# One Score Does Not Fit All: A Configurable BPMN Similarity Framework

Sherri Hadian[1], Adrian Rebmann[1], Robert Blümel[1,*] and Charlotte Balcke[1]

[1] SAP Signavio, Hasso-Plattner-Ring 7, 69190 Walldorf, Germany

**Abstract**

Organizations rely on growing collections of BPMN process models, and managing them increasingly depends on comparing models for similarity. What constitutes similarity, however, has no single answer: it can be assessed along several dimensions, quantified through different metrics, and weighted differently across tasks. Prior work has explored individual aspects and surveyed the field, but a tool that brings them together and leaves the central choices to the user is missing. We present a framework that combines multiple similarity perspectives and lets users configure both the metrics and the weights of the comparison, so the resulting score fits the task at hand.

**Keywords**

Evaluation framework, BPMN model similarity, Process model similarity, Configurable similarity assessment

## 1. Introduction

BPMN process models capture how organizations run. As model collections grow, analysts increasingly need to compare them, reconciling variants of the same process, reviewing whether a newly proposed model overlaps with an existing one, or validating a refactored model against its reference [1]. Such comparisons are inherently pairwise and inherently judgment-laden: what counts as similar depends on the task at hand, so analysts need a way to inspect a comparison closely and tune it to the question they are asking. Beyond their immediate use, well-grounded similarity scores can also serve downstream research, such as benchmarking machine learning models that learn process model representations.

Determining the right similarity score is not trivial, for three reasons. (1) Similarity between BPMN models can be measured along different dimensions, structural elements, modeled behavior, or semantic and syntactic content [2]. (2) Once one or more dimensions are chosen, there are still different ways to quantify similarity, for instance, through set-based metrics such as Jaccard or Dice. (3) Organizations and even analysts within them weigh these dimensions differently depending on the task and on which part of a BPMN model carries the most meaning for the comparison at hand. Prior work has studied similarity along individual dimensions [3, 4], combined dimensions [5], examined correlations between proposed measures [6], and surveyed the field [2, 7]. BPMNDiffViz [8] is the only publicly available tool for comparing BPMN models. It computes a structural graph edit distance between two models, a single metric over a single dimension, which relies on graph-matching whose cost grows steeply with model size. No tool currently lets analysts combine the structural, behavioral, and semantic dimensions in one comparison, choose among interpretable metrics, and weight components to fit the task at hand, while keeping the per-comparison cost low enough for interactive use.

We present a framework that addresses all three gaps. It first uses a semantic similarity-based normalization mechanism to align element labels across the two models, accounting for naming differences, and then compares the aligned models along both structural and behavioral dimensions. Analysts can select from a set of similarity metrics and adjust the weights of its underlying components, so that all three dimensions contribute to the resulting score in proportion to the task. The configured score can be applied to tasks such as variant analysis, candidate-vs-repository comparison, and benchmarking

*Proceedings of the BPM 2026 Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Forum (BPM 2026), Toronto, Canada, September 27th to October 2nd, 2026.*

*Corresponding author.

sherri.hadian@sap.com (S. Hadian); adrian.rebmann@sap.com (A. Rebmann); robert.bluemel@sap.com (R. Blümel); charlotte.balcke@sap.com (C. Balcke)

learned model representations. This work extends a previously introduced similarity evaluation component [9] into a standalone framework for configurable BPMN model comparison, while remaining extensible to additional ways of calculating similarity scores as evaluation needs evolve.

## 2. Framework Description

The framework takes two BPMN models, as standard BPMN 2.0 XML or as Signavio JSON, together with a configuration as input. The configuration specifies the similarity metric, a semantic-alignment threshold, and the relative weights of its components. The output is a similarity score, broken down by similarity dimension. This detailed analysis is presented in an interactive dashboard that lets the analyst inspect both diagrams side-by-side and recompute the score under different configurations.

### 2.1. Architecture and Implementation

The framework uses a modular architecture that makes it both extensible and usable through two interfaces. Each similarity metric is a separate component, allowing developers to add new semantic normalization approaches or structural distance measures without affecting the rest of the framework. It is implemented as a Python library and as an interactive dashboard in a reactive marimo notebook[1] for exploratory analysis. The source code and documentation are available at https://github.com/SAP/process-evaluation-framework/ and a short video illustrates the framework's main functionality[2].

### 2.2. Steps of the Framework

To produce the similarity score, the framework proceeds in three steps, shown in Figure 1: (i) a conversion step that brings the two models into a common internal representation, (ii) a semantic normalization step that aligns their labels, and (iii) a similarity calculation step that produces the score from the structural and behavioral comparison of the normalized models.

