

Compliance Checking: from Object-Centric Process Mining to Blockchain

Alessio Galassi¹

¹University of Camerino, School of Science and Technology, Italy

Abstract

Ensuring compliance in business process execution is crucial for regulatory adherence and operational efficiency. Compliance checking techniques verify whether executions conform to predefined rules, but existing approaches struggle with complex, multi-entity interactions and the lack of trust among participants. This research explores a twofold advancement on compliance checking topic. On the one hand is the enabling of object-centric compliance checking. On the other hand is the intertwining of novel compliance checking techniques with blockchain technology. Respectively, object-centric process mining provides a structured and accurate process representation. Blockchain, instead, ensures trust and transparency through its immutable transaction records. The connection point lies in the fact that the structured process representation of object-centric process mining aligns well with blockchain's transaction records, enabling seamless integration. Preliminary results include a compliance rule language that supports the object-centric perspective and a domain-specific language for blockchain-aware rule specifications. Future work aims to integrate these techniques into a unified compliance checking framework while addressing additional challenges in decentralized environments, leveraging smart contracts and real-time verification.

Keywords

Process mining, Compliance checking, Object-centric event data, Blockchain technology

1. Research Problem and Motivation

Ensuring compliance in business process execution is crucial for regulatory adherence and operational efficiency [1]. Compliance checking is an emerging process mining technique that verifies whether business process executions adhere to predefined rules. These rules represent standards, policies, laws, or performance requirements that pertain to the process [2]. A compliance checking technique consists of two main steps. The first step foresees the definition of structured compliance rules based on compliance requirements that need to be checked. The second step consists of checking the defined rules against the event log containing the execution data. Proposed compliance rule languages [3, 4] are mainly focused on checking a single process in a single organization. In this regard, object-centric process mining has been proposed to overcome the difficulties of dealing with complex processes that involve multiple interacting objects. In distributed scenarios, compliance checking becomes even more challenging due to the involvement of multiple independent participants, each maintaining their own processes. Trust among participants and data integrity issues represent additional challenges to address. In this regard, blockchain technology can guarantee distributed control of business process execution and a trustworthy environment due to the immutability of transactions. These characteristics enable transparent auditability of distributed business processes using compliance checking techniques.

This research aims to address these challenges by proposing advancements on the compliance checking topic for both the fields of object-centric process mining and blockchain technology. Object-centric process mining enables a structured representation of process execution, which aligns well with blockchain's immutable transaction records. Indeed, smart contract execution data reflects the business logic of DApps, as process execution data captures the operational behavior of business processes in business process management systems [5].

Proceedings of the Best BPM Dissertation Award, Doctoral Consortium, and Demonstrations & Resources Forum co-located with 23rd International Conference on Business Process Management (BPM 2025), Seville, Spain, August 31st to September 5th, 2025

✉ alessio.galassi@unicam.it (A. Galassi)

ORCID 0009-0005-8782-1593 (A. Galassi)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

We envision developing a comprehensive compliance checking framework that integrates object-centric process mining potentialities and blockchain environment characteristics. The research questions guiding this study include:

RQ1 How can compliance rules be effectively formalized within object-centric process mining to represent real-world compliance requirements?

RQ2 How can compliance checking mechanisms be designed to operate efficiently in the blockchain environment?

After presenting the research problem, Section 2 describes the approach and proposed solution. Section 3 details the relation to the state of the art. Finally, Section 4 illustrates results achieved so far, a roadmap for the remainder of the PhD project, and open issues.

2. Research Approach and Solution

The research journey started with the consolidation of a collaboration between the University of Camerino and the INGKA Group (<https://www.ingka.com/>), i.e., a leading multinational home furnishing retailer controlling 379 stores worldwide. The project focused on answering business challenges in the stock check process, which is performed daily in the INGKA Group stores and produces, on average, 450,000 inventory-related events per day. The experience revealed a significant gap in using compliance checking techniques in industrial real-world scenarios and increased awareness of the stakeholders' perception of applying compliance checking for business process management [6]. From this experience, we acquired the needed competencies to start exploring the fields and kick off this doctoral project.

