Automating Business Process Modeling: An LLM-Based Approach with Situation-Based Modeling Notation

Gean Paulo O. Souza[1], **Hilário Tomaz A. de Oliveira**[1], **Mateus Barcellos Costa**[1]

[1]Postgraduate Program in Applied Computing (PPComp)
Instituto Federal do Espírito Santo (IFES) – Serra, ES – Brazil

gean@lastrum.co, {hilario.oliveira, mcosta}@ifes.edu.br

Abstract. ***Research Context:*** *Although business process modeling is beneficial for clarifying organizational processes, it remains a task with little automation and requires significant human effort.* ***Scientific and/or Practical Problem:*** *To reduce this effort, research has advanced toward automating the task, mainly focusing on extracting imperative models from textual descriptions. However, imperative models may lead to a single solution that does not necessarily represent the best fit.* ***Proposed Solution and/or Analysis:*** *The declarative Situation-Based Modeling Notation (SBMN) allows modelers to generate multiple modeling structure alternatives, enabling them to choose the one that best fits the context. This work investigates the use of Large Language Models (LLMs) to identify core business process entities and constraints represented in SBMN from textual descriptions.* ***Related IS Theory:*** *This research is grounded in the Task-Technology Fit (TTF) theory, applied to evaluate the adequacy of LLMs in supporting business process analysts during modeling activities.* ***Research Method:*** *A dataset of 133 textual descriptions paired with SBMN models was employed. Each description was processed by three medium-scale LLMs and three extra-large LLMs using zero-shot prompts to identify domain situations and Active Flow Objects. The outputs were semantically compared with the corresponding SBMN models to measure their similarity.* ***Summary of Results:*** *Large-scale LLMs (e.g., Claude Sonnet and Qwen3-80B) were able to identify flow objects and situations in SBMN with relevant accuracy, demonstrating the feasibility of their use to support the modeling task.* ***Contributions and Impact to IS area:*** *This research provides: (i) empirical evidence highlighting the strengths and limitations of current LLMs in the process modeling domain; (ii) a dataset containing aligned pairs of textual descriptions and SBMN models; and (iii) a semantic–structural similarity metric enabling the quantitative evaluation of SBMN component extraction accuracy from text.*

1. Introduction

A business process is a structured sequence of interrelated activities designed to achieve a well-defined organizational objective, thereby creating value for both the organization and its customers. To support the understanding, analysis, and continuous improvement of such processes, Business Process Modeling (BPM) is a core discipline that fosters shared understanding among stakeholders involved in their execution [Dumas et al. 2018]. As discussed by Beerepoot et al., the design, modeling, and analysis of business processes remain among the most human-dependent tasks in the BPM life cycle [Beerepoot et al. 2023, Avila et al. 2023].

Business process modeling comprises a set of tasks aimed at providing a well-defined representation of organizational processes [Kourani et al. 2024]. Process Elicitation [Motta et al. 1989] is an early step in modeling, with the objective of obtaining a general understanding of the process. This step typically involves multiple participants and the collection of information from diverse data sources, which often consumes a substantial portion of the effort invested in the initial stages of modeling. Depending on the size of the organization and the complexity of its processes, modeling may require significant human effort. In this context, different technical approaches have been proposed, such as process mining techniques that explore event log data, and apply Natural Language Processing (NLP) methods to parse and extract information from unstructured data sources, such as textual documents [Ferreira and Thom 2016].

Most elicitation support activities have focused on the considerable effort required to generate complete process models using semantically rich notations or languages as the target notations for the resulting models. Such notations are generally grounded in an imperative or procedural paradigm, exemplified by Petri nets or Business Process Model and Notation (BPMN), although several studies have also explored the generation of models based on declarative notations, such as Declare. Imperative models are characterized by a rigid specification of the execution flow that must be followed within a process. Nevertheless, even typical operational business processes may exhibit different control-flow structures while yielding equivalent outcomes [Silva de Castro et al. 2025].

Given that such structural variations may be intertwined within process information sources and should themselves be evaluated before model definition, the early construction of a complete process model from elicitation data becomes a challenging endeavor. Declarative approaches, such as the one proposed by [Balabko et al. 2005], enable the definition of more flexible, or even incomplete but consistent, models expressed as a set of constraints over entities and their interrelations. This latter approach can therefore serve as a bridge between elicitation data and complete process models.

