

PC-VERA: Process Compliance Verification, Extraction, and Repair

Johannes Loebbecke^{1,*}, Marisol Barrientos¹, Karolin Winter², Juergen Mangler¹ and Stefanie Rinderle-Ma¹

¹Technical University of Munich, School of Computation, Information, and Technology, Boltzmannstr. 3 85748, Garching, Germany

²Eindhoven University of Technology, Department of Industrial Engineering and Innovation Sciences, Groene Loper 3 5612 AE, Eindhoven, The Netherlands

Abstract

Adhering to compliance constraints is essential for all companies regardless of size or domain. Business Process Compliance (BPC) uses Business Process Management (BPM) techniques to ensure that a company's processes adhere to compliance constraints. However, existing approaches generally address only one or a few aspects of BPC, such as requirements extraction or model verification, and typically have low technology readiness. We present a compliance suite integrated into a process execution engine that supports extracting constraints from natural language and using them to verify and repair the compliance of process models. The resulting suite combines the automation capabilities of LLM-based requirements extraction and compliance repairing, with the trustworthiness of deterministic verification and engine-integrated modeling, simulating, and execution.

Keywords

Business Process Compliance, Compliance Verification, Requirements Extraction, Compliance Repair

1. Introduction

Compliance with constraints imposed by regulations and legal contracts remains a critical challenge for organizations across all domains. Furthermore, the increasing frequency of newly added or changed regulations presents unique challenges [1]. BPC aims to use BPM techniques to ensure compliance by orchestrating services according to business process models in formats such as BPMN or process trees and verifying the process models w.r.t. requirements extracted from Natural Language (NL) sources. While Large Language Models (LLMs) offer a high automation potential for BPC, as rules can be automatically extracted from NL and used to verify process executions [2] or generate compliant processes [3] directly from text, the unreliability of generative AI-based solutions leads to low trustworthiness [4]. Meanwhile, BPC research has presented trusted verification methods [5, 6, 7], but no approaches combine LLM-based requirements extraction with trusted verification. Furthermore, few approaches have focused on repairing violations [8], i.e., repairing a process that is currently violating compliance constraints to a compliant state. Finally, surveys indicate that compliance approaches generally have a low level of technology readiness [9]. In particular, modeling and execution are often supported in tools separate from those used for compliance extraction and verification. As a result, users must repeatedly export and import process models between different tools.

To address these gaps and limitations, we present PC-VERA, an extensive compliance suite supporting LLM-based requirements extraction from NL rules and textual process descriptions, deterministic compliance verification based on structural tree verification, and LLM-based compliance repairing. Based on several BPC contributions [2, 7, 8, 10] PC-VERA balances the automation potential of LLMs with the trustworthiness of deterministic, model-based verification. Through integration into an open-source Cloud Process Execution Engine (CPEE¹), the tool is at a high technology readiness level, as verified

ER2026: Companion Proceedings of the 45th International Conference on Conceptual Modeling: Doctoral Consortium, Tutorials, Posters and Demos, October 5-8, 2026, St. John's, NL, Canada.

*Corresponding author.

✉ johannes.loebbecke@tum.de (J. Loebbecke); marisol.barrientos@tum.de (M. Barrientos); k.m.winter@tue.nl (K. Winter); juergen.mangler@tum.de (J. Mangler); stefanie.rinderle-ma@tum.de (S. Rinderle-Ma)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹cpee.org, last accessed: 27 July 2026

and repaired processes can be directly modified or used for service orchestration. The tool is publicly available through an online demo at <https://cpee.org/redir/pc-vera>, where PC-VERA functionalities are highlighted in red, and the source code is publicly provided on GitHub².

2. Process Compliance Extraction, Verification, and Repair

As illustrated in Fig. 2, PC-VERA provides requirements extraction, compliance verification, and compliance repair in a single tool, directly integrated with process modeling and execution.

