Vacuity Matters: Activation-Aware DECLARE Conformance Checking of Judicial Event Logs

Roberto Nai[1,*], Emilio Sulis[1], Vittoria Margherita Sofia Trifiletti[2] and Alessandro Gianola[3]

[1]Computer Science Department, University of Turin, Corso Svizzera 185, 10149, Torino (IT)
[2]Department of Law, University of Turin, Lungo Dora Siena 100, 10153, Torino (IT)
[3]INESC-ID/Instituto Superior Técnico, Universidade de Lisboa, Rua Alves Redol 9, Lisbon (PT)

Abstract

Declarative conformance checking is well suited to judicial processes because it can represent procedural obligations without assuming a single predefined execution path. However, vacuous satisfaction may influence its results, whereby a constraint is classified as satisfied because its activation event does not occur. This aspect is relevant to heterogeneous judicial proceedings, in which constraints may apply only to subsets of cases and public event logs may provide a partial representation of the underlying process. This paper examines the role of constraint activation in DECLARE conformance checking through a case study based on proceedings of the Court of Justice of the European Union. We distinguish between satisfied and violated trace–constraint pairs, as well as inactive pairs, namely cases in which a constraint is not evaluated in practice because its activation event does not occur. The analysis considers activation coverage, conditional conformance among activated cases, and variation in aggregate conformance under alternative treatments of inactive pairs. The findings indicate that the resulting conformance level may vary depending on whether inactive pairs are considered satisfied, considered non-satisfied, or excluded from the calculation. These results suggest that aggregate scores can be usefully complemented by information about constraint activation and conditional conformance, particularly when analysing heterogeneous or partially observable legal processes.

Keywords

Process Mining, Declarative Conformance Checking, DECLARE, Vacuous Satisfaction, Judicial Processes

1. Introduction

The increasing digitalisation of judicial systems and public administrations is creating new opportunities to analyse legal procedures using data-driven methods [1]. The growing availability of digital legal data has fostered the development of computational methods for analysing legal documents, supporting legal decision-making, and assisting legal practitioners in a variety of analytical tasks [2]. While Artificial Intelligence (AI) methods may enhance access to justice, reduce costs, save time, and potentially improve the quality and consistency of judicial services [3], translating high-level digital governance strategies into effective mechanisms for monitoring procedural execution and compliance remains an open challenge [4]. In particular, there is an increasing need for approaches that can systematically assess whether the execution of legal procedures conforms to the requirements established by regulations and procedural guidelines.

From a process modelling perspective, legal procedures can be represented through both imperative and declarative process models. Imperative models explicitly define the admissible execution flows, whereas declarative models specify the behavioural constraints that executions must satisfy without prescribing a single execution path. Because of their flexibility, declarative models are particularly suitable for representing highly variable, knowledge-intensive processes, such as those commonly found in legal and administrative domains [5]. Among declarative process modelling languages, DECLARE

The 25th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2026) - Satellite Events (Workshops, Tutorials, Demos, Posters)
*Corresponding author.
roberto.nai@unito.it (R. Nai); emilio.sulis@unito.it (E. Sulis); vittoriamargheritasofia.trifiletti@unito.it (V. M. S. Trifiletti); alessandro.gianola@tecnico.ulisboa.pt (A. Gianola)
0000-0003-4031-5376 (R. Nai); 0000-0003-1746-3733 (E. Sulis); 0009–0000–3155–758X (V. M. S. Trifiletti); 0000-0003-4216-5199 (A. Gianola)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-26

CEUR-WS.org/Vol-4340/paper17.pdf

has become the de facto standard, providing a rich collection of constraint templates widely used to model procedural requirements and perform declarative conformance checking [6].

In this paper, we investigate how the absence of observable constraint activations affects the interpretation of declarative conformance results in heterogeneous judicial proceedings. Rather than relying exclusively on aggregate conformance scores, we propose an activation-aware analysis that distinguishes inactive, satisfied, and violated trace–constraint pairs, thereby separating the absence of evaluable evidence from observed conformance or violation. This distinction provides a more informative interpretation of declarative conformance results, particularly in legal domains where procedural requirements may apply only to subsets of cases and event logs often provide only a partial representation of the underlying process.

To this end, we combine event-log reconstruction, domain-expert validation, declarative formalisation of procedural requirements, and activation-aware conformance analysis within a unified methodological framework. The proposed analysis integrates three complementary perspectives: (i) observable activation coverage, which quantifies the availability of evidence for evaluating each requirement; (ii) conditional conformance, which assesses compliance only among cases in which the corresponding requirements are activated; and (iii) the sensitivity of aggregate conformance results to alternative treatments of inactive trace–constraint pairs. Together, these perspectives provide a more transparent interpretation of declarative conformance outcomes and support a more informed assessment of procedural compliance in heterogeneous judicial processes.

Based on these considerations, this study examines the role of constraint activation in interpreting DECLARE conformance results. In particular, it investigates how frequently constraints are activated, how often they are satisfied when active, and how different treatments of inactive trace–constraint pairs influence the aggregate conformance score. Accordingly, we formulate the following research questions:

RQ1. To what extent does the event log provide observable activation evidence for each process requirement?

RQ2. What conformance patterns are observed among traces in which the corresponding process requirements are activated?

RQ3. How sensitive are aggregate conformance results to alternative treatments of trace–requirement pairs without observed activations?

Together, these questions structure the analysis by separating the availability of activation evidence from conformance among activated cases and from the aggregation choices applied to inactive cases.

Given the complexity and domain specificity of legal procedures, human expertise remains essential throughout the knowledge acquisition and formalisation phases. In particular, we adopt a Human-in-the-Loop approach by involving a legal domain expert in different stages of the methodology to validate the extracted procedural knowledge, resolve ambiguities, and interpret the conformance results [7].

The remainder of this paper is organised as follows. Section 2 reviews the relevant literature. Section 3 presents the methodology, Section 4 describes the case study, and Section 5 reports the results, while Section 6 discusses the findings. Finally, Section 7 concludes the paper and outlines future work.