In a previous work [de Santana Candido et al. 2025], a declarative notation was employed to explore the use of advanced NLP techniques, Machine Learning (ML), and hybrid methods for automating the extraction of business process entities and constraints from text, using an annotated dataset for training and evaluation. A key contribution of that work was the creation of a publicly annotated dataset comprising textual descriptions of business processes from different domains. In the present work, the use of Large Language Models (LLMs) is explored with the same purpose as [de Santana Candido et al. 2025], and based on the same dataset. LLMs were applied to extract entities and constraints in accordance with the declarative notation of the Situation-Based Modeling Notation (SBMN). As in [de Santana Candido et al. 2025], the obtained model can subsequently be used to generate imperative models that represent the same business process.

Six Large Language Models were evaluated to extract different types of entities and their respective relations: Qwen3-Next-80B, Claude-Sonnet-4.5, Llama-3.3-70B, Qwen2.5-7B, Meta-Llama-3.1-8B, and Granite-3.3-8B. The evaluation relied on a dataset composed of 133 business textual descriptions from different domains, each paired with a corresponding SBMN model. This dataset was derived from the human-annotated corpus presented in [Candido et al. 2024], where entities and constraints representing Active

Flow Objects and Situations, respectively, were manually labeled. The LLM-generated models were then semantically compared with those in the reference dataset. Among the evaluated models, Claude-Sonnet-4.5, Qwen3-Next-80B, and Llama-3.3-70B achieved the best overall performance in terms of semantic similarity to human-generated SBMN models.

The remainder of this paper is organized as follows. Section 2 discusses the theoretical and conceptual foundations of process model extraction. Section 3 presents the experimental methodology, detailing the dataset configuration, the criteria adopted for selecting the language models, and the procedures used for comparative assessment. Section 4 reports and analyzes the experimental results, while Section 5 concludes the paper and outlines directions for future research.

2. Theoretical Background

2.1. Extracting Imperative-style Models from Text

Early research focused on extracting imperative process models, particularly BPMN. [Friedrich et al. 2011], for example, proposed a rule-based approach to identify activities, gateways, and flows, generating executable BPMN diagrams from natural language textual documents. The proposed extraction method builds upon theories of computational linguistics and NLP techniques to automatically derive imperative process models, particularly in the BPMN notation, from unstructured textual descriptions. The approach assumes that process knowledge embedded in natural language can be systematically uncovered through syntactic, semantic, and referential analyses.

At the syntactic level, the method employs the *Stanford Parser* [De Marneffe and Manning 2008] to generate dependency trees and grammatical relations, enabling the identification of key process entities such as *Actors*, *Actions*, and *Objects*. Semantic interpretation is then performed using lexical databases such as *WordNet* and *FrameNet*, which provide synonymy, hypernymy, and frame-based relations to normalize and enrich the extracted predicates. Additionally, a dedicated anaphora resolution mechanism addresses referential ambiguity across sentences, ensuring coherence in the linking of actions and participants.

The extracted linguistic information is stored in an intermediate structure called the *World Model*, adapted from the CREWS scenario metamodel, proposed by Achour et al. [Achour 1999]. The Cooperative Requirements Engineering With Scenarios (CREWS) framework was originally designed to support scenario-based requirements elicitation and formalization. Its metamodel defines a structured way to represent system behavior in terms of *Actors*, *Goals*, *Actions*, and *Flows of events*, emphasizing traceability between textual requirements and conceptual models. This representation formalizes four main concepts (*Actor*, *Action*, *Resource*, and *Flow*) and preserves traceability between textual and model elements. It serves as a semantic bridge between language structures and BPMN constructs.

The overall transformation pipeline consists of three analytical phases: (i) sentence-level analysis, responsible for identifying atomic activities and filtering non-processive content; (ii) text-level analysis, which resolves conditional and temporal markers to define control-flow semantics; and (iii) model generation, which instantiates BPMN

flow objects and gateways from the enriched World Model. Conceptually, the approach operationalizes the mapping between linguistic semantics and process modeling constructs. By integrating syntax, semantics, and discourse cues, it transforms textual process descriptions into executable BPMN models. The empirical evaluation across 47 text–model pairs demonstrated that the approach can correctly reproduce about 77% of the original models, confirming the feasibility of linguistically based process model acquisition.