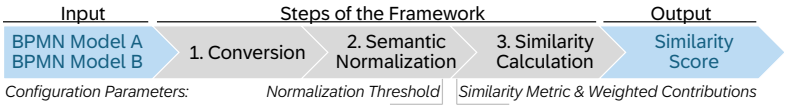


**Figure 1:** Overview of the framework's main steps.

To illustrate the steps of the framework, we use the two procure-to-pay BPMN models shown in Figure 2 as a running example: a standard process and a variant. Both begin with a purchase requisition followed by a model-specific approval subprocess before procurement. The received goods, invoice, and purchase requisition are then checked for consistency. In the first model, this check includes an additional subprocess. If discrepancies are found, the supplier is contacted.

**Step 1: Conversion.** The conversion step converts the two input BPMN models into a common internal representation, enabling the subsequent comparison to be independent of their formats. To this end, each model is parsed into a minimal representation covering activities, events, gateways, pools with their lanes, sequence flows, and message flows. Subprocesses are preserved alongside the activity that contains them, so that they can be compared at the level of their internal structure.

**Step 2: Semantic normalization.** The second step aligns the labels of the two converted models so that synonymous but textually different elements are recognized as the same. Without this alignment, the similarity comparison would treat such elements as different, even though they refer to the same activity, event, or role, leading to a similarity score that underestimates how related the two models actually are. For this, the labels of both models are embedded using a sentence-transformer model[3]

[1] https://marimo.io

[2] https://doi.org/10.6084/m9.figshare.32867483

[3] We use `Alibaba-NLP/gte-large-en-v1.5`, a lightweight English language model; the embedding model should be updated when better performing models become available.

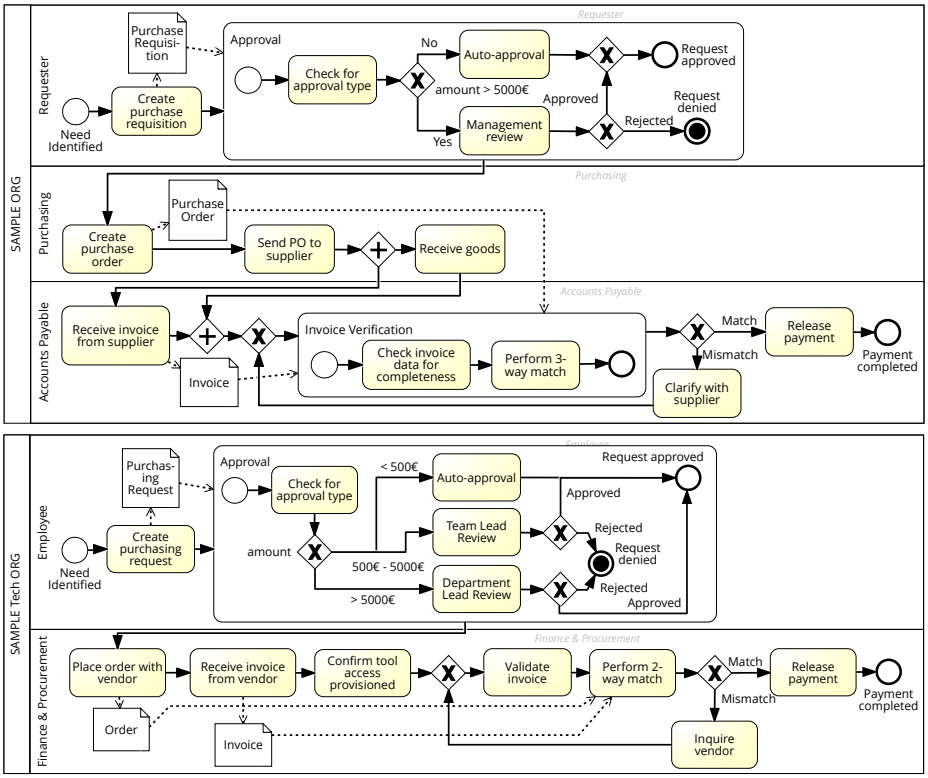


**Figure 2:** Two example BPMN process models of a procure-to-pay process.

and each element is matched to the most similar element in the other model, provided that the cosine similarity exceeds a given threshold. Matched labels in the second model are then rewritten to the corresponding label of the first, providing the next step with a shared vocabulary to compare over.

In our running example, the activity label *Create purchasing request* from the second model is mapped to *Create purchase requisition* in the first, and the lane *Finance & Procurement* is mapped to *Purchasing*.

**Step 3: Similarity calculation.** The third step computes the structural and behavioral similarity between the two normalized models and combines the results into a single configurable score. Both dimensions reduce to a set comparison: over element names/types on the structural side, and over execution traces or sub-traces on the behavioral side. The user selects one overlap metric (Dice, Jaccard, Overlap, Precision, Recall, or F1) that is applied consistently across all set comparisons, and the separate dimension scores are combined into the final score through a user-configured weighted sum.