2. Background & Related Work

The growing availability of digital legal data has fostered the application of Artificial Intelligence (AI) and Machine Learning (ML) techniques to legal analytics and decision support. ML has been successfully applied to predicting judicial decisions, including judgments of the European Court of Human Rights [8]. A recent work identified several open challenges for AI-supported judicial process analysis, including information extraction from heterogeneous legal data, process modelling, real-time monitoring, conformance checking, predictive analysis, workload assessment, and domain-specific AI [9].

The analysis and modelling of legal procedural workflows are closely related to Business Process Management (BPM), which provides conceptual and methodological foundations for modelling, analysing, and improving organisational processes [10]. Within this field, Process Mining (PM) extracts process knowledge from event data recorded by information systems, supporting process discovery, performance analysis, conformance checking, and operational monitoring [11].

The effectiveness of PM techniques depends on the availability of suitable event logs that describe process-instance execution through timestamped events. While many business domains meet this requirement, legal processes often present additional challenges. Judicial and administrative procedures are frequently documented through heterogeneous information sources, and complete event logs are not always available [12]. Furthermore, much procedural knowledge is embedded in legal texts, regulations, judicial decisions, and administrative documents, requiring information extraction techniques to reconstruct process-related information [13].

Public procurement provides a related application domain for applying AI techniques to legal and administrative data [14, 15]. Related work has also addressed integrating heterogeneous information sources and constructing structured procurement datasets [16, 17]. Data-driven analyses of public procurement have further combined process-oriented and machine-learning approaches [18, 19]. Process mining and event-log enrichment have been applied to European public procurement workflows to reconstruct and analyse process behaviour [20, 21]. More recent research has explored LLM-based event-log augmentation and legal process mining for procurement workflows [22, 23]. These developments are part of broader research at the intersection of legal informatics and automated process analysis [24].

Early work focused on formal representations of legal procedures, aiming to capture procedural rules, normative constraints, and admissible sequences of legal actions through logic-based models [25, 26]. More recent studies have examined how to extract procedural and regulatory knowledge from textual sources and transform it into formal process representations. This includes deriving declarative process models from natural-language descriptions and identifying constraints and their relations in regulatory documents through natural language processing techniques [27, 28]. Researchers have also investigated LLM-based approaches to extract and evaluate structured procedural knowledge from textual sources [29, 30].

A complementary line of research has compared textual or formal process specifications with observed behaviour. Existing approaches include compliance checking against natural-language specifications using behavioural spaces and the semantic detection of non-compliances with declarative process constraints [31, 32].

From a modelling perspective, judicial procedures vary considerably and include procedural obligations that often depend on the specific circumstances of a case. These features make declarative process models particularly attractive, as they specify behavioural constraints rather than prescribing a single execution flow. Among declarative languages, DECLARE has become one of the most widely adopted formalisms for representing flexible processes and provides a natural foundation for conformance checking in domains governed by procedural rules and legal obligations [6, 11].

Recent research has further extended declarative conformance checking by supporting richer data-aware DECLARE specifications, enabling the efficient evaluation of constraints that combine control-flow and data conditions. These advances considerably broaden the applicability of declarative conformance checking to complex real-world processes, including domains characterised by rich contextual information and heterogeneous execution scenarios [33].

A relevant issue in declarative conformance checking is vacuous satisfaction. Many DECLARE constraints are evaluated only when a specific activation occurs; when it does not occur, a trace may formally satisfy the constraint even though no corresponding behaviour has been observed [34]. Earlier work on a posteriori analysis of declarative processes introduced activation-, fulfilment-, and violation-based information for analysing the behaviour of DECLARE constraints [35]. Previous research has therefore investigated constraint activation to distinguish vacuous satisfaction from behaviour that provides relevant evidence for evaluating a constraint [36, 37]. This distinction matters especially for heterogeneous or partially observable event logs, where the absence of an activation may result from either the non-applicability of a requirement or incomplete information about the underlying

process [38].

Building on this literature, our study focuses on the effect of inactive trace–constraint pairs on calculating and interpreting aggregate conformance scores in an event log reconstructed from real-world judicial proceedings. It complements existing approaches by separating activation coverage from conditional conformance and comparing alternative treatments of inactive pairs in aggregate calculations. Rather than introducing new semantics for constraint activation, this work is methodological: it integrates these established concepts into a unified, activation-aware analysis of aggregate DECLARE conformance, jointly considering activation evidence, conformance among activated cases, and aggregation choices to make the effect of inactive pairs explicit.

3. Methodology

We conducted a descriptive conformance analysis to examine how the uneven presence of relevant events across traces affects interpreting conformance results. The proposed analysis addresses this issue in three steps: it measures the observable activation coverage for each process requirement, evaluates conformance only where an activation is observed, and compares alternative treatments of trace–requirement pairs that lack activations.

The approach was operationalised using a DECLARE model applied to the event log under consideration. The unit of analysis is the *trace–constraint pair*, namely the evaluation of a single DECLARE constraint against a single trace. We evaluated each trace in the event log against every constraint included in the model. For each trace–constraint pair, we recorded the number of activations (i.e., occurrences of the activity that triggers the evaluation of the constraint), fulfilments (i.e., activations for which the required condition was satisfied), and violations (i.e., activations for which the required condition was not satisfied) [35]. We then combined these activation-level results into a single evaluation outcome for each trace–constraint pair. Specifically, each pair was assigned to one of *three* categories:

1. *inactive*, if the trace contains no activation of the constraint;
2. *satisfied*, if the constraint is activated and none of its activations is violated;
3. *violated*, if the constraint is activated and at least one activation is violated.

An inactive pair is therefore distinct from both a satisfied and a violated active pair. It indicates that the event required to activate the constraint does not occur in the trace and that the constraint is not evaluated in practice for that case [36, 37].

Figure 1 provides an overview of the analysis pipeline, from evaluating trace–constraint pairs to the three complementary perspectives used to address the RQs.