More recent studies leverage machine learning and deep learning. [Licardo et al. 2024] propose a method to automatically derive BPMN models from textual descriptions using a pipeline combining NLP, a fine-tuned BERT token classification model, and large language models (GPT-3.5-Turbo, GPT-4). They evaluated their approach on 31 textual descriptions, using the Relative Graph Edit Distance (RGED) to compare the generated models with the reference ones. The results showed an average accuracy of about 96 % with GPT-4 and around 80 % with GPT-3.5-Turbo, considering the accuracy metric derived from (1 – RGED). [Nivon and Salaün 2025] proposed a tool to generate BPMN diagrams from textual process requirements. The pipeline was implemented as a web tool and tested on 200 textual process descriptions (25 % from literature, 75 % written by 9 users, experts, and novices). Each generated BPMN model was compared with an expert-defined reference and classified as correct, ambiguous, or incorrect. 78.5 % of the models matched the reference, 8 % were ambiguous, and 13.5 % incorrect.

2.2. Extracting Declarative Models from Text

Although most studies on the automation of business process elicitation have focused on the generation of imperative-style models, there is a growing research interest in exploring declarative approaches, which enable more flexible and generalizable representations of process behavior. Declarative process model extraction from natural language text was investigated, for instance, by [van der Aa et al. 2019]. In their work, NLP techniques were employed to identify semantic components essential for representing behavioral constraints in the DECLARE notation [Pesic et al. 2007], which were then combined to construct the final model. The experiments achieved a precision of approximately 0.77 and a recall of 0.72 when compared to manually created models. This study was among the first to demonstrate that process knowledge can be systematically captured beyond imperative modeling, thereby establishing a theoretical foundation for research seeking to derive flexible, constraint-based process representations directly from textual descriptions.

In this direction, [de Santana Candido et al. 2025] advanced the theoretical integration of NLP and Machine Learning (ML) within the domain of process model extraction by proposing a hybrid architecture grounded in both symbolic and statistical paradigms. Their approach conceptualizes textual process descriptions as structured linguistic artefacts from which semantic entities and control-flow constraints can be systematically derived. In this framework, *Named Entity Recognition* (NER) operates as a semantic segmentation task that maps linguistic constituents to ontological categories of the Situation-Based Modeling Notation (SBMN), namely *Actor*, *Activity*, *Trigger*, *Catch*, and *Conditional*. The subsequent *relation classification* stage models the relational semantics between these entities, distinguishing temporal and logical dependencies such as

Strict Dependence, *Circumstantial Dependence*, *Union*, and *Non-Coexistence*.

From a theoretical perspective, the methodology aligns with the paradigm of *information extraction* and extends it to represent declarative process semantics. By combining BiLSTM-CRF architectures for sequential labeling with contextual word embeddings (GloVe and Flair), the authors formalize the correspondence between linguistic cues and process constructs through distributed representations that capture both syntagmatic and paradigmatic relations. Furthermore, by employing transformer-based models, specifically RoBERTa, for relation classification, the study leverages attention mechanisms to approximate predicate-argument structures, thereby modeling long-range dependencies and implicit constraints that transcend sentence boundaries.

Empirically, the research is grounded on an annotated corpus of 133 documents containing 5,395 expert labels, which operationalizes the mapping between linguistic entities and SBMN concepts. The results demonstrate that the hybrid neural-symbolic strategy achieved high precision in identifying frequent entities and relations, validating the hypothesis that deep contextual embeddings can encode both lexical and relational semantics essential to process modeling. However, the authors also report that relations with lower occurrence frequencies, such as *Union* and *Non-Coexistence*, yield lower accuracy. This limitation is theoretically attributable to data sparsity and class imbalance, which constrain the model's capacity to generalize to low-density semantic phenomena. Overall, the study consolidates a theoretical bridge between linguistic representation learning and declarative process modeling, suggesting that neural architectures can serve as functional approximations of semantic interpretation mechanisms traditionally addressed through rule-based formalisms.

