A Novel Approach and a Language for Facilitating Collaborative Production Processes in Virtual Organizations Based on DLT Networks

Nikola Todorović[1][a], Marko Vještica[1][b], Nenad Todorović[1][c], Vladimir Dimitrieski[1][d] and Ivan Luković[2][e]

[1]*University of Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia*

[2]*University of Belgrade, Faculty of Organizational Sciences, Jove Ilića 154, 11000 Belgrade, Serbia*

Keywords: Virtual Organizations, Process Modeling, Distributed Ledger Technology, Blockchain, Smart Contracts, Internet of Things.

Abstract: Due to strong competition and rapidly shifting market conditions, it is becoming harder for Small and Medium-sized Enterprises (SMEs) to achieve business success. To deal with rising challenges, SMEs form Virtual Organizations (VOs) and seize business opportunities jointly. In this paper, we present an outline of a novel methodological approach that promotes trustworthy collaborative production execution within a non-hierarchical VO. Furthermore, we propose using Distributed Ledger Technology (DLT) platforms and smart contracts to facilitate VO integration. The approach is based on the MultiProLan Domain-Specific Modeling Language (DSML) extended with concepts required to allow process designers to (i) model collaborative production processes while preserving the confidentiality of private enterprise data and (ii) configure what data should be shared between participants during the collaborative production execution. Designed process models are used to automatically generate smart contracts by following the Model-Driven (MD) principles. Finally, generated smart contracts are stored in a DLT network and used to distribute production data between VO participants and monitor production execution in near real-time. The application of our methodological approach is demonstrated by showcasing the use of the Collaborative Extension of MultiProLan (CE-MultiProLan) modeling language and its concepts for modeling collaborative production processes.

1 INTRODUCTION

A Virtual Organization (VO) represents a temporary alliance for integrating competencies and resources from several independent collaborative companies to satisfy customer's requirements, or seize business opportunities by jointly developing complex products (Priego-Roche et al., 2009). VOs are mainly formed by SMEs because their unique capabilities are no longer sufficient for them to individually compete with large companies and countries with lower labor costs (Shamsuzzoha et al., 2013). In this context, Non-Hierarchical Networks (NHNs) represent VOs formed by companies of similar product portfolios and sizes where SMEs enjoy equal rights and controlling power (Shamsuzzoha et al., 2016).

There have been various efforts to assist SMEs in forming and operating NHNs for the collaborative development and delivery of customized products. In (Carneiro et al., 2010), authors present results of a business requirements analysis they conducted in cooperation with industry partners. The goal of the analysis was to determine what major requirements should be addressed to promote collaboration in NHNs. Among the most important requirements, they listed: (i) selection of partners for a specific business opportunity, (ii) standardization and improvement of communication within a VO, and (iii) updating production statuses.

The existing solutions address these requirements by integrating participants' IT systems to enable sharing data about events of interest during production planning and execution, gathered by relying on the Internet of Things (IoT). Here, IoT refers to the networked interconnection of devices that can be used in production, e.g., Radio Frequency IDentification

[a] https://orcid.org/0000-0001-8850-8439
[b] https://orcid.org/0000-0003-2368-5818
[c] https://orcid.org/0000-0002-9809-5719
[d] https://orcid.org/0000-0003-3234-6543
[e] https://orcid.org/0000-0003-1319-488X

Todorović, N., Vještica, M., Todorović, N., Dimitrieski, V. and Luković, I.
A Novel Approach and a Language for Facilitating Collaborative Production Processes in Virtual Organizations Based on DLT Networks.
DOI: 10.5220/0010720900003062
In *Proceedings of the 2nd International Conference on Innovative Intelligent Industrial Production and Logistics (IN4PL 2021)*, pages 197-208
ISBN: 978-989-758-535-7

(RFID) tags, sensors, actuators, and similar. However, on the one hand side, these solutions do not promote transparent cooperation, i.e., they do not put an emphasis on providing a deeper insight into how a VO participant executes a specific production operation and if the production is executed in accordance with the contracted specification. Consequentially, within these solutions, records of events that occur during production are still primarily stored and maintained within isolated IT systems of participants that execute a specific production operation. A solution that facilitates VOs must promote transparent cooperation between participants of a VO because participants may be subjected to a liability action for faults in their performance as part of a VO. A lack of transparency could make it almost impossible to determine who performed a particular action, resulting in uncertainty regarding legal liability. Correct attribution of liability should be facilitated by providing precise documentary evidence concerning the different manufacturing steps and system statuses. Raising the level of integration and sharing additional data between participants would provide much greater insight into executed production processes and significantly increase the transparency within a VO.

On the other hand side, the issue of exposing confidential enterprise data to collaborating parties should also be addressed when facilitating VOs. If not adequately managed, malicious participants could misuse exchanged sensitive data. Therefore, exchanged data must be protected by imposing strict authorization rules that regulate whom and under what circumstances may obtain it. Participants should also be provided with a mechanism that enables configuration of what production data should be shared with other participants of a VO while preserving the confidentiality of private enterprise data.

By emphasizing data privacy and transparency as key attributes of a solution that facilitates VOs, we propose an environment that secures trustworthy conditions for cooperation, i.e., participants could rely on the system to ensure a certain degree of trust is achieved and maintained within a VO. Our research aims at assisting the VO cooperation within NHNs by offering a novel methodological approach for supporting the execution of collaborative production processes. The approach allows VO participants to (i) model collaborative production processes while preserving the confidentiality of private enterprise data and (ii) configure what data should be shared between participants during the production execution. Although our approach primarily aims to support production processes, the concepts it introduces can be applied to other VO domains as well.