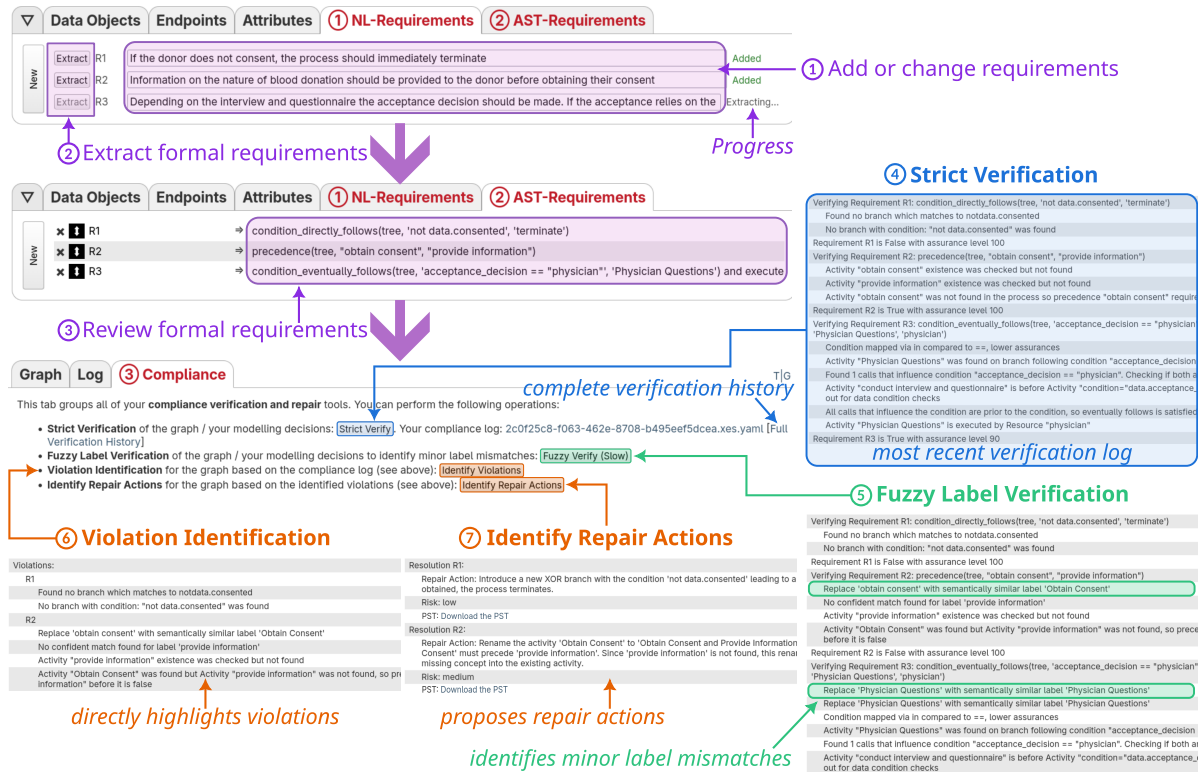


Figure 1: PC-VERA UI Overview: Modelers can change or add requirements in the NL-Requirements tab. Extracted ASTs can be reviewed and modified in the AST-Requirements tab. Below the Requirements tab, modelers can view and edit the model and inspect compliance verification and repair results.

Central to process compliance are the requirements to which the process execution must adhere. In PC-VERA, a list of NL requirements can be extended or modified in the NL-Requirements tab ①. To enable verification, the NL requirements can be transformed via LLM-based extraction to formal Abstract Syntax Trees (ASTs) via the *Extract* ② buttons next to each requirement. The extracted ASTs are displayed in the AST-Requirements tab, and can be reviewed and adjusted as necessary ③. Furthermore, the modeler can change or edit the process model in the graph tab below the requirements tab, including data objects, conditions, and resource assignments. Meanwhile, the verification and repair functionalities can be found in the Compliance tab. Central to the compliance functionalities is the strict structural verification, as described in [7], which can be called via the verify button ④. It verifies the compliance of the process w.r.t. the requirements based on intertwined tree traversal of the ASTs, representing the compliance requirements, and the Process Structure Tree [11], representing the process model. The verification results are summarized in an XES-compatible compliance log, which contains the rules, the verification result per rule, and a history of verification steps. Modelers can either review the most recent compliance log in a human-readable format via the *Verify* button ④ or view and download the complete verification history. The complete verification history is created

²<https://github.com/etm/CPEE/blob/cpee2/cockpit/compliance.html>, <https://github.com/JohannesLbck/ProcessTreeVerify>, <https://github.com/marisol-barrientos/PTRepair>, last accessed: 27 July 2026

through a subscription on every process model change, enabling mining of the process of modeling a compliant process using process mining [12].