The study demonstrated that the approach achieved strong performance in identifying entities and relations; however, significant differences were observed for elements with fewer occurrences in the dataset. This finding underscores that the scarcity of annotated datasets in the target domain remains a key limitation to the application of such methods. Building upon this foundation, the present work investigates the use of Large Language Models (LLMs) for the same purpose, employing the Situation-Based Modeling Notation (SBMN) as the target declarative representation and using the same annotated dataset to evaluate extraction consistency and semantic alignment.

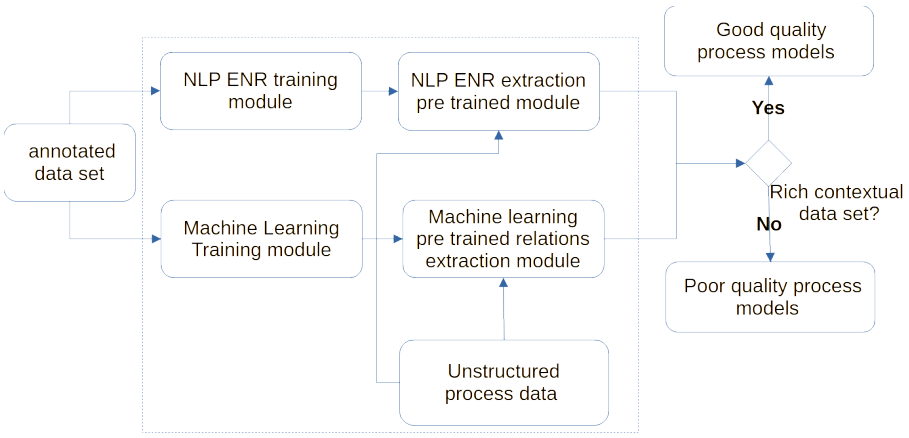

Figure 1. Schematic representation of the NLP and Machine Learning pipeline for process model extraction.

The hybrid approach proposed by [de Santana Candido et al. 2025] is illustrated in Figure 2, whereas the architecture investigated in this work is depicted in Figure 2. In this novel approach, the central challenge is to design a *prompt-based module* that integrates structured semantic relations connecting both *syntagmatic* and *paradigmatic* elements within textual descriptions. The goal is to capture these latent linguistic dependencies and transform them into the corresponding entities and constraint relations that constitute the SBMN models. This integration demands not only lexical understanding but also a representation of how semantic roles interact across the text, bridging distributed linguistic features with symbolic process knowledge.

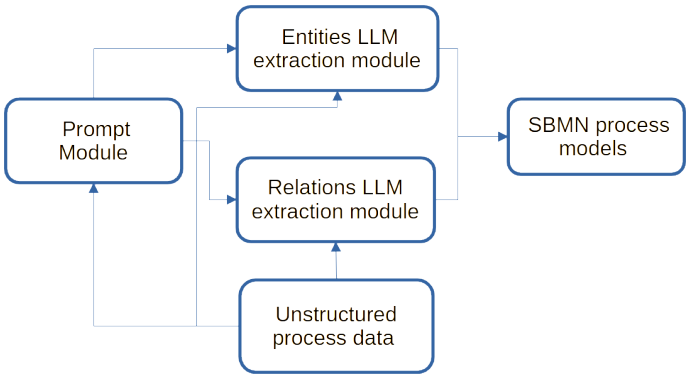

Figure 2. Schematic representation of the LLM Approach for process model extraction.

2.3. Large Language Models

Large Language Models (LLMs) are neural architectures designed to learn and reproduce linguistic patterns by predicting the next token in a sequence, given a large corpus of textual data. They are typically based on the Transformer architecture [Vaswani et al. 2017], which employs self-attention mechanisms to capture long-range dependencies and contextual relationships within text. Through this architecture, LLMs acquire statistical and semantic representations that enable them to perform a wide variety of tasks, including text generation, classification, translation, and reasoning.

