# Kogi: A tool for assessing High-Level Business Process Compliance*

Juanita Caballero-Villalobos[1], Hugo A. López[1]

[1]Technical University of Denmark, DTU Compute, Denmark

## Abstract

The goal of a business process is to orchestrate activities that achieve a specific business objective. However, most process simulation tools do not assess how activities influence the satisfaction of business goals. *Kogi*[1] addresses this limitation by aligning imperative and declarative process models with goal models to evaluate compliance with high-level and non-functional requirements. Unlike existing tools, Kogi traces how process executions affect goal satisfaction in both runtime and design-time scenarios. The tool focuses on monitoring the fulfillment of organizational objectives rather than procedural correctness alone. This support shows potential to improve traceability and interpretability of compliance outcomes and enhance communication across stakeholders involved in the business process lifecycle.

## Keywords

Business Process Compliance, Business Process Monitoring, Goal-driven Compliance, I-star models, WF-nets

## 1. Introduction

The workflow specified in business process models defines how organizational goals are operationalized. These goals can be classified into high-level business requirements or non-functional requirements, such as strategic objectives, compliance expectations, or performance constraints. Goal-oriented analysis is crucial for strategic decision-making at the executive level, where the focus is not on specific actions but on achieving broad business objectives. Current commercial platforms for process execution and simulation (i.e., SAP Signavio[1], Camunda[2], DCR Solutions[3]) primarily emphasize activity execution and process automation, without unveiling how those process activities contribute toward goal satisfaction.

To the best of our knowledge, the approaches that have been developed to align process activities and organizational goals focus on the correctness of the design of the process and goal model by capturing the behaviours of each other [1, 2, 3] (i.e., goal decomposition, sequence order, constraints). Still, no work has been carried out so far specifically monitoring the composed evolution of the satisfaction of goals based on process execution as based on business process meta-models insights [4]. Moreover, all conformant traces identified for most compliance checking approaches [5, 6, 7] are treated as equally satisfactory, without distinguishing which compliant executions are more desirable or better aligned with organizational priorities [8]. For instance, consider a municipal caseworker handling construction permits. They follow a set procedure with activities *A*, *B*, and *C*. This process leads to either acceptance, rejection, or further investigation of the case. They complete activities *A*, *B*, and *C* as required. Regardless of the outcome, they are compliant with the procedure. However, if we look at the case from the perspective of the caseworker's interests, like wanting to save time on decisions, or the municipality's goal to promote urban development, we see a different picture. They prefer a *steady* increase in acceptance rates. This shows that not all compliant outcomes are equally favorable based

[1]Pronounced *KOH-gee*, or Cogui, in honour of the Indigenous group that resides in the Sierra Nevada de Santa Marta mountains in northern Colombia.

*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.*

jcavi@dtu.dk (J. Caballero-Villalobos); hulo@dtu.dk (H. A. López)
0000-0002-4915-0961 (J. Caballero-Villalobos); 0000-0001-5162-7936 (H. A. López)



[1]https://www.signavio.com/
[2]https://camunda.com/
[3]https://dcrsolutions.net/

on the agent's goals, and a simple scenario of the hidden things in process execution that potentially compromise transparency and interpretability of compliance outcomes.

We introduce *Kogi*, a tool that addresses this gap by enabling compliance monitoring between process execution and high-level and non-functional requirements. The tool provides a graphical interface that allows users to upload their process models, goal models, and mappings between process activities and goal elements to conduct compliance analysis. Kogi is primarily designed to support legal compliance assessments, targeting process analysts and compliance officers. It aims to establish a shared vocabulary that facilitates co-creation and interdisciplinary collaboration in legal compliance contexts. However, the tool's applicability extends beyond the legal domain, depending on its configurable features. The compliance evaluation focuses on the satisfaction of the goals and follows the execution semantics described in [9]. The tool supports three main usage scenarios: (1) run-time monitoring, (2) design-time compliance evaluation, and (3) interactive what-if simulation, each enabling different modes of reasoning about compliance across the business process lifecycle. To sum up, the tool advocates for enhancing traceability between process execution and organizational goals and improving communication across the business process compliance lifecycle.