We envision this doctoral project consisting of a four-step methodological approach: State-of-the-Art study, approach design, formalization, and validation. However, the plan may be fixed, and some adjustments are likely to occur depending on the opportunities and advancements in the fields. An overview of the project phases and the current status is presented in Table 1.

Initially, a comprehensive literature review was conducted to analyze the actual state-of-the-art about compliance checking techniques and the theoretical research of blockchain technology to explore possible combinations that leverage both technologies. This review revealed limitations in current compliance rule languages, particularly their inability to capture complex interactions among multiple entities. Given the community efforts and attention to the new object-centric paradigm, this phase cannot be considered closed, since new advancements and proposals in the field are constantly under observation. Indeed, the shift from case-centric to object-centric process mining is a promising step to also solve the compliance checking of complex process behaviors. Building on these insights, we plan to design a new technique to formalize compliance rules that must exploit the object-centric concepts that are the reasons for the change of paradigm. The problems that object-centric process mining intends to solve are: *deficiency*, *convergence*, and *divergence*; which are addressed by the introduction of object entities (as important as events), event-to-object relations, and object-to-object relations, that

Phase	Overview	Status
State-of-the-Art	Analyze existing approaches in compliance checking, object-centric process mining, and blockchain-based process management	Neverending
Design	Develop novel techniques exploiting object-centric mining and blockchain to enhance compliance checking	In progress
Formalization	Define formal models and rules to capture compliance logic and distributed process behaviors	In progress
Validation	Evaluate the approach via case studies and simulations to assess effectiveness and scalability	Pending

Table 1
Planned phases of the doctoral project.

enhance the expressivity of the representation of process in the event log. These new entities and components in the object-centric event log are the cornerstones for increasing the effectiveness of compliance checking techniques to represent real-world compliance requirements (RQ1). Regarding the decentralization of compliance checking, a feasible solution is represented by the intertwined use of such techniques with the mechanisms of smart contract execution. Indeed, smart contracts could allow auditors to agree on what and how to check compliance rules without relying on a central authority, thus guaranteeing transparency and security (RQ2). Once the complete approach is designed and a suitable compliance rule language formalized, the overall validation will be strategically planned through a series of case studies representative of both industrial and governmental contexts. These domains are characterized by complex regulatory environments, where compliance requirements span organizational boundaries and involve multiple, often heterogeneous, stakeholders. The evaluation will go beyond functional verification; it will critically assess the effectiveness of the approach in ensuring secure, transparent, and scalable compliance checking. Where possible, objective criteria such as correctness and performance will be used to evaluate the outcomes. The case studies will also help highlight any practical challenges or limitations of the approach, informing potential refinements and contributing to a clearer understanding of its applicability in real-world contexts.

3. Relation to the State-of-the-Art

The State-of-the-Art analysis explored both the fields of compliance checking (in case-centric and object-centric settings) and blockchain-based process management.

Compliance checking. The most relevant traditional compliance checking techniques are the extended Compliance Rule Graph (eCRG) [3] and DECLARE [4]. The eCRG [3] is a visual notation for compliance rule modeling covering all process perspectives, i.e., control flow, interaction, time, resource, and data. An eCRG is an acyclic graph composed of an antecedence pattern and at least one consequence pattern. These patterns are represented by nodes indicating event or process element occurrences or absences, with edges denoting control flow dependencies. DECLARE [4] is a declarative process modeling language that utilizes constraint-based specifications to define flexible workflows. It focuses on what must or must not happen rather than prescribing a strict sequence of activities and provides a set of predefined templates that express common workflow constraints. Nevertheless, the shift toward object-centric process mining has highlighted the necessity for techniques tailored to object-centric processes [7]. In this regard, Park and van der Aalst [8] define object-centric constraint graphs to perform monitoring operations over an object-centric event log and obtain behavioral metrics. The graphs can define the ordering relation between events related to the same object, event-to-object relations, and performance constraints. However, the technique cannot handle object-centric flow patterns over multiple objects and temporal constraints. van der Aalst et al. [9] propose the object-centric behavioral constraints model, a unified representation of processes and data-related constraints. The model combines ideas from declarative constraint-based languages and a general-purpose visual data/object modeling language for information systems. They incorporate multiple business process compliance perspectives such as control-flow on a single object, data dependencies, and object-to-object relations. Even this work suffers from the absence of temporal and performance constraints. Lastly, Li et al. [10], introduce a technique for graphically defining constraints on process executions extracted from object-centric event logs. The models enable the definition of the desired cardinality between two event types, or temporal and performance constraints. However, the technique does not consider object-to-object relations and constraints on events' or objects' attributes, thus discarding part of the knowledge contained in the log. In summary, although compliance checking is advancing towards the object-centric domain, there are still gaps in managing, especially flow patterns involving multiple objects, which prevents fully leveraging object-centric event data.