On the implementation level, the approach will be facilitated with the use of a DLT platform and blockchain technology with smart contracts for VO integration. DLT is a type of a distributed database, while blockchain represents a distributed data structure that implements DLT and comprises cryptographically linked blocks that contain immutable records of network transactions (Hileman and Rauchs, 2017). Using a DLT platform would provide VO participants with a mechanism for distributing production records that assures a shared control over the access to and evolution of data. Having a shared control over data evolution would increase transparency within a VO, as all stakeholders would be aware of changes made to the shared data. Data stored on a DLT platform are generated and validated using smart contracts – computer programs whose execution is guaranteed by system rules and for which the outcome of execution is verifiable and auditable by all network participants. Smart contracts have the potential to improve coordination and verification within a VO by automatically verifying that the production process actions are executed according to the contracted production specification. Data privacy could be addressed by relying on a private, permissioned, consortium-based DLT platform for storing production records. Such platforms impose strict restrictions regarding ’read’ and ’write’ permissions on the ledger (Wust and Gervais, 2018), which is critical for protecting private enterprise data.

One downside of using DLT platforms is that they usually provide low-level, general-purpose programming languages for implementing smart contracts. This is not always suitable for facilitating VOs because a manual specification of smart contracts would reduce the capability of VOs to synchronize and adapt their production in a timely manner. Moreover, it would mean that process designers, i.e., process and quality engineers responsible for the production specification, need to be proficient in these languages. This downside could be mitigated by (i) providing process designers with a modeling language that is based on concepts and notations they are familiar with and already use in their domains, and then (ii) relying on automatic generation of smart contracts (France and Rumpe, 2007). Thus, our approach is centered around a DSML that provides collaborating parties with concepts required for describing Cross-Organizational Business Processes (CBPs) (Berre et al., 2007) in a sufficiently detailed and understandable way to enable the execution of specified processes. It is also facilitated by a software solution in which the MD principles and DSMLs are used to (i) specify contracted CBPs and (ii) automatically generate smart contracts used for production monitoring. The described approach enables an analysis of records of events that occur during the production and allows parties to derive conclusions and determine if there are any discrepancies between negotiated and executed process steps. The approach aims to create trustworthy conditions for collaboration between VO participants and allow them to be more competitive in the market.

The presented work is structured as follows. Apart from Introduction and Conclusion, the paper has three sections. Section II examines the background and related work for the approach. It also discusses the notational aspects and execution significance of a DSML used to create collaborative production process models. Next, Section III provides an outline of the approach. Finally, in Section IV, we present the capabilities of the CE-MultiProLan DSML used to model collaborative production processes.

2 BACKGROUND AND RELATED WORK

There have been various efforts to assist SMEs in forming and operating NHNs for the collaborative development and delivery of customized products (Priego-Roche et al., 2016). The Net-Challenge project (Carneiro et al., 2010; Carneiro et al., 2014) proposed a methodology for supporting the creation and operation of VOs. The methodology is structured in five main phases: (i) Build and develop a Business Community, i.e., a phase of creating an environment that comprises a significant number of organizations interested in joining VOs, (ii) Qualify, a phase where data about potential partners is collected and stored to be taken into consideration when a VO is being formed, (iii) Form, a phase in which a VO is created, and product concept and cost estimations are defined, (iv) Operate, a phase in which detailed production plan is defined, after which the production is executed and monitored, and (v) Dissolve, a phase in which performance evaluation is conducted, after which a VO is dissolved. The methodology also introduces three types of roles included in VO management: (i) the Broker role, representing an organization responsible for coordinating a VO; (ii) the Core partner role, representing organizations that collaborate actively in the formation of the VO and the definition of the product concept; and (iii) the Additional partner role, representing organizations that are invited to provide quotations for specific operations or materials and do not participate in the product design phase. Based on the methodology, authors (Shamsuzzoha et al., 2016) created an innovative solution for supporting collaboration between SMEs and real-time information sharing during production execution. IoT devices in production plants are used to collect data about possible production process disruptions during the execution of the Operate phase. This data is then processed and stored in a centralized database. Stakeholders can access the collected data through a web app to identify levels of alerts and statuses of executed process activities (e.g., machine breakdown, shortage of raw materials).

Data about process disruptions collected in the presented solution still represents only a fraction of data generated during the Operate phase of the VO lifecycle. As a result, records of events that occur during production are primarily stored and maintained within isolated IT systems of participants that executed a specific production activity. Sharing additional data between participants would provide much greater insight into executed production processes and significantly increase the transparency within a VO. In addition, participants should be provided with a mechanism that enables modeling of production processes and configuring what production data should be shared with other participants of a VO while preserving the confidentiality of private enterprise data.

The feasibility of our approach is impacted heavily by the interoperability concerns of participant's production systems. To enable interoperable event-sharing configuration and distribution of production data, we utilized concepts identified in the ATHENA Interoperability Framework (AIF) (Berre et al., 2007), a methodological framework that enables collaborative modeling and execution of CBPs. AIF facilitates the enactment of CBPs by merging research areas that support the development of interoperable enterprise solutions, most notably: (i) enterprise modeling used to define interoperability requirements and support solution implementation, and (ii) frameworks for implementation of interoperable platforms.

Production process modeling is an important research topic (Xu, 2011), but it is still not adequately covered with the existing studies (Petrasch and Hentschke, 2016). Different notational aspects of production process models, regarding their expressiveness and configuration of what data should be shared in a VO, are examined in Section 2.1. Our research also promotes trust-building between partners by relying on a DLT platform and smart contracts for data distribution. Different aspects of DLT platforms that have significance for the enactment and integration of CBPs are addressed in Section 2.2.

2.1 Event-sharing Configuration based on Process Modeling

Several research challenges should be taken into consideration when modeling processes that are collaboratively executed in a VO. On the one hand side, modeling of CBPs implies that a modeling language supports designers in describing production process specifications in a sufficiently detailed and understandable way to enable the execution of specified processes. On the other hand side, these specifications should be displayed to related parties through different process interfaces that facilitate understanding of collaboration in a VO while preserving the confidentiality of private, internal enterprise information. Thus, one of the most significant challenges for designing such language is to devise concepts that connect private production processes with openly exposed process interfaces and combine different representations of intra-organizational processes at CBP level (Lippe et al., 2005). Additionally, the modeling language should allow users to model details needed on the execution level, e.g., showing invoked smart contracts and executed transactions, while separating CBP modeling from a specific deployment architecture.

