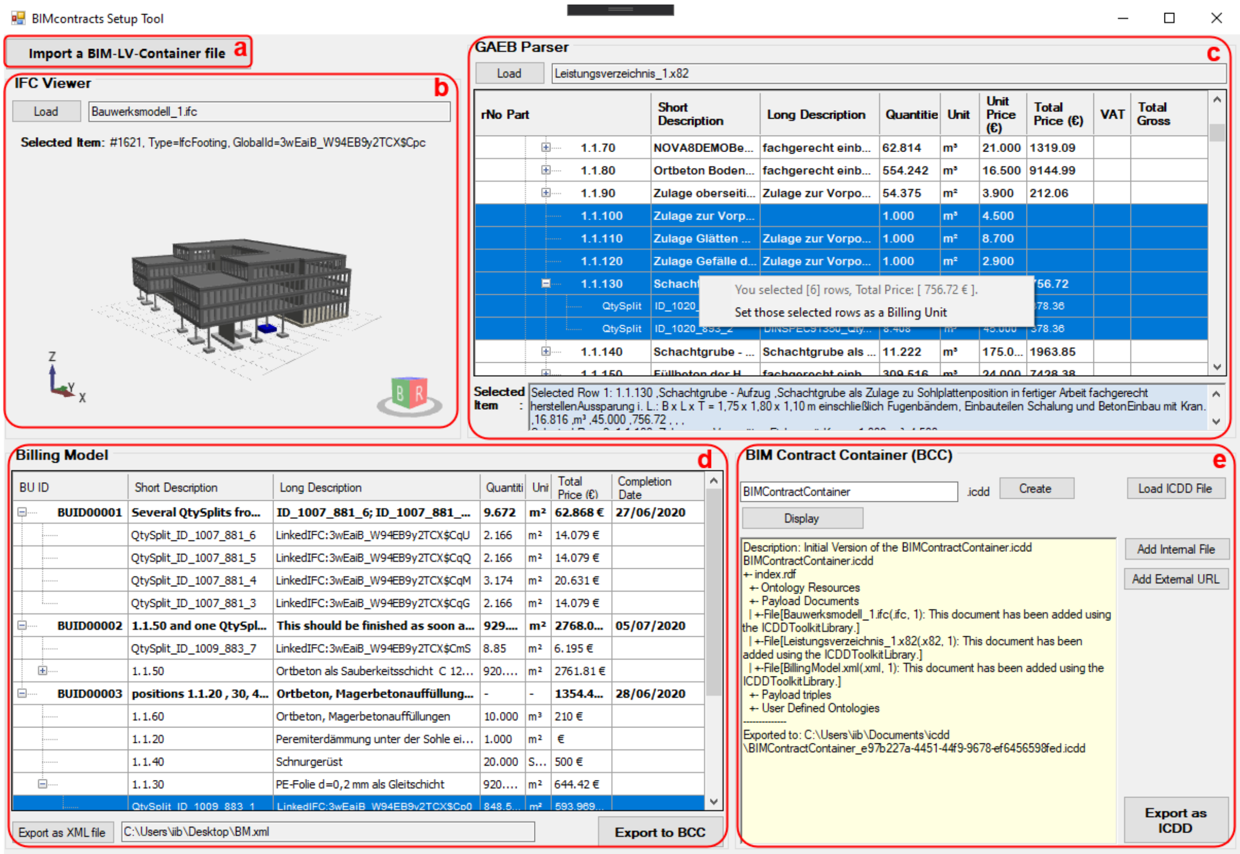

Figure 5. Implemented prototype for generating a billing model and a BIM Contact Container (BCC): (a) “Import a BIM-LV-Container file” button, (b) the IFC Viewer component, (c) the GAEB Parser component, (d) the Billing Model component and (e) the BIM Contract Container (BCC) component.