*Structural similarity.* Structural similarity reflects how much two models share in terms of elements, flows, and organizational structure. The framework captures this by extracting element names and types for activities, events, gateways, flows, lanes, and subprocesses from each model. Duplicates are indexed to preserve repeated elements as distinct items. The resulting sets are compared with the selected overlap metric. The scores are averaged into four high-level scores covering (i) activities, events, and gateways, (ii) sequence and message flows, (iii) lane structure, and (iv) subprocesses, which are then combined into a structural similarity score via a user-configurable weighted sum. If no expanded subprocesses exist, the subprocess weight is redistributed proportionally across the remaining categories.

In our example, after normalizing the gateway sets to gateway types, the first model yields *(Exclusive1, Exclusive2, Exclusive3, Parallel1, Parallel2, Exclusive4, Exclusive5)* and the second *(Exclusive1, Exclusive2, Exclusive3, Exclusive4, Exclusive5).* With two shared elements, the Dice score is $(2 \cdot 5)/(7 + 5) \approx 0.83$.

*Behavioral similarity.* Behavioral similarity reflects how much of the execution behavior permitted by one model is also permitted by the other. This requires a representation that captures the control-flow semantics of the BPMN gateways rather than only the static structure. Each model is therefore converted to a Petri net that preserves these semantics, following an established mapping [10]. A state-space

exploration of the Petri net enumerates the model's feasible execution traces up to a configurable loop depth, ensuring that the resulting trace sets remain finite and comparable. The two trace sets are then compared at two levels of granularity. In the *full-trace mode*, each trace is treated as a single element, so two models score as similar only when they share entire variants. In the *n-gram mode*, each trace is sliced into its length-$n$ contiguous activity subsequences, and the resulting n-grams from all traces form the set that is compared. While small $n$ capture local control-flow and are more forgiving of structural differences, larger $n$ approach the strictness of the full-trace comparison.

For the full example models, full-trace and n-gram comparison lead to different scores. Of the 17 distinct traces, 3 are shared, 8 occur only in the first model, and 6 only in the second, yielding a full-trace Dice score of 0.30. After decomposing the same traces into 2-grams, i.e., directly-follows pairs, 16 of 37 unique 2-grams are shared, increasing the Dice score to 0.60.

**Output.** The framework returns the semantic matches between the two models, the overall similarity score, and the per-dimension and per-category scores underlying it. This lets users assess not only the models' overall similarity, but also the dimensions and categories that contribute to it. The same output is presented in the interactive dashboard, where it is recomputed whenever the user changes the metric, the semantic-normalization threshold, or the weights of the structural and behavioral components.