The collaboration is based on a distributed process model where parties manage their part of the overall production process. Three different process types have been investigated and customized for use in collaborative production to allow disclosure of private process data to VO participants: (i) private production processes that represent internal processes executed by an organization; (ii) interface processes used to coordinate internal actions with activities of external partners while concealing private data; and (iii) CBPs used to describe how participants collaborate to produce the end product (Lippe et al., 2005). In Section 3.1, we describe how models of these process types are utilized to provide a mechanism that enables configuring what production data should be shared between VO participants and generate smart contracts that enable production monitoring based on the shared data.

Modeling of private production processes has been an important topic of our previous research (Vještica et al., 2021b). The research resulted in an MD approach, and a DSML tool named MultiProLan, which can be used to model production processes suitable for automatic execution in smart factories (Vještica et al., 2021a). By relying on the possibilities of the MultiProLan, process designers can be focused only on process steps that must be executed and need not worry about production logistics and resources that will execute the process steps.

The modeling of private production processes in MultiProLan is performed at two different levels of detail to make modeling easier for process designers. First, the approach provides Master-Level (MasL) models at a lower level of details, used by process designers to create process models independent of the production facility in which the production will be executed. MasL models contain operation and inspection activities with their corresponding inputs and outputs, capabilities required to execute them, and similar. Second, at a higher level of details, the approach provides Detail-Level (DetL) models, created by enriching MasL models with details about resources available in the specific production facility in which the production will take place. DetL models are either created manually, by a process designer, or are automatically generated by an Orchestrator, a system that delegates instructions to different smart resources in a smart factory (Pisarić et al.,). The approach we introduce in this paper extends the capabilities of the MultiProLan tool by providing concepts required for collaborative production planning and execution. We named this extension as Collaborative Extension of MultiProLan.

MultiProLan was selected as a basis for modeling collaborative production processes in our approach because it was built to support process designers in modeling execution-ready production processes in multiple levels of detail. In addition, it was devised in a way that makes it relatively easy to extend it to support additional concepts required for supporting CBPs. Also, the language has already been tested in an assembly use case that included collaboration between human workers and robots in a production setting. Our approach could be based on the use of a general-purpose process modeling language, like BPMN, for supporting the collaborative execution of production processes. However, BPMN lacks the semantics of production processes as it was mainly tailored to model business processes. Hence, BPMN is not adequate for modeling production processes ready for automatic code generation and execution of the code, especially when modeling products, capability constraints, parameters, and the material flow (Vještica et al., 2021b). Still, we could not find a fitting modeling language that adequately facilitates the modeling of execution-ready production processes.

2.2 DLT as an Execution Platform

An execution platform that facilitates horizontal integration should provide mechanisms that guarantee a secure and transparent distribution of records to related parties to achieve a common understanding of these events. The architecture recommended by AIF can be expanded to encourage the use of a DLT platform and smart contracts for information sharing and supporting trust-building between parties. To identify technical requirements that should be taken into consideration when developing an execution platform, we relied on a list of Quality Attributes highlighted by the network for Interoperability Development of Enterprise Applications and Software (IDEAS) (Chen and Doumeingts, 2003). We selected the most important attributes that have to be considered when developing a platform for sharing data during the enactment of CBPs: (i) security, i.e., mechanisms that platform offers to protect private enterprise data; and (ii) scalability and performance, i.e., the possibility of a platform to process a large amount of data generated by smart resources on the shop floor.

There have been several attempts at utilizing DLT platforms and smart contracts for supporting VOs. For example, in (O'Leary, 2019), the author suggests that, because of their unique nature, VOs provide an important potential setting for the use of blockchain-like designs. However, the presented paper primarily deals with integrating participants' accounting systems rather than integrating their production processes and production systems. Multiple existing solutions consider using DLT platforms for monitoring the enactment of CBPs (Weber et al., 2016) (Klinger and Bodendorf, 2020). These solutions are based on BPMN and rely on an MD approach to generate smart contracts that facilitate the integration of collaborative processes supported by a DLT network. In these solutions, authors present tools that take business process models as an input and generate smart contracts that are then deployed on the Ethereum public DLT network (Chowdhury et al., 2019). Described methods have limitations regarding their usability in the collaborative production scenario. The use of a public DLT network like Ethereum may not fit the high data security requirements of the collaborative production domain. Instead, enterprise solutions that rely on a private, consortium federated DLT network should be used to protect highly sensitive corporate data. In addition, these solutions lack mechanisms for preserving sensitive corporate data when modeling and executing collaborative processes.

Scalability and performance may also become a concern with the use of a public DLT network, where each transaction needs to be processed by every single node in the network. For instance, Ethereum supports up to 15 transactions per second (tps), which creates a severe bottleneck when supporting the execution of production processes in collaborative production, where machines involved in manufacturing generate transactions at a much higher pace. To the best of our knowledge, none of the existing solutions consider high security, performance, and scalability requirements in a unified way.

2.3 Summary

During the research, we found several solutions that rely on an MD approach for generating smart contracts that support the execution of collaborative processes within a VO. However, these solutions lack mechanisms for concealing sensitive corporate data when modeling and executing collaborative processes. In addition, these solutions are based on BPMN, which is not adequate for modeling production processes ready for automatic instruction generation. Since we could not find a fitting solution, we decided to introduce a novel methodological approach presented in this paper. The approach enables VO participants to model and execute collaborative production processes while preserving confidential information by carefully selecting what private data will be disclosed to collaborating parties. Furthermore, the proposed approach unifies all collaborative production process aspects, as presented in the rest of this paper, and thus enables the specification of production process models used for automatic smart contract generation and production process execution monitoring.

3 AN OUTLINE OF THE APPROACH

To promote trustworthy and transparent collaboration between participants of a VO, we introduce a methodological approach in which CE-MultiProLan is used to model collaborative production processes and configure what data should be shared between participants during the execution of CBPs. A software solution that utilizes MD principles is used to generate smart contracts based on these models automatically. Finally, generated smart contracts are stored in a DLT network and used for production execution monitoring and trustworthy distribution of production data between VO participants. This section presents the outline of the approach. The complete methodological approach will be described in detail in our following paper.