Strict verification uses strict label and data-object matching between requirements and the process model. While this leads to stronger verification correctness guarantees, it does not account for minor label mismatches resulting from spelling differences. As such, PC-VERA also supports verification based on fuzzy label-matching using a publicly available semantic matching model³. As this fuzzy verification is slower and more error-prone, it is not called on every process model change, but can be used via the *Fuzzy Verify* button (5). We recommend using fuzzy verification after extracting requirements from text to identify and fix small label mismatches, and then using strict verification.

Finally, the compliance log and the process model can be passed to a repairing pipeline described in [8], via the *Identify Violations* (6) and *Identify Repair Actions* (7) buttons. Identify violations deterministically parses the compliance log to identify specific violations per requirement. The identified violations can then be used to identify repair actions via LLM-based processing. The repair actions textually describe what actions have to be taken to repair the process model, while the created trees can be reviewed and used to create a final compliant model. As the repaired models may overlook requirement conflicts and are more prone to hallucinations [8], they require careful review.

Fig. 2 highlights how the tool’s architecture combines the automation potential of LLM-based processing with the trustworthiness of deterministic verification, while enabling human oversight through various review and adjustment options. The compliance requirements are extracted via an LLM-based service. The process model can either be manually created, mined from event logs using process mining [12], or extracted from text using LLM-based Text2Model (T2M) approaches [3, 13, 14, 15]. The process model is verified w.r.t. the AST-Requirements via deterministic, structural verification. Finally, the repair pipeline combines deterministic and LLM-based processing steps to suggest compliant models. The modeler (or a specific domain expert, e.g., from the legal domain) can review and edit any intermediate results to ensure tractability.

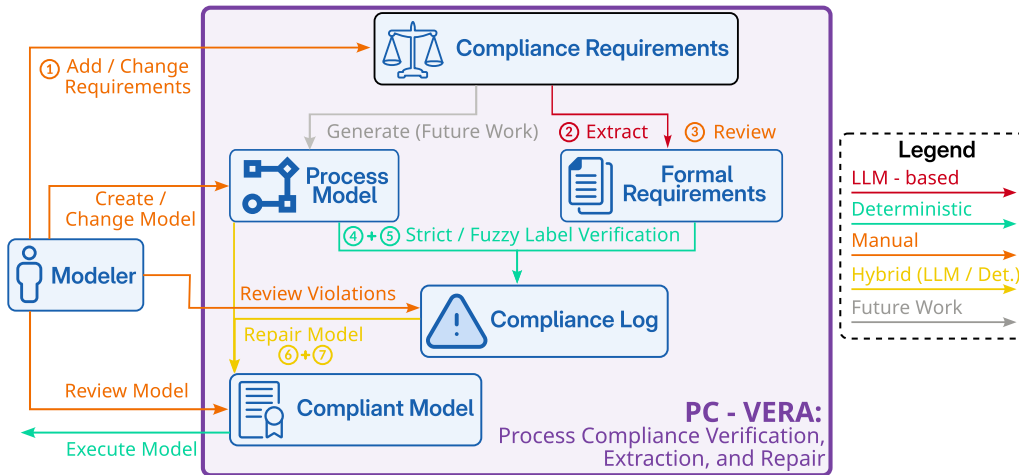


Figure 2: Overview of PC-VERA Architecture: Formal requirements are extracted from textual sources and used for structural process verification. LLM-based extraction is balanced with deterministic verification.