client needs to fill in the short description and the completion date of the billing unit. The long description of a billing unit is formed by the IDs of the selected items by default, which can be modified by the client. If an item is selected, all its sub-items will be selected and become the sub-items of the selected item in the billing unit (Figure 5d). When an item is linked with an IFC element, its long description in the billing model will display the *GlobalID* of the linked IFC element. For data exchange, an XML-based data format of billing models has been developed. Such an XML-format billing model can be directly exported to a BIM Contract Container (BCC) using the framework presented in Hagedorn [25]), or stored as an XML file. A generated billing unit in XML format is shown in Figure 6.


```xml
<BillingUnit ID="BUID00001">
  <ShortDescription>
    <span>Several QtySplits from 1.1.10</span>
  </ShortDescription>
  <LongDescription>
    <span>ID_1007_881_6; ID_1007_881_5; ID_1007_881_4; ID_1007_881_3; </span>
  </LongDescription>
  <TotalQuantities>9.672</TotalQuantities>
  <Unit>m²</Unit>
  <TotalPrice>62.868</TotalPrice>
  <CompletionDate>27/06/2020</CompletionDate>
  <Items>
    <Item ID="QtySplit_ID_1007_881_6">
      <Quantities>2.166</Quantities>
      <Unit>m²</Unit>
      <Price>14.079</Price>
      <ItemDescription>LinkedIFC:3wEaiB_W94EB9y2TCX$CqU</ItemDescription>
    </Item>
    <Item ID="QtySplit_ID_1007_881_5">
      <Quantities>2.166</Quantities>
      <Unit>m²</Unit>
      <Price>14.079</Price>
      <ItemDescription>LinkedIFC:3wEaiB_W94EB9y2TCX$CqQ</ItemDescription>
    </Item>
    <Item ID="QtySplit_ID_1007_881_4">
      <Quantities>3.174</Quantities>
      <Unit>m²</Unit>
      <Price>20.631</Price>
      <ItemDescription>LinkedIFC:3wEaiB_W94EB9y2TCX$CqM</ItemDescription>
    </Item>
    <Item ID="QtySplit_ID_1007_881_3">
      <Quantities>2.166</Quantities>
      <Unit>m²</Unit>
      <Price>14.079</Price>
      <ItemDescription>LinkedIFC:3wEaiB_W94EB9y2TCX$CqG</ItemDescription>
    </Item>
  </Items>
</BillingUnit>
```


Figure 6. A billing unit generated via the software tool shown in Figure 5.

3.3 BIM Contract Container (BCC)

An information container (called the BIM Contract Container, or BCC for short), which contains the list of services (i.e. the BoQ with QTO), the digital building model (i.e. the BIM model) and the agreed billing model, can be generated via the software tool (Figure 5e). The loaded IFC file and GAEB file are automatically added to the container. The agreed billing model can be exported to this container afterward (as introduced in section 3.2). In this way, a BCC is generated accordingly.

The BCC's folder structure is shown in Figure 7. The content of such a BCC is stored in a ZIP64 encoded folder with the extension .icdd. The RDF index file includes the overall description. It contains all necessary documents and link sets of the container. The building model ("BuildingModel.ifc") is considered as an internal document and therefore its referenced file is stored inside the "Payload documents" subfolder. Its assigned format is IFC2x3. The BoQ and QTO ("BoQ_gaebxml.x86") and the billing model ("BillingModel.xml") are also stored in the "Payload documents" subfolder. Apart from these documents, the container description also refers to linksets, which includes the linking for the BoQ and billing model. The contents are therefore defined in two separated files named "BoQLinks.rdf" and "BillingUnitLinks.rdf", and stored inside the "Payload triples" subfolder.

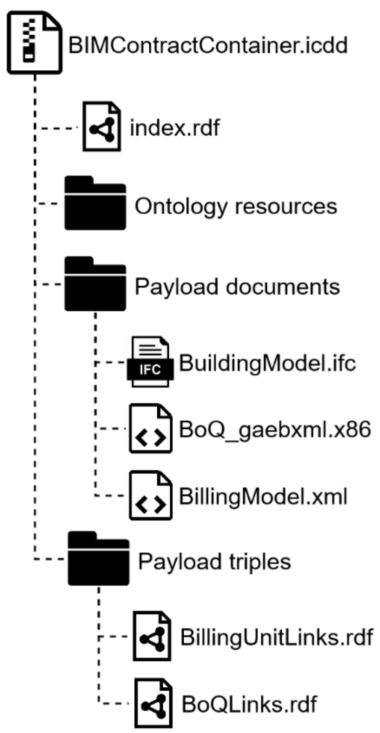

Figure 7. The folder structure of the BCC

3.4 Billing Process

Subsequently, based on the individual construction works and calculated quantities, a schedule and time planning can be made, taking into account the design and the building site. For this purpose, individual activities are defined, connected and saved as a network plan. The durations can be determined, for example, considering the selected construction methods and calculated quantities. The components of the BIM model are then linked to the individual activities using the calculated quantities.

A billing process with corresponding rules is created for each billing unit and stored in the smart contract. Such a billing process can be instantiated for each billing unit in the smart contract and linked to the digital billing unit of the billing model stored in the BCC. A simple and defect-free process for a billing unit is shown in Figure 8. The actions (start of construction works, report of the completion, checking and acceptance, execution of the payment and confirmation of the payment) are viewed as a sequential process.