The approach is based on the Net-Challenge methodology (Carneiro et al., 2010) but introduces improvements regarding how CBPs are modeled and executed in a VO, allowing for a higher level of integration between participants while preserving private enterprise data. The expected advantages of applying the presented approach for supporting the collaborative production execution within a VO are: (i) a more real-time insight into production status; (ii) improved trust between participants as transparency within the network is increased, and contract validations are automated and tamper-proof; and (iii) faster time to market due to the automatic generation of smart contracts. These advantages jointly create trustworthy conditions for collaboration between SMEs involved in a VO and allow them to be more competitive in the market. The outline of the approach is depicted in Fig. 1. The modeling of collaborative processes, shown on the left-hand side of the figure, is described in 3.1, while smart contract generation and process monitoring, shown on the right-hand side of the figure, is described in 3.2.

The approach we outline in this paper was initially proposed in our previous paper (Todorovic et al., 2020), but has since been refined to fit the VO domain better. To evaluate the feasibility of our approach, we have developed a simple implementation of a prototype solution. Within this solution, we created the CE-MultiProLan modeling tool. We also created a code generator that generates smart contracts based on process models defined with CE-MultiProLan. The generated smart contracts were deployed to an established DLT network, from where users can access shared records and monitor the production execution. The use of CE-MultiProLan is demonstrated in Section 4.

3.1 Modeling Collaborative Processes

During the Form phase of a VO lifecycle of Net-Challenge methodology, participants are required to develop a product concept collaboratively and define a corresponding Engineering Bill-Of-Materials (eBOM). eBOM represents a structure of a product at the product design phase. It contains relationships between product's materials, parts and subassemblies. In this lifecycle phase, a corresponding Bill-Of-Operations (BOO) also needs to be defined. The operations specified in BOO are then allocated to VO participants. Defined eBOM and BOO documents are used as input to our approach.

In the Operate phase, as the first step of our approach, depicted as *1* in Fig. 1, eBOM and BOO are used by collaborating parties to specify a CBP Model (CBPM). A CBPM is created to coordinate production between participants and contains a sequence of production activities (e.g., fabricating a part, assembling a product) allocated to participants responsible for executing them. Collaborating parties can also use CBPM to specify roles of the included participants, production milestones, i.e., critical points in production used to determine the state of an activity (e.g., how much time each activity requires), and a due date for an activity.

Next, in step *2*, VO participants need to design their allocated individual production operations from BOO. Participant's process designers are responsible for designing a MasL Production Process Model (MasL-PPM), which represents a private production process specification containing: (i) production steps, (ii) capabilities required to execute each of the steps, (iii) input and output products, i.e., transformed resources like raw materials and components from eBOM, (iv) constraints and (v) capability parameters. After MasL-PPM is created, it is utilized for creating a DetL Production Process Model (DetL-PPM) in step *3*, and an Interface Production Process Model (I-PPM) step *4*.

DetL-PPM is an execution-ready process model generated by enriching MasL-PPM with details about IoT devices, i.e., Smart Resources available in a specific production facility that satisfy the requirements defined in MasL-PPM. DetL-PPM is either created manually by the process designer or automatically generated by the Orchestrator and is used to generate commands for orchestrating dedicated resources and executing the specified production process. This paper does not cover DetL-PPM in detail as process execution is out of the scope of our approach, but its details can be found in (Vještica et al., 2021b).

I-PPM represents a public interface created over a private MasL-PPM that provides an insight into how the responsible VO participant executes a specific value-adding CBPM operation. It is created as a viewpoint over MasL-PPM and is defined by process designers manually, with some details present in MasL-PPM anonymized or concealed to preserve enterprise data privacy. I-PPM is also used to configure what data should be shared between collaborating parties by specifying (i) what production steps from MasL-PPM should be traced during production execution and (ii) what data should be persisted alongside those step traces. All operations from CBPM for which the execution should be monitored must refer to a corresponding I-PPM. Data to be exchanged during CBPM operation execution is agreed upon with a responsible VO participant.

3.2 Smart Contract Generation and Production Monitoring

Smart contracts that monitor the enactment of CBPs and track the state of each activity in the collaborative

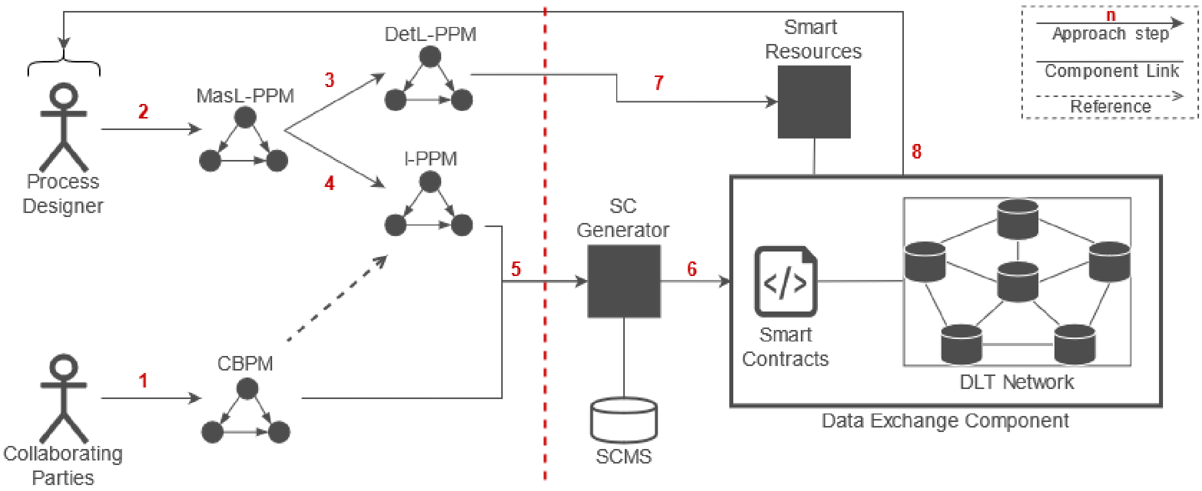

Figure 1: The outline of the approach.