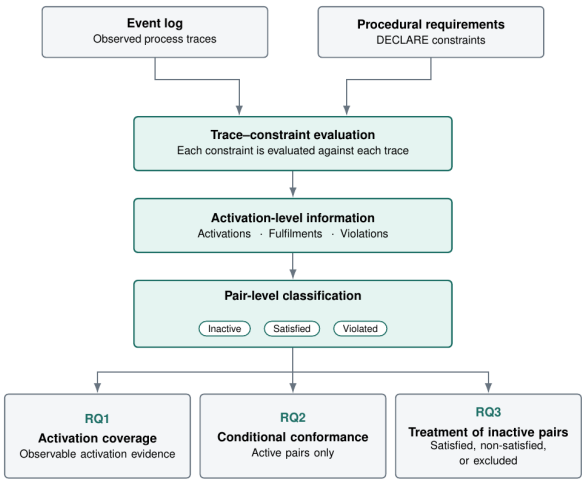


Figure 1: Overview of the activation-aware conformance analysis methodology

3.1. Constraint Activation Coverage

The *Constraint Activation Coverage* examines how frequently each DECLARE constraint is activated in the event log. A constraint is active in a trace if it activates at least once. For each constraint, we counted the number of traces containing at least one activation. We then calculated activation coverage as the proportion of activated traces over the total number of traces in the event log. Conversely, the inactivity rate represents the proportion of traces in which the constraint is never activated.

We report activation coverage separately from conformance because we cannot empirically evaluate a constraint when its activation event does not occur. Therefore, the absence of an activation does not provide direct evidence about whether the constraint would have been satisfied or violated.

We also report the total number of activations for each constraint. This value may differ from the number of activated traces because the same constraint can be activated more than once within a single trace.

3.2. Conditional Conformance

The *Conditional Conformance* evaluates each constraint only among the traces in which it is activated. Inactive trace–constraint pairs are excluded from the calculation, so the resulting rates refer exclusively to activated traces.

For each constraint, we divided activated traces into satisfied and violated cases. We calculated the conditional satisfaction rate by dividing the number of satisfied cases by the total number of activated cases. Similarly, we calculated the conditional violation rate as the proportion of violated cases among all activated cases. The two rates are complementary [34]: for each constraint, every activated trace is classified as either satisfied or violated. This approach separates how frequently a constraint is activated in the observed log from how frequently it is satisfied when active.

The classification is performed at the trace–constraint level. When a constraint is activated multiple times within the same trace, the pair is classified as violated if at least one activation is violated. We retain counts of individual fulfilments and violations as diagnostic information but do not treat them as independent observations when calculating the conditional rates.

3.3. Treatment of Inactive Pairs

The *Treatment of Inactive Pairs* examines how the aggregate conformance score varies under three alternative aggregation approaches. All three approaches use the same activation, fulfilment, and violation results; only the contribution assigned to inactive pairs changes. Differences among the resulting scores therefore reflect the adopted aggregation approach. Let P be the total number of trace–constraint pairs, A the number of active pairs, I the number of inactive pairs, and S the number of satisfied active pairs.

Inactive pairs considered satisfied. Under the vacuous-satisfaction convention, both satisfied active pairs and inactive pairs contribute positively to the aggregate score. The score therefore represents the proportion of these pairs, $S + I$, over the complete set of trace–constraint pairs, P:

$$C_{\text{vacuous}} = \frac{S + I}{P} \tag{1}$$

Inactive pairs considered non-satisfied. In the second approach, we include all trace–constraint pairs in the aggregate calculation, while only satisfied active pairs contribute positively to the score. The score therefore represents the proportion of satisfied active pairs, S, over the complete set of pairs, P:

$$C_{\text{non-vacuous}} = \frac{S}{P} \tag{2}$$

Under this approach, inactive pairs receive a value of zero for aggregation purposes but are not interpreted as observed violations.

Inactive pairs excluded. The third approach excludes inactive pairs and calculates conformance only over the pairs in which the constraint is activated:

$$C_{\text{activation-aware}} = \frac{S}{A} \tag{3}$$

This measure represents conformance conditional on constraint activation. Comparing the three approaches assesses how treating inactive pairs influences the aggregate result. The scores are micro-averages, meaning each trace–constraint pair contributes equally to the calculation.

4. Case Study

The selected application domain concerns judicial proceedings of the *Court of Justice of the European Union* (CJEU), in which procedural norms govern the ordering and execution of judicial acts. This context provides a representative scenario for studying how procedural knowledge expressed in regulatory text can be acquired, formalised into executable specifications, and evaluated against observed workflows.

Judicial proceedings are publicly accessible through the InfoCuria repository[1], which provides structured information on judicial cases, procedural developments, and official documents. Regulatory provisions governing these proceedings are publicly available through the CJEU website[2]. The availability of both procedural traces and regulatory sources makes this context particularly suitable for evaluating compliance specifications derived from executable procedural knowledge against real-world observations. We collected cases formally closed between September 2017 and September 2025 to maximise observed lifecycles. We refer to the event log reconstructed from these data as the *CURIA* event log.

Data representation. The dataset combines case-level metadata and event-level procedural records. Case attributes describe the proceeding as a whole, including the case identifier, parties, legal domain, country, and judicial body, and distinguish between General Court ("T") and Court of Justice ("C") cases.

Event-level records describe dated procedural activities that reflect each case's judicial lifecycle. In particular, InfoCuria records the lodging of an *application*, which initiates judicial proceedings; requests for a *preliminary ruling*, through which national courts refer questions concerning the interpretation or validity of EU law; and the submission of *statements of case* or *written observations* by parties and institutions. The recorded activities also include Advocate General *opinions*, which provide non-binding legal assessments to assist the Court, as well as *judgments*, *orders*, and their publication on the InfoCuria website. Each activity includes a timestamp, enabling chronological ordering of events and reconstruction of the temporal evolution of individual proceedings. Together with the case-level metadata, these records provide the contextual and temporal information required to construct process traces and perform compliance analysis. Further details on the dataset construction and its transformation into an event log are provided in [39].