## 2. Architecture

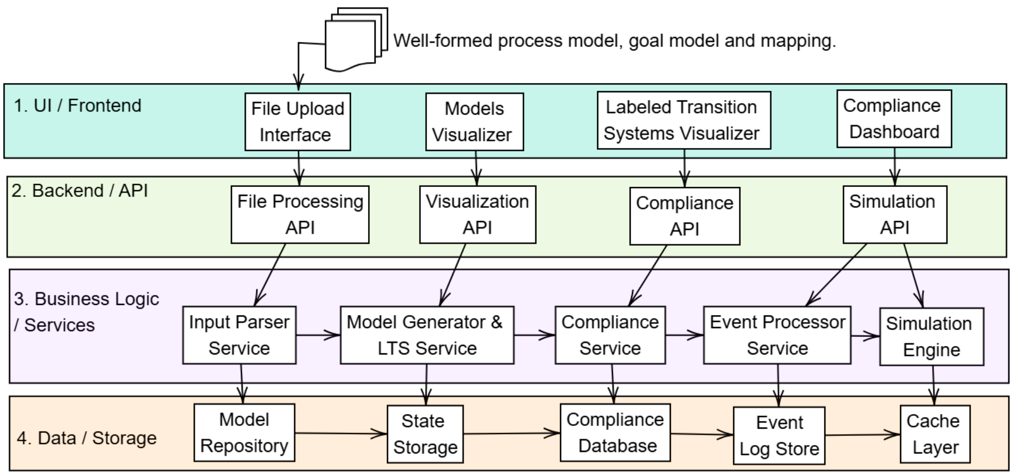


**Figure 1:** Application Architecture Overview

Kogi follows a four-layer architecture designed for flexible scalability. The system processes well-formed process models, goal models, and mapping files to provide comprehensive compliance verification across three operational modes: runtime monitoring, design-time evaluation, and what-if simulation. **Layer 1 (UI/Frontend)** offers four main components: a file upload interface for ingesting models, visualizers for process and goal models, an LTS (Labeled Transition System) viewer for exploring state spaces, and a compliance dashboard for monitoring results. Once models are uploaded, **Layer 2 (Backend/APIs)** manages input validation and rendering through RESTful services. It processes and visualizes the models, allowing users to inspect their structure. **Layer 3 (Business Logic/Services)** contains the system's core transformation logic. An input parser feeds the model generator and LTS service, which construct LTSs for both process and goal models. These are synchronized by the compliance service to generate a composed LTS representing the integrated system. Users can then initiate runtime or simulation analysis. **Layer 4 (Data/Storage)** manages persistence. The model repository stores uploaded models; state storage holds generated LTSs and traces; the compliance database logs violations; the event log store captures runtime activity; and a caching layer improves performance.

## 3. Functionality

To demonstrate Kogi's capabilities, consider the statement below. Figure 2 models the process using the identifier of the activities (i.e., ⓐ) as transition, and Figure 3, shows how some activities contribute to achieve the *goal* and *quality* of marketing's team.

> ⓐ The process begins with the registration of a customer request, which is followed by an examination phase. This examination includes ⓑ a review of the applicant's credit history and/or ⓒ validation of income and employment status to assess creditworthiness. After the registration, the ticket undergoes ⓓ a verification process that confirms the authenticity of the submitted documents and verifies the applicant's identity through Know Your Customer (KYC) procedures. ⓔ A decision can only be made once both examination and verification are completed. At this point, the process proceeds to either ⓖ approve the request and issue a credit card or ⓖ reject the request. If the available information is insufficient, the process restarts to ⓕ collect additional data.
>
> Typically, if the examination reveals that the applicant has no credit history, the credit card request is rejected. However, to attract younger credit card holders, the marketing department has launched a campaign targeting specific customer profiles. Under this initiative, selected applicants are only required to undergo ⓒ—the validation of income and employment status— to be considered for a credit card offer, regardless of their credit history. The overall objective of the marketing team is to increase credit card adoption among young adults (Ⓖ *goal*), to increase *flexibility* in onboarding young adult applicants (Ⓠ *quality*).

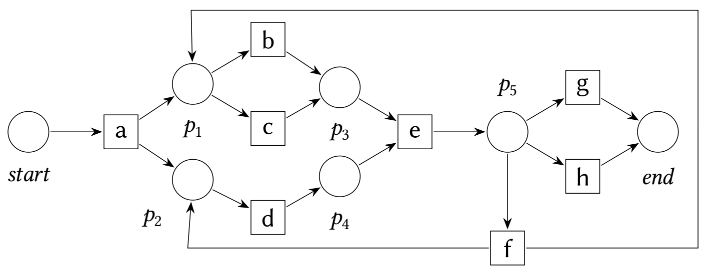


**Figure 2:** Workflow net

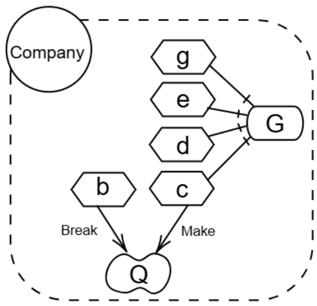


**Figure 3:** Goal model

**Run-time Monitoring** The first use case illustrates run-time monitoring of how event traces affect the satisfaction of the goal model elements. Figure 4 shows the evolution of the status of process model (Figures 2) and goal model (Figures 3) elements given a trace vs Kogi's result.