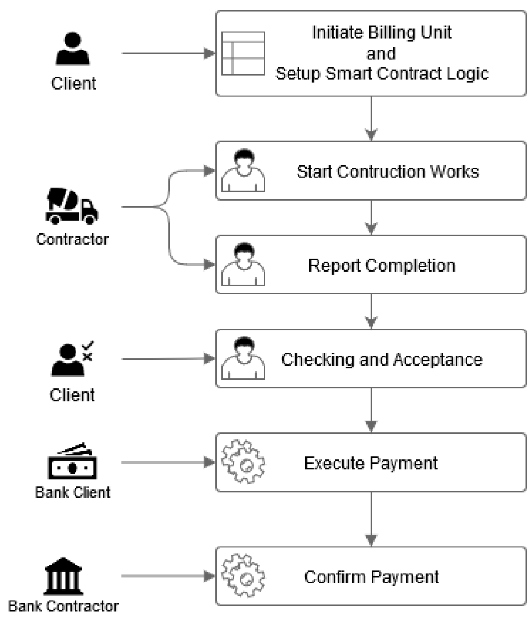

Figure 8. A defect-free process of a billing unit via BCC and smart contract logic

After the client finishes the "Checking and Acceptance" process, the possible smart contract logic as shown in Figure 9 can be executed. The values of *CompletionDate*, *isPartialPaymentAllowed*, *RejectionThreshold* and *Penalty* are predefined in the smart contract based on the construction contract and the billing unit, while the values of *FinishedDate* and *AcceptanceRate* are obtained via the actual acceptance process. When the contractor reports completion, its *FinishedDate* will be recorded. This *FinishedDate* can be compared with the *CompletionDate* set in the billing unit. The acceptance results of the client can be represented as *AcceptanceRate*. Only if the contractor completed the works as scheduled (i.e. the *FinishedDate* not later than the *CompletionDate*) and the client fully accepted the works (i.e. the *AcceptanceRate* is 100%) will the contractor get fully paid. If the contractor did not finish the work as scheduled, he/she will either get the payment with a penalty deducted or be refused to be paid altogether. If it is allowed to have partial payments and the value of *AcceptanceRate* is more than the *RejectionThreshold*, the contractor can get a partial payment. In other cases, the client refuses to pay.

```
if(FinishedDate <= CompletionDate)
{
   if(AcceptanceRate == 1.0)
   {
      Payment Execution;
      PaymentStatus = "Fully Paid";
   }
   else if(isPartialPaymentAllowed == true
     && RejectionThreshold < AcceptanceRate < 1.0)
   {
      Payment = Payment*AcceptanceRate;
      Payment Execution;
      PaymentStatus = "Partial Paid";
   }
   else{PaymentStatus = "Payment Refused";}
}
else if(AcceptanceRate == 1.0)
{
   Payment = Payment - Penalty;
   Payment Execution;
   PaymentStatus = "Paid With Penalty";
}
else{PaymentStatus = "Payment Refused";}
```

Figure 9. A smart contract logic for the “Execute Payment” step

4 Conclusion and Outlook

Digitization is widely considered as the solution for the optimization and automation of processes in the construction industry. However, the current practice of a construction project still relies on paper contracts and conventional communications, which are time-consuming and non-transparent. This paper presents a framework for simplification and automation of payments during the construction process by combining the BIM contract container (BCC) with smart contracts. All the project-required data (i.e. the BIM model, the BoQ with QTO, and the billing model) are stored and linked in the BCC. The BCC is stored in an off-chain storage (i.e. CDE), while the sensitive and important information (e.g. the hash values of the documents stored in the CDE, results of acceptance procedure and payments) is safely stored and recorded in the Blockchain. Integration of the billing model and smart contract automates the payments between the client and the contractor during acceptance procedures.

This study has several limitations. First, the subcontractors are not taken into account in this study. Moreover, to define the smart contract rules, there are a lot of logic and legal issues to be settled, which are not handled in this study. Finally, the payment procedure during a construction process is more complex and has more factors to be considered in the future.

Acknowledgment

This work is part of the BIMcontracts research project which is funded within the scope of the “Smarte Datenwirtschaft” technology program run by the German Federal Ministry for Economic Affairs and Energy. The Ruhr-Universität Bochum, Universität Duisburg-Essen, FREUNDLIEB Bauunternehmung GmbH & Co. KG, Kapellmann und Partner Rechtsanwälte mbB and adesso SE (Consortium leader) are involved in the consortium. The authors would like to express their gratitude to the support and funding for the project activities. A further acknowledgement is addressed to the FREUNDLIEB Bauunternehmung GmbH & Co for the provided BIM models and construction data.