Procedural Requirements and DECLARE Model. We instantiated the general analysis described in Section 3 using five procedural requirements validated by a domain expert, namely a legal scholar with expertise in EU procedural law. We formalised each *procedural requirement* (rule) as a corresponding *DECLARE constraint*.

The selected constraints provide an illustrative set covering different types of procedural relations, including precedence, response, and prohibition, while remaining directly observable in the reconstructed event log; they represent a focused subset of the broader procedural requirements governing CJEU proceedings.

[1] InfoCuria available at https://infocuria.curia.europa.eu/tabs/tout?lang=EN
[2] Regulatory provisions available at https://eur-lex.europa.eu/eli/proc_rules/2024/2173/oj

Table 1 reports their natural-language formulation and corresponding DECLARE representation. These constraints constitute the evaluation model applied to all traces in the CURIA event log. The activity labels used in the formalisation correspond directly to those recorded in the event log. Further details on the identification, expert validation, and DECLARE formalisation of these requirements are provided in [39].

Table 1
Expert-validated procedural requirements and their DECLARE formalisation.

Rule ID	Procedural requirement	DECLARE constraint
R1	A judgment must be preceded by an opinion	`Precedence[Opinion, Judgment]`
R2	A judgment must be followed by its official publication	`Response[Judgment, Judgment (OJ)]`
R3	An order must be followed by its official publication	`Response[Order, Order (OJ)]`
R4	A judgment must be preceded by a statement of case or written observations	`Precedence[Observations, Judgment]`
R5	No statement of case or written observations may occur after a judgment	`NotResponse[Judgment, Observations]`

We evaluate the constraints against each recorded occurrence of *judgment*; multiple occurrences are rare in the event log, with 706 *judgment* events observed across 703 traces.

5. Results

Implementation Details. The analysis was implemented in Python 3.12. We processed the CURIA data with *pandas* and used it to construct the event log, which we represented and managed using *PM4Py* [40]. We evaluated the DECLARE constraints using *Declare4Py* [41]. The complete experimental implementation and the materials required to reproduce the analysis are publicly available in the accompanying repository: https://github.com/roberto-nai/KM4LAW-2026.

Event log. The event log contains 1,284 cases, 4,744 events, 19 activities, and 187 variants; the median and mean case durations were 16.6 and 24.6 months, respectively. Figure 2 provides directly-follows graph (DFG) [11] representations of the reconstructed event log. A domain expert confirmed that the main directly-follows relations matched institutional practice. Table 2 shows an excerpt of the reconstructed event log for an illustrative judicial proceeding, showing how case metadata and timestamped procedural activities combine to form executable process traces. The selected trace comprises a request for a preliminary ruling, the publication of the application in the Official Journal, two written-observation events, an Advocate General opinion, the final judgment, and its subsequent publication in the Official Journal. The resulting event log and the validation of its reconstruction are described more extensively in [39].

Table 2
Excerpt of the reconstructed event log for case C-745/23. The omitted metadata include parties, area of law, ECLI, EUR-Lex reference, and country.

Case ID	Activity	Timestamp	Metadata	Category
C-745/23	Request for a preliminary ruling	2023-12-05	…	C
C-745/23	Application (OJ)	2024-02-02	…	C
C-745/23	Statement of case or written observations	2024-03-25	…	C
C-745/23	…	…	…	…

5.1. Constraint Activation Coverage

Table 3 reports each constraint's activation coverage. R1, R2, R4, and R5 share *Judgment* as their activation activity and were therefore activated in the same 703 traces, corresponding to a coverage of

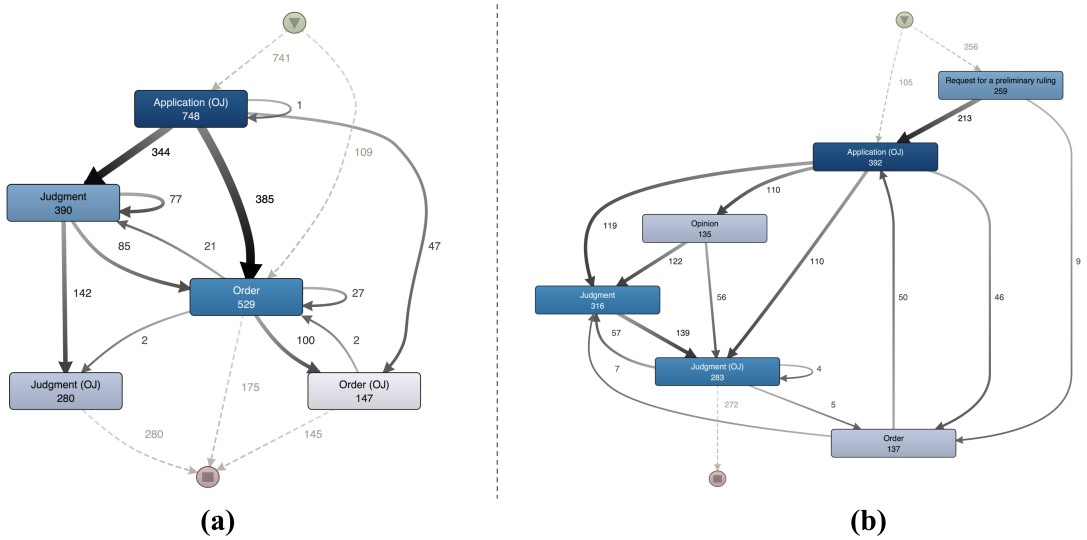


Figure 2: DFG representations of judicial workflows: (a) cases handled by the General Court; (b) the Court of Justice, showing the most frequent control-flow relations.

54.75%. For each constraint, 581 trace–constraint pairs were inactive. R3, whose activation activity is *Order*, was activated in 624 traces, corresponding to 48.60% coverage. The remaining 660 pairs were inactive. R3 therefore presents a slightly lower activation coverage than the other constraints. Across all constraints, the total number of activations exceeded the number of activated traces. In particular, we identified 706 occurrences of *Judgment* across 703 traces, and 666 occurrences of *Order* across 624 traces. This indicates that some traces contain multiple activations of the same constraint.