3. Maturity

Scientific Maturity The compliance verification approach is based on structural tree verification discussed in [7]. The LLM-based tree repair pipeline is presented in our paper at this year’s Conference on Conceptual Modeling [8]. Finally, the requirements extraction is custom-made for the tool based on lessons learned regarding label matching and extraction prompts from existing extraction approaches [16,

³<https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2>, last accessed: 27 July 2026

17]. E.g., we do not provide the process model to requirements extraction, as this would substantially increase tokens (costs) and risk overfitting, while label mismatches can be identified during verification. **CPEE Maturity** The CPEE provides the open-source execution and interface backbone of the presented compliance suite. Together with its connected components, it has been downloaded 1,901,161 times and is currently used for process orchestration by seven manufacturing companies [18].

PC-VERA Maturity The verification and repair steps have been verified on a composite dataset of 11 processes with 46 associated requirements covering the typical process perspectives of control flow, data, resource, and time in [7, 10] and [8], respectively. However, there are still several limitations for general industrial applicability: the evaluation dataset was synthetic and comparatively small, and the application would benefit from integration with organization-specific resource management tools. Furthermore, verification currently relies on heuristics for handling data conditions with lower correctness guarantees [7]. Here, the tool could be extended to alternate between tree-traversal, AST-based verification and state-exploration, LTL-based verification depending on the requirement [10]. At the same time, repair only provides compliant models per requirement, without resolving potential conflicts between repair actions [8]. Finally, LLM-based processing, particularly compliance repair, is influenced by the quality of the used model. For the publicly available demo, we use a smaller, less costly model (gemini-2.5-flash) which appears less consistent in correctly identifying which part of a requirement is the condition and which is the consequence and correctly creating repaired processes when multiple changes have to be applied. A feature for modelers to provide their own API keys is currently not included due to security risks.

4. Videos and Tool

PC-VERA can be tested directly through the live instance at <https://cpee.org/redir/pc-vera>, and a video demonstrating the tool’s usage is available at <https://cpee.org/videos/PC-VERA.mp4>. In addition, modelers can also create new process models or experiment with existing models in the CPEE Process Hub⁴. In general, modelers can access PC-VERA in the CPEE by pressing *Draft* next to the CPEE logo at the top right and selecting *Compliance* in the opened drop-down menu.

5. Conclusion

We present PC-VERA, an extensive, web-based compliance suite integrated into an open-source process execution engine. PC-VERA supports requirement extraction from natural language, structural verification based on model traversal, and functionalities to help modelers with repairing non-compliant process models. As PC-VERA is integrated into an open-source process execution engine, modelers can also directly review, edit, and execute the process model at any step in the pipeline.

In the future, we plan to extend PC-VERA with additional visualizations for requirements, compliance violations, and repaired processes. Furthermore, the verification and repair steps do not currently consider all aspects of process compliance, such as vague constraints that rely on context information for correct formalization [19], or risk assessment in cases of probabilistic compliance issues, such as over-compliance [20]. Finally, extensive research has been conducted on LLM-based T2M conversion [3, 13, 14], which can be used with PC-VERA but is currently not directly integrated. Specifically, [3] and [14] are well-suited, as they extract process models in the same PST format used by PC-VERA.

Declaration on Generative AI and Acknowledgements

During the preparation of this work, the author(s) used Grammarly to check grammar and spelling. The authors reviewed and edited the content as needed and take full responsibility for the content.

⁴<https://cpee.org/hub/?stage=compliance&dir=Staff.dir/Loebbi.dir/Demo.dir/>, last accessed: 27 July 2026

This study was funded as part of the TRPro project by the German Research Association (Deutsche Forschungsgemeinschaft) as "Sachbeihilfe" under the project number 514769482.