production process are generated in step *5* of the approach using the Smart Contract Generator (SC Generator) component. For smart contract generation to be possible, SC Generator must implement an algorithm that enables translation I-PPMs and CBPMs to executable smart contract code. The algorithm that we used represents a modified version of the algorithm presented in (Weber et al., 2016). In addition, we rely on a Smart Contract Meta-Store (SCMS), a component built and maintained by a business community. SCMS is used to store (i) smart contract templates, (ii) id values for all generated smart contracts, and (iii) data about references between smart contracts. SC Generator uses I-PPMs and templates from SCMS to generate smart contracts to monitor a production process executed by a single participant. Based on CBPMs and templates from SCMS, SC Generator automatically generates smart contracts to monitor the enactment of CBPs.

Once smart contracts are generated, in step *6* they are stored on a DLT Network to which all parties involved in a business network have access. The DLT Network and smart contracts represent a basis for implementing the Data Exchange Component (DEC). Smart Resources can send records about production execution events from the shop floor to the DEC, automatically triggering actions specified as a part of the stored smart contracts. The role of smart contracts, generated based on CBPMs and I-PPMs, is to monitor these records and validate that the production execution is conducted according to the contracted specifications. As a final step of the approach, step *8*, collaborating parties can oversee the state of production and contract fulfillment by looking at the immutable store using a set of data visualization tools.

When selecting a production-ready DLT platform with the appropriate characteristics, we have taken into consideration the quality attributes defined in Section 2.2. In (Chowdhury et al., 2019), authors presented a comparative analysis of the existing production-ready DLT platforms. Based on their findings, we selected Hyperledger Fabric as the most appropriate platform for facilitating the collaborative production domain. The requirements related to the security quality attribute were addressed by selecting a private, permissioned, consortium-based DLT platform administered by a set of identified parties involved in a business network operating under a governance model that enforces a certain degree of trust (Wust and Gervais, 2018). Private DLT networks impose restrictions on 'read' access to the ledger, i.e., who can access the network and see transactions. Furthermore, permissioned DLT networks allow only a selected set of parties to submit new transactions to the distributed ledger. In addition, Hyperledger Fabric introduces advanced mechanisms for

Scalability and performance concerns have also been considered. Machines used in production can generate numerous records that need to be processed by a selected DLT network with low latency to enable a sufficiently reliable distribution of data between participants. By relying on the identities of participants, Hyperledger Fabric can use a more traditional Crash Fault-Tolerant (CFT) consensus protocol that is suitable for scaling the transaction throughput in the network. The use of CFT enables Hyperledger Fabric to support throughput of more than 3,500 tps, and it has also been shown to allow up to 20,000 tps in certain setting(Gorenflo et al., 2019). Such throughput can cover many different collaborative production use-cases.

4 AN APPLICATION OF CE-MULTIPROLAN ON A COLLABORATION USE CASE

In this section, we present a use case that demonstrates the application of CE-MultiProLan, our DSML developed to model collaborative production processes and configure what data should be shared between VO participants during the production execution. We used the Ecore meta-meta-model, which is a part of the Eclipse Modeling Framework (EMF) (Steinberg et al., 2008), to create the abstract syntax of CE-MultiProLan. Also, we used the Eclipse Sirius framework (sir,) to create the graphical concrete syntax of CE-MultiProLan and to enable the simple implementation of a prototype solution.

The use case presented in this section demonstrates the use of CE-MultiProLan for describing the production process for a decorative wooden wine box with an engraved acrylic front. It was devised to cover core concepts for the domain of collaborative production process modeling, introduced below. Models covered in this use case and the smart contracts generated based on these models are available [1].

CBPM, displayed in Fig. 2, is used to describe how VO participants collaborate to produce the end product jointly. The graphical syntax of CE-MultiProLan that supports the creation of CBPMs introduces a pool, depicted as a rectangle with a light-blue filling for visual grouping of elements related to a single VO. Within a pool, each participant of a VO is assigned a swimlane, depicted as a rectangle with a white filling, where each swimlane contains process steps executed by one participant of a process. For example, the presented CBPM displays a single VO named ACME Wine Inc, in which three different organizations collaborate to produce a wooden wine box: (i) the Winery, with the Broker (B) role in the VO, (ii) the Woodworking Company that has a role of a Core Partner (CP), and (iii) the Acrylic Engraving Company, with the Additional Partner (AP) role within the VO. The introduction of pools and swimlanes was motivated by similar concepts that exist in BPMN. Here, they are used for creating clear and well-structured models that specify roles and responsibilities for each VO participant.

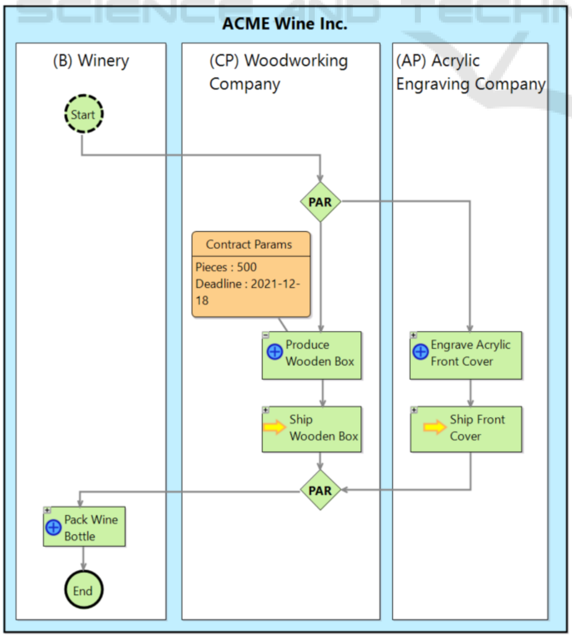

Figure 2: CBPM for a decorative wine box production.