Table 3
Activation coverage by DECLARE constraint.

Rule ID	Activation	Active cases	Coverage	Inactive cases	Activations
R1	Judgment	703	54.75%	581	706
R2	Judgment	703	54.75%	581	706
R3	Order	624	48.60%	660	666
R4	Judgment	703	54.75%	581	706
R5	Judgment	703	54.75%	581	706

Overall, the Constraint Activation Coverage shows that observable activation evidence was available for approximately half of the traces for each constraint. Coverage ranged from 48.60% for R3 to 54.75% for the other constraints, indicating relatively limited variation between requirements activated by *Order* and those activated by *Judgment*. Nevertheless, even for the constraints with the highest coverage, a substantial proportion of traces contained no observed activation. These inactive trace–constraint pairs are therefore analytically distinct from cases in which the corresponding requirement was activated and could be evaluated as satisfied or violated.

5.2. Conditional Conformance

Table 4 reports the Conditional Conformance results computed exclusively over activated trace–constraint pairs, highlighting substantial variation in satisfaction outcomes across the five analysed constraints. Unlike traditional conformance metrics, which also include traces in which a constraint is never activated, Conditional Conformance focuses only on the subset of traces where compliance can actually be assessed. As a result, the reported satisfaction rates more directly indicate how consistently each constraint is respected once its activation condition is met.

Among the analysed constraints, R5 was satisfied in all 703 traces in which it was activated, yielding a conditional satisfaction rate of 100%. R2 was satisfied in 561 of its 703 activated traces, corresponding

to a conditional satisfaction rate of 79.80%, while the remaining 20.20% of activated traces resulted in violations. In contrast, R3 was satisfied in only 168 of its 624 activated traces, leading to a conditional satisfaction rate of 26.92%. R1 and R4 showed even lower conditional satisfaction rates, at 18.35% and 15.65%, respectively. Overall, the results reveal considerable differences in the degree of compliance exhibited by the analysed constraints. While R5 was consistently satisfied when activated, R4 had the lowest conditional satisfaction rate, indicating it was violated in most traces in which it was activated.

Table 4
Conditional conformance among activated cases.

Rule ID	Active	Satisfied	Satisfaction	Violated	Violation
R1	703	129	18.35%	574	81.65%
R2	703	561	79.80%	142	20.20%
R3	624	168	26.92%	456	73.08%
R4	703	110	15.65%	593	84.35%
R5	703	703	100.00%	0	0.00%

Figure 3 combines Constraint Activation Coverage and Conditional Conformance by representing inactive, satisfied, and violated pairs as proportions of all 1,284 traces. The identical inactive proportions for R1, R2, R4, and R5 reflect their common activation activity, *Judgment*, whereas R3 has a larger inactive portion because its activation activity, *Order*, occurred in fewer traces. Despite sharing the same activation coverage, R1, R2, R4, and R5 exhibit markedly different outcomes among their activated pairs: all active pairs satisfied R5, a majority satisfied R2, while violations predominated for R1 and R4. Violations also predominated for R3. The comparison therefore shows that similar activation coverage does not necessarily correspond to similar conditional conformance. Overall, satisfaction rates ranged from 15.65% for R4 to 100.00% for R5 when considering only activated pairs.

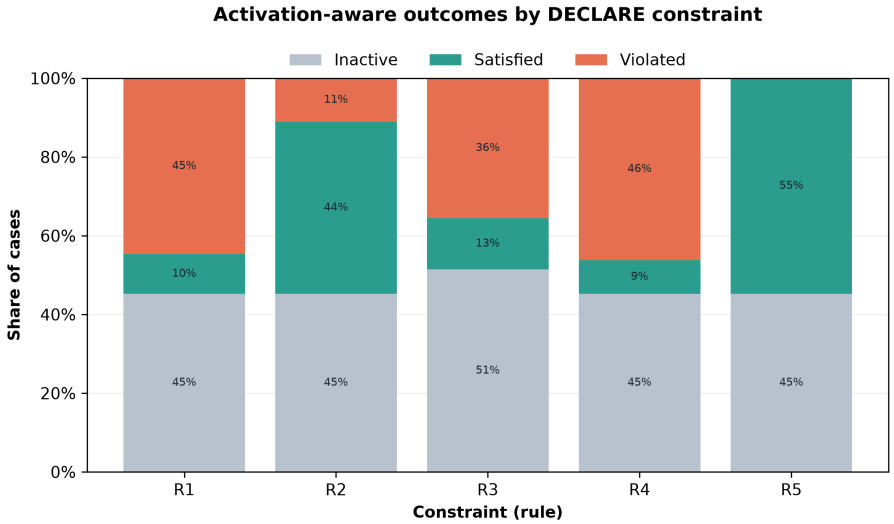


Figure 3: Distribution of inactive, satisfied, and violated trace–constraint pairs. Percentages are calculated over all traces for each constraint

5.3. Treatment of Inactive Pairs

To examine how treating inactive pairs affects aggregate conformance, we evaluated each of the 1,284 traces against 5 DECLARE constraints, resulting in $P = 6{,}420$ trace–constraint pairs. Following the notation introduced in Section 3.3, $A = 3{,}436$ pairs were active and $I = 2{,}984$ were inactive, corresponding to 53.52% and 46.48% of all pairs, respectively. Among the active pairs, $S = 1{,}671$ were satisfied, while the remaining 1,765 were violated. Based on these counts, Table 5 compares the aggregate conformance scores obtained using the three calculations defined in Equations 1–3.

When we treated inactive pairs as satisfied, both the 1,671 satisfied active pairs and the 2,984 inactive pairs contributed positively to the score. Under the vacuous-satisfaction treatment defined in Eq. (1), 4,655 of the 6,420 trace–constraint pairs were scored as satisfied, producing an aggregate conformance score of 72.51%. When inactive pairs were considered non-satisfied, they remained part of the aggregate calculation but did not contribute positively to the score. Under the non-vacuous treatment defined in Eq. (2), only the 1,671 satisfied active pairs were scored as satisfied out of all 6,420 pairs, resulting in a score of 26.03%. Inactive pairs were assigned a non-satisfied state for aggregation purposes but were not interpreted as observed violations. Finally, after excluding inactive pairs, we restricted the calculation to the 3,436 active pairs. Under the activation-aware calculation defined in Eq. (3), 1,671 of these pairs were satisfied, yielding a score of 48.63%. This value expresses the proportion of satisfied pairs among those in which the corresponding constraint was activated.