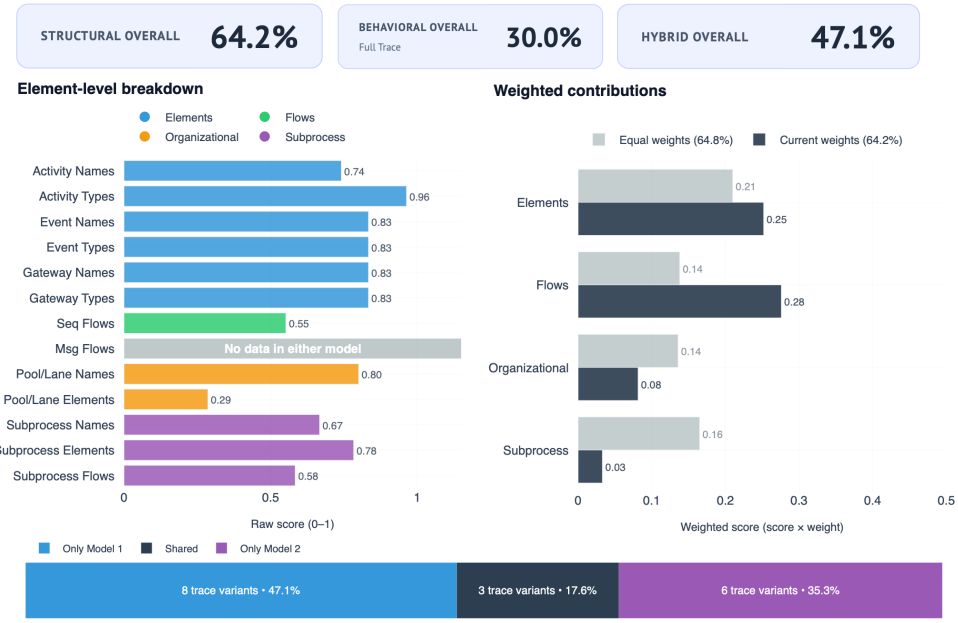


**Figure 3:** Visualization of the structural and behavioral similarity scores of the two provided exemplary models.

For the provided example models, Figure 3 shows the dashboard's structural and behavioral components under default weights and using the Dice metric. The models exhibit high structural similarity, with scores of 0.85 for activities, 0.83 for events, and 0.83 for gateways, reflecting that both models refer to a similar procure-to-pay process. In contrast, the equally weighted organizational similarity score reaches only 0.14, as the two models distribute activities across differently named pools and lanes. In particular, the absence of a third lane in the second model results in disagreement over lane-element pairings. Behaviorally, the dashboard reports a full-trace Dice of 0.30, reflecting a divergence in their fine-grained execution variants. With equal weights for both dimensions, the hybrid similarity is 47.1%.

## 3. Maturity and Practical Relevance

To assess the maturity of our framework, we verified that it works as expected on edge-case inputs, tested its scalability, and assessed the practical relevance of the insights it provides on real-world BPMN models. First, BPMN models often contain errors or edge cases. To verify that our framework works as expected on such input, we implemented a fixture suite of curated example models with automated assertions

about the expected output, covering three behaviors: (i) no crashes on degenerate inputs (empty, single-task, or structurally unsound models), (ii) equivalent BPMN 2.0 XML and SAP Signavio JSON models yield equivalent scores, and (iii) identical models yield ≥95% similarity. Second, we evaluated scalability by running the full framework on $C(154, 2) = 11{,}781$ unordered pairs of 154 proprietary BPMN models ranging from 3 to 52 flow nodes (median 13). Using Dice as the overlap metric, a single comparison took 26 ms on average, and the full run completed in under two hours on a single laptop, with the behavioral dimension dominating runtime. Third, the same experiment demonstrates practical relevance. At a threshold of 0.8 on the weaker of the structural and behavioral scores, the 154 models formed 137 groups, with all 17 non-singleton groups corresponding to known copy-and-versioning duplicates.

## 4. Conclusion

We presented a configurable framework for assessing the similarity of BPMN process models, allowing users to choose the similarity metric, the weights of the structural and behavioral components, and the semantic normalization settings. This supports task-specific similarity scores, for example, for variant analysis and de-duplication, while remaining extensible to additional similarity measures as evaluation needs evolve. The framework has several limitations. Semantic alignment relies on label similarity and does not capture granularity differences. Structural comparison supports subprocesses only to a depth of one. Behavioral comparison captures control-flow via Petri net-based trace exploration, but not other behavioral aspects. Moreover, the framework does not guide users in configuring the similarity score for a given problem. Addressing these limitations, including support for task-specific configuration, is a key direction for future work.

## Declaration of generative AI

During the preparation of this work, the authors used ChatGPT in order to improve the readability and language of the manuscript. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

## References


[1] R. Dijkman, M. L. Rosa, H. Reijers, Managing large collections of business process models - current techniques and challenges, Comput. Ind. 63 (2012) 91–97.

[2] A. Schoknecht, T. Thaler, P. Fettke, A. Oberweis, R. Laue, Similarity of business process models - A state-of-the-art analysis, ACM Comput. Surv. 50 (2017) 52:1–52:33.

[3] M. Garcia, M. Nunes, M. Fantinato, S. Peres, L. Thom, BPMN-Sim: A multilevel structural similarity technique for bpmn process models, Information Systems 116 (2023) 102211.

[4] W. van der Aalst, A. de Medeiros, A. Weijters, Process equivalence: Comparing two process models based on observed behavior, in: BPM, Springer, 2006, pp. 129–144.

[5] R. Dijkman, M. Dumas, B. Van Dongen, R. Käärik, J. Mendling, Similarity of business process models: Metrics and evaluation, Information Systems 36 (2011) 498–516.

[6] T. Thaler, A. Schoknecht, P. Fettke, A. Oberweis, R. Laue, A comparative analysis of business process model similarity measures, in: BPM Workshops, Springer, 2016, pp. 310–322.

[7] F. Zampino, L. Genga, A. Longo, Behavioral similarity in business process models: A perspective that needs more attention, Information Systems (2025) 102608.

[8] S. Ivanov, A. Kalenkova, W. van der Aalst, BPMNDiffViz: A tool for BPMN models comparison, in: BPM Demo, CEUR-WS.org, 2015, pp. 35–39.

[9] M. Voelter, R. Hadian, T. Kampik, M. Breitmayer, M. Reichert, Leveraging generative vision models for extracting process models from documents, in: BPM Workshops, Springer, 2024, pp. 271–282.

[10] R. Dijkman, M. Dumas, C. Ouyang, Formal semantics and analysis of bpmn process models using petri nets, Queensland University of Technology, Tech. Rep (2007) 1–30.