Blockchain-based process management. Blockchain technology, particularly smart contracts, has been proposed as a means of ensuring data integrity and trust in distributed environments. This research builds upon prior work by integrating blockchain-based audit trails with object-centric compliance verification, addressing gaps in existing methodologies. Over the years, smart contract auditing has attracted significant interest, leading to the development of various approaches [11], mostly tackling the problem from a code implementation perspective. Established methods include symbolic execution, formal verification, and fuzzing static analysis. While effective, these approaches focus on analyzing smart contract code at design time, mainly to find vulnerabilities or bugs and test behaviors. Differently, our aim is to focus on the data produced by the execution of the smart contracts [12], which represent the DApp’s logic that needs to be checked in order to reveal unexpected behaviors. In the direction of executing data analysis, few works are currently available. HighGuard [13] is a runtime monitoring tool in cross-chain environments. It utilizes Dynamic Condition Response (DCR) graph models, enhanced with data and time, as formal specifications to verify contract executions against predefined business processes. DCR graphs can model smart contract semantics by mapping functions to activities and require statements to guarded relations. Other data taken into analysis are time constraints and smart contract roles, i.e., sender or receiver, mapped respectively into relations with deadlines and DCR model roles. Using DCR graphs, HighGuard identifies transactions that violate the specifications. However, while the enriched DCR graphs provide a graphical and usable notation to specify requirements, the lack of specific blockchain characterizations and concepts is a major limitation. Differently, we aim at providing a language and related approach that considers blockchain components as first-class citizens and enables the specification of related compliance rules. Shyamasundar [14] presents a framework performing automatic identification of intended behavior using the developer’s code annotations. Annotations are phrases specified within the Solidity code and that are used to generate an appropriate runtime monitor. The phrases reflect the constraints on the execution of programs by observing the relative objects involved, such as methods, calls, or exceptions. In order to perform the monitoring, Solidity annotations are used. This kind of annotation requires technical expertise to express desired constraints, representing a barrier for non-expert auditors. Additionally, this requires access to the smart contract and precise knowledge of its structure to specify what to check. With CoBLOCK we shift the focus from the design-time development to the post-deployment execution, creating rules directly on generated data. In this way, checks are specified on general logic and are unbounded from the smart contract code. DAppHunter [15] introduces an approach for identifying behavioral inconsistencies in DApps on EVM-compatible blockchains through the analysis of interactions derived from the front-end, blockchain wallet, and smart contracts. The authors propose an intention-driven approach utilizing a 2-layer intention-action graph, which consists of a high-level user intention graph and multiple low-level front-end action graphs to explore feasible interactions with the front-end of DApps. In this approach, the main issue is the correlation between blockchain execution data and user actions, including in the front-end. Furthermore, the analysis is currently limited to standard DApp functionality, making it impossible to audit complex logic and observe fine-grained data. In contrast, our approach shifts from design-time verification to post-deployment execution analysis. Defining compliance rules directly on smart contract execution data enables more flexible, fine-grained assessment of DApp behavior, independent of contract structure or interaction assumptions.

4. Progress, Open Issues, and Roadmap

Progress. The in-progress *design* and *formalization* phases include two preliminary studies.