Table 5
Aggregate conformance under alternative treatments of inactive pairs.

Treatment of inactive pairs	Pairs scored as satisfied	Pairs considered	Score
Considered satisfied (C_{vacuous})	4,655	6,420	72.51%
Considered non-satisfied ($C_{\text{non-vacuous}}$)	1,671	6,420	26.03%
Excluded ($C_{\text{activation-aware}}$)	1,671	3,436	48.63%

The difference between the scores for satisfied and unsatisfied inactive pairs was 46.48 percentage points. This difference is exactly equal to the proportion of inactive pairs in the complete set of trace–constraint pairs:

$$\frac{I}{P} = \frac{2{,}984}{6{,}420} = 46.48\%. \tag{4}$$

The Treatment of inactive pairs therefore shows that the aggregate score varies according to the treatment assigned to inactive pairs. The activation-aware calculation, which excludes these pairs, reports the satisfaction rate among the pairs in which the corresponding constraints were activated.

6. Discussion

The proposed activation-aware analysis suggests that declarative conformance checking can provide a useful perspective for studying judicial procedures, particularly when event logs represent heterogeneous proceedings or only partially observable processes. Rather than relying exclusively on aggregate conformance values, distinguishing between inactive, satisfied, and violated trace–constraint pairs provides additional information about the extent to which the available evidence can actually evaluate procedural requirements.

From a legal perspective, this distinction is relevant because procedural requirements are not necessarily applicable to every judicial proceeding. Many procedural obligations take effect only after specific procedural events occur, and different categories of cases may legitimately follow different procedural paths. For instance, delivering an Advocate General's opinion or submitting specific observations is legally mandated only for certain types of proceedings. Consequently, the absence of a constraint activation should not automatically be interpreted as evidence of compliance or non-compliance, but should instead be considered in light of the procedural context.

Beyond methodological considerations, activation-aware conformance analysis may contribute to the broader objective of monitoring the lawful execution of judicial procedures. By formalising procedural requirements and evaluating their occurrence in observed proceedings, the approach can help identify situations in which further legal scrutiny may be warranted. In this respect, process mining can provide an additional layer of evidence for procedural oversight, while legal interpretation remains necessary to determine the significance of observed deviations.

The proposed analysis also improves the interpretability of declarative conformance results. Reporting activation coverage together with conditional conformance allows researchers and legal experts to

distinguish between the applicability of procedural requirements and their observed fulfilment. This additional information may support a more transparent interpretation of compliance analyses, especially when event logs are reconstructed from publicly available judicial data rather than directly extracted from operational information systems.

From an operational perspective, the three aggregate measures serve different interpretative purposes. C_{vacuous} reflects the standard vacuous-satisfaction convention, whereas $C_{\text{activation-aware}}$ focuses on conformance only where activation evidence is observable. Conversely, $C_{\text{non-vacuous}}$ provides a conservative view in which the absence of activation does not contribute positively to the aggregate score. As an informal guideline for legal analysts, C_{vacuous} can be read as the proportion of case–requirement pairs in which no violation is observed, including cases where the requirement is never activated; $C_{\text{activation-aware}}$ indicates how often the requirement is satisfied among cases in which its activation is actually observed; and $C_{\text{non-vacuous}}$ indicates how much positive evidence of conformance is observed across the entire set of cases. Reporting these measures alongside activation coverage therefore supports selecting and interpreting the measure that best fits the monitoring objective.

At the same time, several limitations warrant acknowledgement. First, the study considers a limited number of expert-validated procedural requirements and a single judicial context. Secondly, inactive constraints may arise for different reasons, including the genuine non-applicability of a procedural rule, incomplete event logs, or limitations in the event log reconstruction process. Declarative conformance checking alone cannot distinguish among these situations, making legal expertise essential for interpreting the results. A case-level qualitative analysis would be required to determine which explanation applies to individual inactive pairs, combining procedural context with legal expert assessment. Finally, the proposed indicators should be regarded as complementary to existing conformance measures rather than as replacements for established declarative conformance techniques.

Overall, the findings indicate that activation-aware conformance analysis represents a useful complementary perspective for analysing legal procedural workflows. Combining process mining techniques with legal expertise may facilitate a more nuanced interpretation of procedural compliance, particularly in judicial domains characterised by procedural variability and incomplete observational data.

7. Conclusion and Future Work

This study examined how the presence or absence of observable activation evidence affects interpreting declarative conformance results in judicial event logs. To address this issue, we proposed an activation-aware conformance analysis that explicitly distinguishes inactive, satisfied, and violated trace–constraint pairs, thereby complementing traditional aggregate conformance measures with information about activation coverage and conditional conformance.

Regarding RQ1, the activation coverage analysis allowed us to determine the extent to which the event log contains sufficient evidence to evaluate each procedural requirement. This distinction matters because the absence of an activation should not be interpreted as evidence of either compliance or violation. Instead, it identifies cases in which the available event data do not provide the conditions necessary to meaningfully assess whether a requirement has been satisfied or violated.

For RQ2, Conditional Conformance provided an assessment restricted to trace–constraint pairs in which the corresponding procedural requirement was activated. By excluding non-activated cases from the computation, this metric isolates observed fulfilment or violation from the mere absence of the events needed to evaluate it. Consequently, it enables a more informative and interpretable comparison of compliance levels across different requirements.

Finally, the analysis addressing RQ3 showed that aggregate conformance is strongly influenced by the way trace–constraint pairs without observed activations are handled. As a consequence, an aggregate conformance score should not be interpreted in isolation, since its meaning depends on how inactive cases are treated. Activation coverage and Conditional Conformance therefore provide complementary perspectives that are essential for understanding the significance of the aggregate score and for distinguishing genuine compliance behaviour from a lack of observable evidence.