References

[1] Agarwal R., Chandrasekaran S. and Sridhar M. Imagining construction’s digital future. *McKinsey Productivity Sciences Center*, 2016.

[2] Caldas C. H., Soibelman L. and Han, J. Automated classification of construction project documents. *Journal of Computing in Civil Engineering*, 16(4): 234–243, 2002.

[3] Buterin V. A next-generation Smart Contract and decentralized application platform. *Ethereum White Paper*, 3:37, 2014.

[4] Eberhardt J. and Tai S. On or off the blockchain? Insights on off-chaining computation and data. In *European Conference on Service-Oriented and Cloud Computing*, pages 3–15, Springer, 2017.

[5] Chen C., Tang L. C. M., and Jin Y. Development of 5D BIM-Based Management System for Pre-Fabricated Construction in China. In *International Conference on Smart Infrastructure and Construction 2019 (ICSIC) Driving data-informed decision-making*, pages 215–224, ICE Publishing, 2019.

[6] buildingSMART. Industry Foundation Classes (IFC). Online: https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/, Accessed: 16/05/2020.

[7] Brook M. Estimating and Tendering for Construction Work. *Butterworth-Heinemann*,

Oxford, 2008.

[8] GAEB. GAEB data exchange XML. Online: https://www.gaeb.de/en/products/gaeb-data-exchange/, Accessed: 17/05/2020.

[9] DIN SPEC 91350: 2016-11: Linked BIM data exchange comprising building information model and specified bill of quantities. Online: https://www.beuth.de/en/technical-rule/ din-spec-91350/263151122, Accessed: 17/05/2020.

[10] DIN EN ISO 21597-1:2018-10 – Draft: Information container for data drop - Exchange specification - Part 1: Container Online: https://www.beuth.de/en/draft-standard/din-en-iso-21597-1/293151537, Accessed: 17/05/2020.

[11] W3C. Resource Description Framework (RDF) Model and Syntax Specification. Online: https://www.w3.org/TR/PR-rdf-syntax/ Overview.html, Accessed: 17/05/2020.

[12] Dorri A., Steger M., Kanhere S. S. and Jurdak R. Blockchain: A distributed solution to automotive security and privacy. *IEEE Communications Magazine*, 55(12): 119–125, 2017.

[13] Bahga A. and Madisetti V. K. Blockchain platform for industrial internet of things. *Journal of Software Engineering and Applications*, 9(10): 533, 2016.

[14] Zheng Z., Xie S., Dai H.N., Chen X. and Wang H. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, 14(4): 352–375, 2018.

[15] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. Online: http://bitcoin.org/bitcoin.pdf, Accessed: 19/05/2020.

[16] Szabo N. Smart Contracts: building blocks for digital markets. *EXTROPY: The Journal of Transhumanist Thought*, (16), 18:2, 1996.

[17] Buterin V. A next-generation Smart Contract and decentralized application platform. *Ethereum White Paper*, 3:37, 2014.

[18] Xu X., Pautasso C., Zhu L., Gramoli V., Ponomarev A., Tran A.B. and Chen S.: The blockchain as a software connector. In *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, pages 182–191, IEEE, 2016.

[19] Li J., Greenwood D. and Kassem M. Blockchain in the built environment and construction industry: A systematic review, conceptual models and practical use cases. *Automation in Construction*, 102: 288–307, 2019.

[20] Kasten J.E. Engineering and Manufacturing on the Blockchain: A Systematic Review. *IEEE Engineering Management Review*, *48*(1): 31–47, 2020.

[21] Shojaei A. Exploring applications of blockchain technology in the construction industry. *Interdependence Between Structural Engineering and Construction Management*, 2019.

[22] Li J., Kassem M., Ciribini A. L. C. and Bolpagni M. A proposed approach integrating DLT, BIM, IOT and smart contracts: Demonstration using a simulated installation task. In *International Conference on Smart Infrastructure and Construction 2019 (ICSIC) Driving data-informed decision-making*, pages 275–282, ICE Publishing, 2019.

[23] Eschenbruch K., Gross D. and König M. Auf dem Weg zum digitalen Bauvertrag - Automatisierung des Zahlungsverkehrs im Bauwesen mittels BIM und Smart Contracts (BIMcontracts). *Bauwirtschaft 1/2020*, 2020.

[24] xBimTeam. XbimWindowsUI. Online: https://github.com/xBimTeam/XbimWindowsUI, Accessed: 25/06/2020.

[25] Hagedorn P. Implementation of a Validation Framework for the Information Container for Data Drop. *Tagungsband 30. Forum Bauinformatik*, pages 147–155, 2018