The first one aims to propose a language to specify object-centric compliance rules, encompassing events based on specific object relations, attribute values, and exploiting flow relations across multiple objects. Particularly, it addresses limitations inherent to case-centric process mining approaches by introducing a novel method to relate events even without a single case-id to build a process instance. The related artifact is The O.C.³ (The Object-Centric Compliance Checking Tool), which is a web application for the definition of object-centric compliance rules and the application of such rules against

an object-centric event log. The graphical interface enables users to: load an object-centric event log, create and apply object-centric compliance rules to generate matching and non-matching event sets.

In the second study, we broadened the application of compliance checking to data coming from smart contract execution. To this aim, we introduced CoBLOCK, a domain-specific language designed for compliance checking in smart contract execution on blockchain. It highlights the challenge that existing compliance checking approaches face due to the lack of blockchain-specific rule specification languages. To address this issue, the work proposes CoBLOCK as a solution, allowing compliance rules to explicitly incorporate blockchain characteristics. We detailed the language's design and demonstrated how it can effectively be used to check smart contract behaviors against predefined compliance requirements.

Open Issues. The newly arising object-centric process mining is currently accompanied by a *lack of a general standard* for the representation of *object-centric event logs* [16]. This issue causes some uncertainty regarding the implementation reference to use for building new process mining techniques, such as The O.C.³. Indeed, The O.C.³ is designed and implemented to fit with the latest object-centric event data metamodel [17], but this choice has its advantages and drawbacks. The advantage consists of the suitability of the technique to any object-centric event log implementation based on the same object-centric event data metamodel; the drawback resides in the untapped potential of specific object-centric event log characteristics, e.g., the object's lifecycle via timed attributes in OCEL 2.0 [18]. For this purpose, we are watching out for any new proposal of object-centric event data metamodel or object-centric event log implementations, as it would require adapting already designed techniques.

Another open issue is represented by the *complexity of compliance requirements* to be modeled into compliance rules. From the broad range of fields of application of object-centric process mining stems a similarly wide set of compliance requirements, each with the peculiarities of the topic on which they rely. In this case, the literature offers some cues for modeling behaviors belonging to distributed scenarios [19], which could provide a solid basis for extending compliance rule languages.

A relevant issue that we experienced during the project with INGKA Group is that *tools* from academia are not always suitable for being employed in real industrial settings. A balance between the expressivity of new techniques and the ability to handle large datasets would enhance the relevancy and the applicability in real contexts where those techniques are highly demanded.

The last issue regards the enhanced transparency and security of compliance checking performed on-chain, thus exploiting the characteristics of the blockchain environment. It is notable to consider problems with the usage of blockchain due to the costs of storing data on-chain, the cost to compute the check, and the cost of changes to rules already deployed [20].

Roadmap. The following are the roads we intend to explore to pursue the doctoral project.

One combines the in-progress works to obtain a language for *object-centric compliance checking for blockchain data*, treating interacting smart contracts as multiple processes across organizations.

Another improvement could be considered to improve the *usability* of the proposed compliance rule languages. The idea is to propose a visual notation, e.g., a block-based notation, for building compliance rules that should ease the usage for non-practitioners or non-experts in auditing. It may be relevant to conduct a study on the usability and understandability of the most used compliance checking approaches, and compare them with the newly proposed block-based compliance rule language.

An additional direction of the research is the role of *smart contracts in automating compliance checking*. By embedding compliance rules within smart contracts, we could explore the potential of decentralized compliance checking mechanisms, where predefined conditions trigger automated checking actions. This advancement will facilitate exploring decentralized compliance requirements, where participants collaboratively define and check compliance rules over their processes without a centralized authority.

Finally, an *empirical validation* will be conducted to demonstrate the applicability and effectiveness of the proposed approach in real-world industrial scenarios. This will include evaluating its performance in diverse scenarios, assessing its usability, and gathering feedback from domain experts to refine the framework further.