Future work will investigate how to distinguish non-applicable requirements from missing or unobservable events, building on previous approaches to compliance checking over incomplete event logs [38]. Another important direction is validating inactive and violated cases with legal experts and by analysing contextual case information. The approach will also be extended to additional legal event logs and broader sets of expert-validated procedural requirements.

Future work will extend this evaluation to larger and richer sets of procedural requirements, including data-aware DECLARE constraints incorporating case metadata and procedural time conditions, to assess the applicability of the activation-aware analysis to more expressive compliance specifications.

Acknowledgments

The research work in this article was partially conducted as part of the following initiatives: PiemontAIs - PR FESR 2021/2027, Grant Agreement No. 187173; AI4Prisma - PR FESR 21/27 SWIch - Regione Piemonte - CUP D19J25001150006.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly and ChatGPT for grammar and spelling checks. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] R. Nai, E. Sulis, S. Montanelli, L. Genga, H. A. López, Advancing legal systems through compliance, processes, and ai technologies, Computer Law & Security Review 61 (2026) 106338. doi:10.1016/j.clsr.2026.106338.

[2] K. D. Ashley, Artificial intelligence and legal analytics: new tools for law practice in the digital age, Cambridge University Press, 2017.

[3] M. Zalnieriute, A. Limantė (Eds.), The Cambridge Handbook of AI and Technologies in Courts, Cambridge Law Handbooks, Cambridge University Press, 2026. doi:10.1017/9781009744225.

[4] OECD, Digital Government Outlook 2026: From Foundations to Transformational Impact, OECD Publishing, 2026. doi:10.1787/0496b2bc-en.

[5] C. Di Ciccio, F. M. Maggi, J. Mendling, W. M. P. van der Aalst, Declarative process modeling in BPM: A systematic literature review, Software and Systems Modeling 17 (2018) 403–432. doi:10.1007/s10270-016-0537-0.

[6] M. Pesic, H. Schonenberg, W. M. P. van der Aalst, DECLARE: Full support for loosely-structured processes, in: Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC), IEEE Computer Society, 2007, pp. 287–300. doi:10.1109/EDOC.2007.14.

[7] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, L. He, A survey of human-in-the-loop for machine learning, Future Generation Computer Systems 135 (2022) 364–381. doi:10.1016/j.future.2022.05.014.

[8] M. Medvedeva, M. Vols, M. Wieling, Using machine learning to predict decisions of the European Court of Human Rights, Artificial Intelligence and Law 28 (2020) 237–266. doi:10.1007/s10506-019-09255-y.

[9] D. Bianchini, C. Bono, A. Campi, C. Cappiello, S. Ceri, F. De Luzi, M. Mecella, B. Pernici, P. Plebani, Challenges in AI-supported process analysis in the Italian judicial system: What after digitalization?, Digital Government: Research and Practice 5 (2024) 1–10.

[10] M. Dumas, M. L. Rosa, J. Mendling, H. A. Reijers, Fundamentals of Business Process Management, Springer, 2018. doi:10.1007/978-3-662-56509-4.

[11] W. M. P. van der Aalst, Process Mining: Data Science in Action, 2nd ed., Springer, 2016. doi:10.1007/978-3-662-49851-4.

[12] F. König, A. Egger, W. Kratsch, M. Röglinger, N. Wördehoff, Unstructured data in process mining: A systematic literature review, ACM Transactions on Management Information Systems 16 (2025) 25:1–25:34. doi:10.1145/3727148.

[13] L. Martin, N. Whitehouse, S. Yiu, L. Catterson, R. Perera, Better call GPT, comparing Large Language Models against lawyers, arXiv preprint arXiv:2401.16212 (2024).

[14] R. Nai, I. Fatima, G. Morina, E. Sulis, L. Genga, R. Meo, P. Pasteris, AI applied to the analysis of the contracts of the italian public administrations, in: Proceedings of the Ital-IA 2023 Workshops, volume 3486 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 255–260. URL: https://ceur-ws.org/Vol-3486/100.pdf.

[15] R. Nai, E. Sulis, R. Meo, Public procurement fraud detection and artificial intelligence techniques: a literature review, in: Companion Proceedings of the 23rd International Conference on Knowledge Engineering and Knowledge Management (EKAW 2022), volume 3256 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 1–13. URL: https://ceur-ws.org/Vol-3256/km4law4.pdf.

[16] R. Nai, E. Sulis, P. Pasteris, M. Giunta, R. Meo, Exploitation and merge of information sources for public procurement improvement, in: Machine Learning and Principles and Practice of Knowledge Discovery in Databases, volume 1752 of *Communications in Computer and Information Science*, Springer, 2023, pp. 89–102. doi:10.1007/978-3-031-23618-1_6.

[17] R. Nai, E. Sulis, R. Meo, ITH: An open database on italian tenders 2016–2023, Scientific Data 11 (2024) 1452. doi:10.1038/s41597-024-04342-5.

[18] R. Nai, E. Sulis, R. Meo, F. Gorgerino, G. M. Racca, L. Genga, Process mining on a public procurement dataset: A case study, in: International Workshops of ECML PKDD 2023, Turin, Italy, September 18–22, 2023, Revised Selected Papers, Part I, volume 2133 of *Communications in Computer and Information Science*, Springer, 2025, pp. 477–492. doi:10.1007/978-3-031-74630-7_35.

[19] R. Nai, R. Meo, G. Morina, P. Pasteris, Public tenders, complaints, machine learning and recommender systems: A case study in public administration, Computer Law & Security Review 51 (2023) 105887. doi:10.1016/j.clsr.2023.105887.

[20] R. Nai, E. Sulis, L. Genga, Automated analysis with event log enrichment of the european public procurement processes, in: Advances in Conceptual Modeling, volume 14319 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 178–188. doi:10.1007/978-3-031-47112-4_17.