[1]https://github.com/TodorovicNikola/CE-MPL

The production is initiated when the Winery receives an order (*Start*). After the production initiation, the CP and the AP perform their allocated production activities in parallel as their activities are modeled between parallelism (PAR) gates. The CP produces a batch of wooden boxes (*Produce Wooden Box*). The production must meet the specified contractual clauses, stating that a total of 500 pieces should be produced until the specified deadline. After the production of the wooden boxes is completed, they are shipped to the Winery (*Ship Wooden Box*). At the same time, the AP engraves a batch of acrylic front covers with a specified pattern (*Engrave Acrylic Front Cover*) and ships processed front covers to the Winery (*Ship Front Cover*). Finally, the Winery packs wine bottles in the produced boxes and inserts the acrylic front covers (*Pack Wine Bottle*). Transportation steps are marked with a yellow arrow, while the value-adding operation steps are marked with a blue circle with a plus sign in it. The plus sign indicates that a CBPM operation represents a high-level process step composed of low-level steps specified in the associated I-PPM. I-PPM provides an insight into how the responsible VO participant executes the CBPM operation to complete the collaborative operation. I-PPM also defines what data will be shared between VO members for each CBP operation during production execution.

Fig. 3 illustrates MasL-PPM for the Produce Wooden Box operation, performed by the CP (left) and corresponding I-PPM (right). Process designers use MasL-PPM to define the operation and inspection steps that need to be executed during production. The presented MasL-PPM model is composed of six parts: (i) the start process step; (ii) parallel process steps of assembling left-bottom (L-B) and right-upper (R-U) sides of a box, after which these two assembled sides

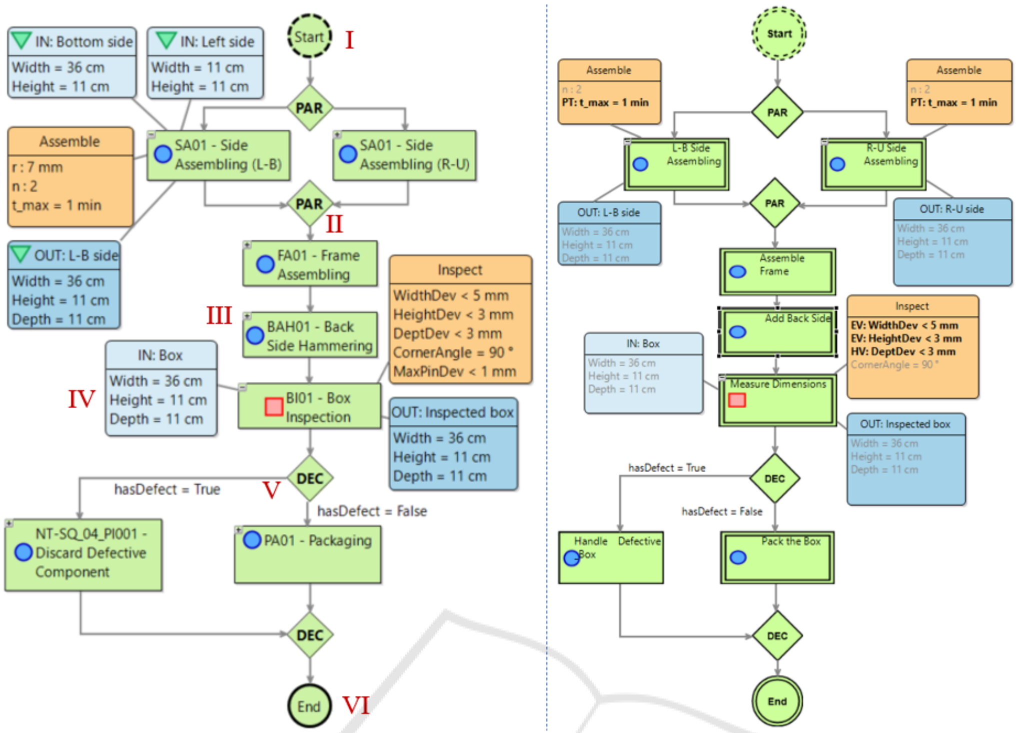

Figure 3: MasL-PPM (left) and I-PPM (right) for the Produce Wooden Box CBPM step.

are assembled into a frame; (iii) hammering a back-side panel into the frame; (iv) a manual inspection of the box that verifies that it conforms to the specified constraints; (v) decision whether the box needs to be stored or discarded, depending on results of the inspection process step; and (vi) the end process step. As depicted, process step names can contain work instruction numbers, e.g., *Discarding Defective Component* step in the model, or id values assigned to process steps by an Enterprise Resource Planning (ERP) tool, present in all other operation or inspection steps in the model.

Input products, output products, and capabilities of a process can be hidden from a diagram using a +/- button at the top left corner of a process step so that a process diagram could be more or less complex depending on the designer's needs. Due to the length limitations of this paper, Fig 3 depicts these detailed specifications for just two process steps, necessary for explaining different concepts, while for the rest, they are specified but not displayed.

The process step of assembling the L-B side represents an operation, depicted with a circle icon at the left side of the process step name. It has two input products, left and bottom sides of the frame, both collected from a storage. The inverted triangle icon at the left side of a product name indicates that an input product should be collected from a storage or that an output product should be placed in a storage. Two input products are associated with two dimension constraints, width and height, that will be considered by the Orchestrator when it assigns a smart resource that is able to pick the plank of these dimensions. The same process step has the *Assemble* capability with parameters representing two wooden pins with the space between them of 7mm, and a maximum time in which this step should be executed. The output product of this process step is the assembled L-B side, which will not be stored but will be used by the following process step. Assembling the R-U side is an equivalent process step to assembling the L-B side process step. Both process steps can be executed in parallel, as they are modeled between two PAR gates. The following process step, *Frame Assembling*, requires assembling the frame and has two input products - L-B and R-U sides from the previous two process steps. After the *Frame Assembling* process step is finished, the back side is hammered into the frame. Next, the *Box Inspection* process step is performed to inspect if the box has any defects. Here, a decision to store or discard the box should be made depending on whether the box passes all checks or not, e.g., if dimension deviations meet the box specified values. Storing and discarding the box steps are modeled between two decision gates (DEC). The process is finished after it reaches the *End* process step.