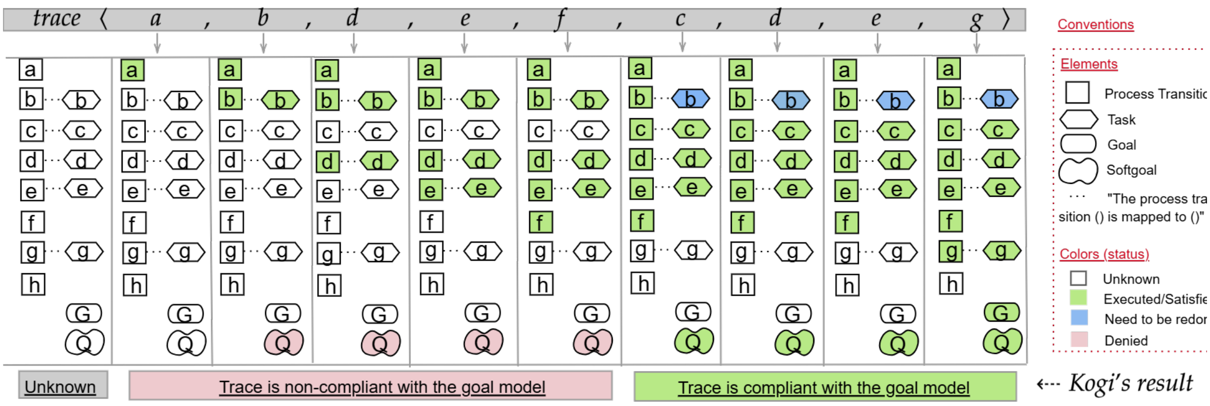


**Figure 4:** Evolution of the status of process model and goal model elements given a trace vs Kogi's result

Each event triggers a re-evaluation of goal satisfaction. Green nodes indicate satisfied elements, red indicate unsatisfied ones, and blue highlight elements marked for re-execution due to prior quality constraints[4]. The satisfaction of non-functional requirements or quality constraints ( ) may evolve, then executions initially classified as non-compliant may become compliant in subsequent traces. Moreover, Kogi supports **what-if scenario** reasoning, where users can manually trigger process events and observe the resulting changes in token flow and goal satisfaction. As shown in Figure 4, if the task g is executed, the goal G will be achieved. This enables analysts, policy designers, and compliance engineers to assess decision alternatives and failure scenarios. The goal model structure reflects how non-functional requirements (i.e., quality constraints) affect compliance: deviations in execution (i.e., time) or unmet quality goals can result in a non-compliant state, even when all functional steps are completed.

**Design-time Compliance** The second use case addresses the design-time evaluation of process models against business goals. In this scenario, the composed labeled transition system (LTS) is computed from the Petri net and synchronized with the goal model via a synchronous product construction. This enables the enumeration and assessment of all reachable execution paths. The system generates a set of compliance reports that indicate which paths satisfy, violate, or remain neutral for specific business requirements. This use case supports early-stage validation and process redesign to ensure alignment with normative goals. Figure 5 shows the interactive interface (Kogi's React version) to perform design-time compliance analysis.

**Availability** Kogi is available[5] in both React and Python versions. Figure 6 shows an embedded interactive notebook's output within the Python application, showcasing its functionality.