References

- [1] M. Barrientos, J. Loebbecke, K. Winter, S. Rinderle-Ma, Taming the complexity of legal change for business process compliance, *Business and Information Systems Engineering* (2026).
- [2] M. Barrientos, K. Winter, S. Rinderle-Ma, Impact analysis of regulatory requirement changes on business process compliance, *Information and Software Technology* 194 (2026).
- [3] A. Damaratskaya, C. Sai, S. Rinderle-Ma, From regulatory text to conceptual models: Generating organizational models, role-task mappings and process descriptions, in: *Conceptual Modeling*, Springer, 2026. Accepted for Publication.
- [4] S. de Kinderen, Q. Ma, J. S. Mercado, K. Winter, The interim experiment report: A systematic account of the experimental design of large language model-based text-to-model approaches, in: *IFIP*, volume 570, Springer, 2025, pp. 102–120. doi:10.1007/978-3-032-12063-2_7.
- [5] A. Awad, M. Weidlich, M. Weske, Visually specifying compliance rules and explaining their violations for business processes, *J. Vis. Lang. Comput.* 22 (2011) 30–55.
- [6] S. C. Tosatto, H. Burke, N. van Beest, H. Groefsema, A first and fast symbolic approach for data-aware business process compliance checking, *Information and Software Technology* (2026) 108047. doi:<https://doi.org/10.1016/j.infsof.2026.108047>.
- [7] J. Loebbecke, J. Mangler, S. Rinderle-Ma, Tree-based compliance verification: Bridging the gap between compliance requirements and process execution, in: *Conceptual Modeling*, volume 16189, Springer, 2025, pp. 185–203.
- [8] M. Barrientos, K. Winter, S. Rinderle-Ma, Optimizing compliance violation resolution in business process models, in: *Conceptual Modeling*, Springer, 2026. Accepted for Publication.
- [9] H. A. López, T. T. Hildebrandt, Three decades of formal methods in business process compliance: A systematic literature review, *CoRR* abs/2410.10906 (2024). [arXiv:2410.10906](https://arxiv.org/abs/2410.10906).
- [10] J. Loebbecke, J. Mangler, S. Rinderle-Ma, Taming the complexity of design time process compliance through structural tree-based multi-perspective verification, *Data Knowledge Engineering* 166 (2026) 102639. doi:<https://doi.org/10.1016/j.datak.2026.102639>.
- [11] J. Vanhatalo, H. Völzer, J. Koehler, The refined process structure tree, in: *Business Process Management*, 2008, pp. 100–115.
- [12] W. M. P. van der Aalst, Process mining, in: *Encyclopedia of Database Systems*, Second Edition, Springer, 2018.
- [13] A. Costa, A. Wimmer, L. Pufahl, Llm4bpmngen: A tool for BPMN generation with llms, in: *Conference on Conceptual Modeling*, volume 4099, CEUR-WS.org, 2025, pp. 333–337.
- [14] N. Klievtsova, M. Ehrendorfer, J. Mangler, S. Rinderle-Ma, Autobpmn.ai: Conversational process modeling and automation, in: *Business Process Management*, volume 4032, CEUR-WS.org, 2025.
- [15] C. Sai, S. Rinderle-Ma, Agentic generation of process models from regulatory texts, in: *CoopIS*, volume 15535, Springer, 2025, pp. 370–387. doi:10.1007/978-3-032-15538-2_21.
- [16] C. Sai, K. Winter, E. Fernanda, S. Rinderle-Ma, Detecting deviations between external and internal regulatory requirements for improved process compliance assessment, in: *Advanced Information Systems Engineering*, 2023, pp. 401–416. URL: https://doi.org/10.1007/978-3-031-34560-9_24.
- [17] K. Winter, H. van der Aa, S. Rinderle-Ma, M. Weidlich, Assessing the compliance of business process models with regulatory documents, in: *Conceptual Modeling*, 2020, pp. 189–203.
- [18] S. Rinderle-Ma, J. Mangler, Process automation and process mining in manufacturing, in: *Business Process Management*, volume 12875, Springer, 2021, pp. 3–14.
- [19] J. Caballero-Villalobos, H. Baumeister, E. Paja, O. Kokoulina, H. A. López, Aligning processes with high-level requirements: Goal-model-based compliance checking (extended version), *SSRN* (2026).
- [20] J. Loebbecke, S. Rinderle-Ma, A framework for assessing overcompliance and undercompliance in business processes, in: *CoopIS*, volume 15535, Springer, 2025, pp. 202–220.