I-PPM, which corresponds to the described MasL-PPM, is shown on the right-hand side of Fig 3. It is created by adapting details available in MasL-PPM for displaying them to collaborating parties without disclosing confidential production details. Several mechanisms have been introduced to protect the confidential information of participants. First, for each of the elements in the model, e.g., process steps, input and output products, capabilities, and constraints, the process designer can choose whether or not to expose them to the collaborating parties. Thus, elements from the MasL-PPM that should not be exposed are omitted from I-PPM. For example, input products of the L-B and R-U side assembling process steps are omitted from the I-PPM, as they are regarded as confidential. Similarly, confidential capabilities and constraints can also be hidden (e.g., space between wooden pins for the L-B and R-U side assembling process steps, or max pin distance deviation in the *Measure Dimensions* process step). Second, since process step names can sometimes disclose private information, e.g., work instruction numbers and ERP id values, aliases were introduced. By introducing aliases, process step labels in I-PPM can contain customized values different from those displayed in MasL-PPM, thus concealing private information.

I-PPM is also used to configure what data should be shared with collaborating parties during production execution. Process steps that should be traced during production execution are displayed as double-edged rectangles. In contrast, process steps that should not be traced are shown as rectangles with a single edge (e.g., *Handle Defective Box* process step). The collaborating parties might decide not to trace a specific process step if deemed irrelevant for the collaboration context. When process steps are traced, additional data can be shared with collaborating parties. Id values for exposed input and output products used within process steps are automatically shared with collaborating parties as they are significant for product provenance. Other details about input and output products are displayed in the model only for documentation purposes. Capabilities and constraints also specify additional data that should be shared with collaborating parties alongside process step traces. For example, the *Measure Dimensions* process step has the related *Inspect* capability containing disclosed inspections performed on every produced wooden box. For constraints given in bold font, data will be shared with collaborating parties, while constraints displayed with a gray color are shown in the model only for documentation purposes, and data about those constraints will not be shared. In addition, for each constraint, parties can specify if shared data should be stored as a Plain Text (PT), as a Hashed Value (HV), or an Encrypted Value (EV).

The presented CBPM and I-PPM are suitable for use in the collaboration scenario. CBPM is based on concepts that allow collaborating parties to describe how they collaborate to produce the end product. Furthermore, I-PPM supports concepts that allow collaborating parties to coordinate internal actions with activities of external partners while concealing private data. For this to be possible, we have extended the scope of concepts supported by the MultiProLan DSML with those required for collaborative process modeling. In addition, I-PPM was built as a viewpoint over MasL-PPM, which makes it easier for process designers to generate I-PPM based on the existing MasL-PPM. By relying on a clear separation of collaborative process models, interface process models, and private process models, collaborating parties can jointly plan production while preserving private enterprise data.

5 CONCLUSION

This paper presents an overview of a novel methodological approach that promotes trustworthy and traceable collaborative production execution within a non-hierarchical VO. The approach is based on the capabilities of the presented CE-MultiProLan DSML used by process designers to (i) model collaborative production processes while preserving the confidentiality of private enterprise data and (ii) configure what data should be shared between participants during the execution of CBPs. Process models designed using CE-MultiProLan DSML are then used as an input for a software solution that relies on MD principles to generate smart contracts automatically. Finally, generated smart contracts are stored in a DLT network and used for production execution monitoring and trustworthy distribution of production data between VO participants. The use of CE-MultiProLan is demonstrated in this paper on a use case that describes the production process for a decorative wooden wine box with an engraved acrylic front.

The core part of the MultiProLan language has been tested by process designers on the shop floor within a small-scale industrial production setup. It has also been evaluated through a questionnaire, and the evaluation results have been published (Vještica et al., 2021a). The evaluation conclusion is that MultiProLan fulfills all the following quality characteristics: functional stability, usability, reliability, expressiveness, and productivity. The language extension that supports the collaborative execution of production processes introduced in this paper will be systematically evaluated and tested on a case study common for VOs with a non-hierarchical structure. Furthermore, we plan to improve the possibilities of CE-MultiProLan for modeling collaborative production processes by expanding the set of concepts available on the interface process level. For example, advanced concepts already present on the private process level, like sub-processes and unordered steps, should also be available on the interface level. The support for the newly introduced concepts should also be implemented in the software that generates smart contracts. In addition, we plan to investigate the possibility of utilizing enterprise modeling constructs defined in the newly introduced ISO standard for Enterprise Modelling and Architecture (ISO 19440:2020, 2020). This standard focuses on engineering and the integration of manufacturing and related services in the enterprise. We plan to analyze the possibility of using those constructs for production process modeling. Even though the standard is rather new, there have already been attempts to utilize it in the production modeling context. For example, authors in (Wu et al., 2021) utilize it for extracting object classes for a meta-model used to describe Cyber-Physical Production Systems(CPPS).

Even though our solution introduces a tamper-proof and immutable way for storing production data, the issue of when a malicious partner tries to submit incorrect data about process execution persists. Although the production process automation somewhat mitigates this issue by decreasing the space for manual data input, there is still a real possibility that a malicious collaboration partner could try to submit the wrong data. For this reason, we plan to investigate the possibility of detecting such behavior with a component that would perform smart contract execution log analysis. As a part of this investigation, we published a paper that expands further on the topic of finding discrepancies between contracted and executed production processes (Ivković and Luković, 2021).

The expected value of our approach for parties involved in a VO is increased structural transparency during the enactment of collaboration as contracts are automated and tamper-proof. In addition, the expected scientific implication is a novel methodological approach for the integration of VO participant's information systems that would create conditions for trustworthy and traceable production execution.

ACKNOWLEDGEMENTS

This paper is supported by the Ministry of Education, Science and Technological Development through the project no. 451-03-68/2020-14/200156: "Innovative scientific and artistic research from the FTS domain".