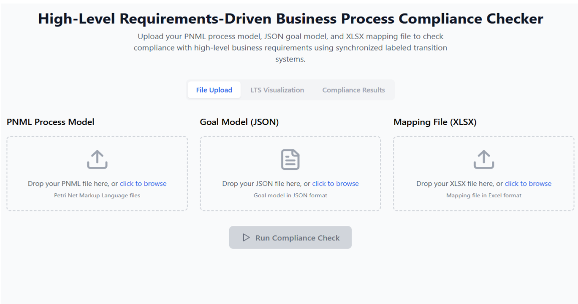


**Figure 5:** Kogi - React V2 Interface

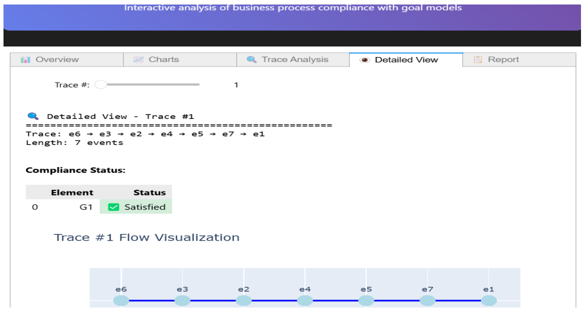


**Figure 6:** Kogi – Interactive Notebook Python V1

## 4. Maturity

Kogi's maturity is limited and was assessed using qualitative criteria relevant to software evaluation. The tool is built using a React-based interface and modular backend services that support model parsing, labeled transition system construction, and compliance reasoning. Input data, including process models in PNML, goal models in JSON, and mappings in CSV, are handled through isolated components to support maintainability and independent testing. The implementation enables three usage scenarios: runtime monitoring, design-time evaluation, and interactive simulation. Each mode presents a distinct view on compliance by linking execution traces to satisfaction states of intentional elements across the process lifecycle. Trace-level feedback is updated incrementally, allowing users to observe the propagation of effects on goal elements. A core maturity feature lies in the ability to determine whether all high-level goals are satisfied by each execution path. Additional maturity aspects relate to portability, interpretability, stability, and testability. The system is deployable through platform-independent technologies and requires no installation beyond a browser. Compliance outcomes are labeled at the

[4] see the semantic details in [9] and Kogi app details in the repository https://github.com/jc4v1/Kogi-App
[5] https://github.com/jc4v1/Kogi-App

trace level and linked to goal elements, supporting interpretation of violations and satisfactions without inspecting internal state transitions. Stability has been assessed through deterministic test cases with reproducible outcomes across execution modes. Testability is supported by the separation of model transformation, goal evaluation, and compliance checking logic, which can be verified independently.

## 5. Conclusions and Future Work

The tool operationalizes goal-driven compliance checking by synchronizing Petri net execution with goal model propagation. It supports runtime monitoring, design-time evaluation, and interactive what-if simulation. Through these functionalities, the tool allows analysts to trace the impact of process behavior on business goal satisfaction. Compliance states are computed and visualized at the level of execution traces, enabling targeted feedback. The implementation demonstrates feasibility for moderately complex models and supports use cases in audit preparation, regulatory alignment, and decision analysis. Future extensions include probabilistic analysis and a mature version of the React application, which will enhance user interface interaction.

## Acknowledgments

This work is supported by the research grant "Center for Digital CompliancE (DICE)" (VIL57420) from VILLUM FONDEN, and by the Innovation Foundation project "Explainable Hybrid-AI for Computational Law and Accurate Legal Chatbots" 4355-00018B XHAILe.

## Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

## References

[1] D. Amyot, O. Akhigbe, M. Baslyman, S. Ghanavati, M. Ghasemi, J. Hassine, L. Lessard, G. Mussbacher, K. Shen, E. Yu, Combining goal modelling with business process modelling, Enterprise Modelling and Information Systems Architectures (EMISAJ) 17 (2022).

[2] M. Ruiz, D. Costal, S. España, X. Franch, Óscar Pastor, Integrating the goal and business process perspectives in information system analysis, in: Lecture Notes in Computer Science, volume 8484 LNCS, 2014, pp. 332–346.

[3] H. Horita, H. Hirayama, Y. Tahara, A. Ohsuga, Towards goal-oriented conformance checking, in: Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE, volume 2015-January, 2015, pp. 484–489.

[4] G. Adamo, C. D. Francescomarino, C. Ghidini, Digging into business process meta-models: A first ontological analysis, in: Lecture Notes in Computer Science, volume 12127 LNCS, 2020, pp. 384–400.

[5] M. Hashmi, G. Governatori, H. P. Lam, M. T. Wynn, Are we done with business process compliance: state of the art and challenges ahead, Knowledge and Information Systems 57 (2018).

[6] H. A. López, T. T. Hildebrandt, Three decades of formal methods in business process compliance: A systematic literature review, 2024. URL: https://arxiv.org/abs/2410.10906. arXiv:2410.10906.

[7] H. A. López, S. Debois, T. Slaats, T. T. Hildebrandt, Business process compliance using reference models of law, in: Lecture Notes in Computer Science, volume 12076 LNCS, 2020, pp. 378–399.

[8] A. Burattin, A. Gianola, H. A. López, M. Montali, Exploring the conformance space (extended abstract), in: CEUR Workshop Proceedings, volume 2952, 2021, pp. 62–67.

[9] J. Caballero-Villalobos, A. Burattin, H. A. López, High-level requirements-driven business process compliance, in: Proceedings of the 23rd International Conference on Business Process Management (BPM 2025), Springer, 2025.