[21] R. Nai, E. Sulis, D. Audrito, V. M. S. Trifiletti, R. Meo, L. Genga, Leveraging process mining and event log enrichment in european public procurement analysis: A case study, Computer Law & Security Review 57 (2025) 106144. doi:10.1016/j.clsr.2025.106144.

[22] I. Spada, E. Sulis, D. Audrito, V. M. S. Trifiletti, R. Nai, Augmenting public procurement event logs with large language models: a legal process mining approach, in: Proceedings of the Joint Ontology Workshops (JOWO 2025), Catania, Italy, September 8–9, 2025, volume 4176 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025, pp. 1–10. URL: https://ceur-ws.org/Vol-4176/km4law-1.pdf.

[23] R. Nai, E. Sulis, D. Audrito, V. M. S. Trifiletti, L. D. Caro, L. Genga, Legal process mining for the analysis of public procurement workflows, Computer Law & Security Review 61 (2026) 106283. doi:10.1016/j.clsr.2026.106283.

[24] D. Audrito, L. D. Caro, L. Genga, R. Mignone, R. Nai, I. Spada, E. Sulis, V. M. S. Trifiletti, Standardization and harmonization of law through automated process analysis and similarity techniques, in: Proceedings of the 1st International Workshop on Processes, Laws and Compliance (PLC 2024), co-located with the 6th International Conference on Process Mining (ICPM 2024), volume 3850 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 34–45. URL: https://ceur-ws.org/Vol-3850/paper4.pdf.

[25] G. Governatori, A. Rotolo, G. Sartor, Logic and the law: Philosophical foundations, deontics, and defeasible reasoning, in: Handbook of Deontic Logic and Normative Systems, College Publications, 2021, pp. 655–760.

[26] G. Sartor, Legal Reasoning: A Cognitive Approach to the Law, volume 5 of *Law and Philosophy Library*, Springer, 2009. doi:10.1007/978-90-481-3146-4.

[27] H. van der Aa, C. Di Ciccio, H. Leopold, H. A. Reijers, Extracting declarative process models from natural language, in: Advanced Information Systems Engineering, Springer, 2019, pp. 365–382.

[28] K. Winter, S. Rinderle-Ma, Detecting constraints and their relations from regulatory documents using NLP techniques, in: OTM 2018 Conferences, Springer, 2018, pp. 261–278.

[29] V. A. Carriero, A. Azzini, I. Baroni, M. Scrocca, I. Celino, Human evaluation of procedural knowledge graph extraction from text with large language models, in: Knowledge Engineering and Knowledge Management, Lecture Notes in Computer Science, Springer, 2024, pp. 434–452. doi:10.1007/978-3-031-77792-9_26.

[30] I. Spada, R. Nai, D. Audrito, V. S. M. Trifiletti, E. Sulis, An expert-validated LLM framework for transforming legal procurement texts into actionable data, in: Legal Knowledge and Information Systems, volume 416 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2025, pp. 356–363. doi:10.3233/FAIA251610.

[31] H. van der Aa, H. Leopold, H. A. Reijers, Checking process compliance against natural language specifications using behavioral spaces, Information Systems 78 (2018) 83–95. doi:10.1016/j.is.2018.01.007.

[32] C. Di Ciccio, F. J. Ekaputra, A. Cecconi, A. Ekelhart, E. Kiesling, Finding non-compliances with declarative process constraints through semantic technologies, in: Information Systems Engineering in Responsible Information Systems, Springer, 2019, pp. 60–74.

[33] A. Gianola, C. Zerva, Compliance checking for public administration processes using retrieval-augmented generation in llms: Novel directions and challenges, in: OVERLAY 2025, Bologna, Italy, October 26, 2025, volume 4142 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025, pp. 199–208. URL: https://ceur-ws.org/Vol-4142/paper22.pdf.

[34] C. Di Ciccio, M. Montali, Declarative process specifications: Reasoning, discovery, monitoring, in: Process Mining Handbook, volume 448 of *Lecture Notes in Business Information Processing*, Springer, 2022, pp. 108–152. doi:10.1007/978-3-031-08848-3_4.

[35] A. Burattin, F. M. Maggi, W. M. P. van der Aalst, A. Sperduti, Techniques for a posteriori analysis of declarative processes, in: Proceedings of the 2012 IEEE 16th International Enterprise Distributed Object Computing Conference (EDOC), IEEE Computer Society, 2012, pp. 41–50. doi:10.1109/EDOC.2012.15.

[36] F. M. Maggi, M. Montali, C. Di Ciccio, J. Mendling, Semantical vacuity detection in declarative process mining, in: Business Process Management, volume 9850 of *Lecture Notes in Computer Science*, Springer, 2016, pp. 158–175. doi:10.1007/978-3-319-45348-4_10.

[37] C. Di Ciccio, F. M. Maggi, M. Montali, J. Mendling, On the relevance of a business constraint to an event log, Information Systems 78 (2018) 144–161. doi:10.1016/j.is.2018.01.011.

[38] F. Chesani, R. De Masellis, C. Di Francescomarino, C. Ghidini, P. Mello, M. Montali, S. Tessaris, Abducing compliance of incomplete event logs, in: AI*IA 2016 Advances in Artificial Intelligence, volume 10037 of *Lecture Notes in Artificial Intelligence*, Springer, 2016, pp. 208–222. doi:10.1007/978-3-319-49130-1_16.

[39] R. Nai, E. Sulis, V. Trifiletti, A. Gianola, G. Boella, Bridging regulatory text and declarative process models for procedural compliance, 2026. Manuscript under review.

[40] A. Berti, S. van Zelst, D. Schuster, PM4Py: A process mining library for Python, Software Impacts 17 (2023) 100556. doi:10.1016/j.simpa.2023.100556.

[41] I. Donadello, F. Riva, F. M. Maggi, A. Shikhizada, Declare4Py: A Python library for declarative process mining, in: BPM 2022 Demonstration and Resources Track, volume 3216 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 117–121. URL: https://ceur-ws.org/Vol-3216/paper_249.pdf.