REFERENCES

Eclipse sirius documentation. `https://www.eclipse.org/sirius/doc/`. Accessed: 2021-07-29.

Berre, A. J., Elvesæter, B., Figay, N., Guglielmina, C., Johnsen, S. G., Karlsen, D., Knothe, T., and Lippe, S. (2007). The athena interoperability framework. In *Enterprise Interoperability II*, pages 569–580.

Carneiro, L., Shamsuzzoha, A., Almeida, R., Azevedo, A., Fornasiero, R., and Ferreira, P. S. (2014). Reference model for collaborative manufacturing of customised products: applications in the fashion industry. *Production Planning & Control*, 25(13-14):1135–1155.

Carneiro, L. M., Almeida, R., Azevedo, A. L., Kankaanpaa, T., and Shamsuzzoha, A. H. M. (2010). An Innovative Framework Supporting SME Networks for Complex Product Manufacturing. In *Collaborative Networks for a Sustainable World*, volume 336.

Chen, D. and Doumeingts, G. (2003). European initiatives to develop interoperability of enterprise applications—basic concepts, framework and roadmap. *Annual Reviews in Control*, 27(2):153–162.

Chowdhury, M. J. M., Ferdous, M. S., Biswas, K., Chowdhury, N., Kayes, A. S. M., Alazab, M., and Watters, P. (2019). A Comparative Analysis of Distributed Ledger Technology Platforms. *IEEE Access*, 7:167930–167943. Conference Name: IEEE Access.

France, R. and Rumpe, B. (2007). Model-driven Development of Complex Software: A Research Roadmap. In *Future of Software Engineering (FOSE '07)*, pages 37–54.

Gorenflo, C., Lee, S., Golab, L., and Keshav, S. (2019). FastFabric: Scaling Hyperledger Fabric to 20,000 Transactions per Second. *arXiv:1901.00910 [cs]*. arXiv: 1901.00910.

Hileman, G. and Rauchs, M. (2017). 2017 Global Blockchain Benchmarking Study. *SSRN Electronic Journal*.

ISO 19440:2020 (2020). Enterprise modelling and architecture — constructs for enterprise modelling. Standard, International Organization for Standardization.

Ivković, V. and Luković, I. (2021). An approach to validation of business-oriented smart contracts based on process mining. *Bellatreche L. et al. (eds) New Trends in Database and Information Systems. ADBIS 2021. Communications in Computer and Information Science*, 1450:303–309.

Klinger, P. and Bodendorf, F. (2020). Blockchain-based Cross-Organizational Execution Framework for Dynamic Integration of Process Collaborations. In *WI2020 Zentrale Tracks*, pages 893–908.

Lippe, S., Greiner, U., and Barros, A. (2005). A Survey on State of the Art to Facilitate Modelling of Cross-Organisational Business Processes. pages 7–22.

O'Leary, D. E. (2019). Some issues in blockchain for accounting and the supply chain, with an application of distributed databases to virtual organizations. *Intelligent Systems in Accounting, Finance and Management*, 26(3):137–149.

Petrasch, R. and Hentschke, R. (2016). Process modeling for industry 4.0 applications: Towards an industry 4.0 process modeling language and method. In *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–5.

Pisarić, M., Dimitrieski, V., Vještica, M., Krajoski, G., and Kapetina, M. Towards a flexible smart factory with a dynamic resource orchestration. *Applied Sciences, Special Issue Information and Communications Technology for Industry 4.0, MDPI*, 11:7956.

Priego-Roche, L. M., Front, A., and Rieu, D. (2016). A framework for virtual organization requirements. *Requirements Engineering*, 21(4):439–460.

Priego-Roche, L.-M., Rieu, D., and Front, A. (2009). A 360° Vision for Virtual Organizations Characterization and Modelling: Two Intentional Level Aspects. In *Software Services for e-Business and e-Society*, volume 305.

Shamsuzzoha, A., Kankaanpaa, T., Carneiro, L. M., Almeida, R., Chiodi, A., and Fornasiero, R. (2013). Dynamic and collaborative business networks in the fashion industry. *International Journal of Computer Integrated Manufacturing*, 26(1-2):125–139.

Shamsuzzoha, A., Toscano, C., Carneiro, L. M., Kumar, V., and Helo, P. (2016). ICT-based solution approach for collaborative delivery of customised products. *Production Planning & Control*, 27(4):280–298.

Steinberg, D., Budinsky, F., Paternostro, M., and Merks, E. (2008). Eclipse Modeling Framework Second Edition.

Todorovic, N., Vjestica, M., Dimitrieski, V., Zaric, M., Todorovic, N., and Lukovic, I. (2020). Towards trustworthy horizontal integration in industry 4.0 based on dlt networks. *Proceedings, Federated Conference on Computer Science and Information Systems (FedCSIS), 4th International Conference on Lean and Agile Software Development (LASD 2020)*, 22:63–69.

Vještica, M., Dimitrieski, V., Pisarić, M., Kordić, S., Ristić, S., and Luković, I. (2021a). Multi-level production process modeling language. *Journal of Computer Languages, Elsevier*, 66:101053.

Vještica, M., Dimitrieski, V., Pisarić, M., Kordić, S., Ristić, S., and Luković, I. (2021b). Towards a Formal Specification of Production Processes Suitable for Automatic Execution. *Open Computer Science*, 11(1):161–179.

Weber, I., Xu, X., Riveret, R., Governatori, G., Ponomarev, A., and Mendling, J. (2016). Untrusted Business Process Monitoring and Execution Using Blockchain. In *Business Process Management*, volume 9850, pages 329–347.

Wu, X., Goepp, V., Siadat, A., and Vernadat, F. (2021). A method for supporting the transformation of an existing production system with its integrated Enterprise Information Systems (EISs) into a Cyber Physical Production System (CPPS). *Computers in Industry*, 131:103483.

Wust, K. and Gervais, A. (2018). Do you Need a Blockchain? In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 45–54.

Xu, L. D. (2011). Enterprise Systems: State-of-the-Art and Future Trends. *IEEE Transactions on Industrial Informatics*, 7(4):630–640.