Declaration on Generative AI

During the preparation of this work, the author used DeepL, and Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the author reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] W. M. P. van der Aalst, Decision support based on process mining, in: Handbook on Decision Support Systems 1, Springer, Berlin, 2008, pp. 637–657.
- [2] J. van der Werf, H. Verbeek, W. M. P. van der Aalst, Context-aware compliance checking, in: Business Process Management, volume 7481 of *LNCS*, Springer, Berlin, 2012, pp. 98–113.
- [3] D. Knuplesch, M. Reichert, L. T. Ly, A. Kumar, S. Rinderle-Ma, Visual modeling of business process compliance rules with the support of multiple perspectives, in: International conference on conceptual modeling, Springer, 2013, pp. 106–120.
- [4] M. Pesic, H. Schonenberg, W. M. Van der Aalst, Declare: Full support for loosely-structured processes, in: 11th IEEE international enterprise distributed object computing conference (EDOC 2007), IEEE, 2007, pp. 287–287.
- [5] M. Weske, Business Process Management - Concepts, Languages, Architectures, Third Edition, Springer, 2019.
- [6] A. Galassi, B. Re, S. Reimann, L. Rossi, Compliance Checking in Action for INGKA Group Inventory Management, in: Industry & Society Forum (Business Process Management 2025), Seville, 2025.
- [7] W. M. P. van der Aalst, Lifting process discovery and conformance checking to the next level: A general approach to object-centric process mining, in: Workshop on Petri Nets and Software Engineering, volume 3730, CEUR.ws, 2024, pp. 1–12.
- [8] G. Park, W. M. P. van der Aalst, Monitoring constraints in business processes using object-centric constraint graphs, in: International Conference on Process Mining, volume 468 of *LNBIP*, Springer, 2022, pp. 479–492.
- [9] W. M. P. van der Aalst, G. Li, M. Montali, Object-centric behavioral constraints, arXiv preprint arXiv:1703.05740 (2017).
- [10] T. Li, G. Park, W. M. P. van der Aalst, Checking constraints for object-centric process executions, in: International Conference on Process Mining, volume 503 of *LNBIP*, Springer, 2023, pp. 392–405.
- [11] G. Wu, H. Wang, X. Lai, M. Wang, D. He, S. Chan, A comprehensive survey of smart contract security: State of the art and research directions, Journal of Network and Computer Applications (2024) 103882.
- [12] F. Corradini, A. Marcelletti, A. Morichetta, B. Re, A methodology for extracting and decoding smart contracts data, Computer Communications 240 (2025) 108204.
- [13] M. Eshghie, C. Artho, H. Stammler, W. Ahrendt, T. T. Hildebrandt, G. Schneider, Highguard: Cross-chain business logic monitoring of smart contracts, in: IEEE/ACM ASE, 2024, pp. 2378–2381.
- [14] R. K. Shyamasundar, A framework of runtime monitoring for correct execution of smart contracts, in: S. Chen, R. K. Shyamasundar, L.-J. Zhang (Eds.), CBC 2022, Springer, 2022, pp. 92–116.
- [15] J. Zhou, T. Jiang, H. Wang, M. Wu, T. Chen, Dapphunter: Identifying inconsistent behaviors of blockchain-based decentralized applications, in: 2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), IEEE, 2023, pp. 24–35.
- [16] A. Goossens, J. De Smedt, J. Vanthienen, Object-centric event logs: Specifications, comparative analysis and refinement, arXiv (2024).
- [17] D. Fahland, M. Montali, et al., Towards a simple and extensible standard for object-centric event data (oced)-core model, design space, and lessons learned, arXiv preprint arXiv:2410.14495 (2024).
- [18] A. Berti, I. Koren, J. N. Adams, et al., Ocel (object-centric event log) 2.0 specification, arXiv preprint arXiv:2403.01975 (2024).
- [19] L. González, A. Delgado, J. Canaparo, F. Gambetta, Evaluation of Compliance Requirements for

collaborative business process with process mining and a model of generic compliance controls, CLEI Electronic Journal 25 (2022) 7:1–7:22.

- [20] A. A. Zarir, G. A. Oliva, Z. M. J. Jiang, A. E. Hassan, Developing cost-effective blockchain-powered applications: A case study of the gas usage of smart contract transactions in the ethereum blockchain platform, ACM Trans. Softw. Eng. Methodol. 30 (2021).