From a theoretical standpoint, LLMs can be distinguished along three complementary dimensions: (i) architecture, such as decoder-only models (e.g., GPT, Llama, Qwen) optimized for generative tasks; (ii) training objectives and datasets, which influence the model's exposure to domain-specific linguistic structures and pragmatic cues; and (iii) parameter scale, which correlates with the model's representational capacity and generalization ability. In this study, six Large Language Models were selected for evaluation. The selection criteria followed the same rationale as in [Pedroso et al. 2025], prioritizing publicly accessible models rather than those typically available through academic or proprietary research platforms, which are distinguishable in terms of scale, training data, and fine-tuning strategy and may lead to distinct behaviors in tasks that require semantic abstraction, such as identifying entities and relations in textual process descriptions. Larger models, for instance, are expected to generalize better to complex linguistic phenomena, while medium-scale models may exhibit variability and inconsistencies in producing structured outputs such as SBMN models.

2.4. Situation Based Modeling Notation

The Situation-Based Modeling Notation (SBMN), adopted as the target notation in the model extraction method, extends a concise and intuitive notation originally introduced by Costa and Tamzalit [Costa and Tamzalit 2017] in the context of recommendation patterns for business process modeling. In its initial formulation, the notation aimed to provide a means of representing *situations* as abstractions that capture recurrent behavioral patterns within processes. The central premise was that a situation should encapsulate the essential or mandatory behaviors of a process, enabling the systematic derivation of procedural model fragments from these high-level declarative constructs. Formally, a *situation* is defined as a binary relation whose operands are sets of *Active Flow Objects* (AFOs) within a process model. The set of AFOs includes all elements that represent activities, whether composite or atomic tasks, as well as events. Table 1 summarizes the *situations* considered in the proposed extraction approach that includes those presented in [Costa and Tamzalit 2017] and an additional situation called *Perform* introduced in [Candido et al. 2024].

Table 1. Situations adopted in this work.

Situation	Description
Strict Dependence	A set of active flow objects with a temporal dependence among them.
Circumstantial Dependence	A set of active flow objects with a conditional dependence among them.
Non-coexistence	A set of flow objects with a non-coexistence relation at the same execution flow. If there is a non-coexistence relation between the flow objects sets A and B, any execution flow executing both A and B is forbidden.
Union	A set of flow objects with a union relation at the same execution flow. A union relation between the flow objects set A and B requires a process model with a flow executing A, a flow executing B and a flow executing both A and B
Perform	Represents the relation between an *Actor* and an *Activity*

Since this study employs a dataset adapted from [Candido et al. 2024], the same catalog of *Active Flow Objects* (AFOs) and *Situations* was adopted. This catalog represents an adaptation of the original one proposed by [Costa and Tamzalit 2017], extended with additional AFO categories, an additional situation called *Perform*, and the absence of the *Independence* relation from the set of Situations. For the *Dependence* Situation, only its two specific types were considered, namely, *Strict Dependence* and *Circumstantial Dependence*. Table 2 summarizes the set of Active Flow Objects adopted in this study.

3. Research Method

In this section, the main aspects of the experimental phase of this research are presented. Subsection 3.1 describes the structure and composition of the dataset employed in the

Table 2. Active Flow Object adopted in this work.

Active Flow Object	Description
Actor	Responsible for actions in the process.
Activity	Tasks or operations in the process.
Trigger	Events that start a process.
Catch	Events that capture conditions.
Conditional	Conditions tied to dependencies.

experiments, while Subsection 3.2 details the procedures adopted for its construction and annotation. The subsequent subsection introduces the large language models utilized in this study, outlining their main characteristics and configuration. Finally, Subsection 3.4 discusses the evaluation methodology, highlighting the criteria used to assess the performance of the language models and to semantically compare the generated SBMN models with the original ones in the dataset.

3.1. Dataset Specifications

The dataset used in this work was adapted from the dataset presented in [Candido et al. 2024], resulting in a new dataset composed of 133 textual descriptions of business processes that mainly cover seven business domains: Human Resources, Finance and Accounting, Procurement, Purchasing and Supply Management, Sales, Commercial and Customer Service, Logistics, Manufacturing and Production, Energy, Utilities and Metering, and Legal and Regulatory Administration. Each textual description is paired with an SBMN model containing entities, represented as Active Flow Objects, and constraints, represented as Situations. The dataset proposed by [Candido et al. 2024] considers the following AFOs and Situations, whose meaning in the SBMN context is discussed in Section 2.4: the AFOs are Actor, Activity, Trigger, Catch, and Conditional; and the Situations are Strict Dependence, Circumstantial Dependence, Union, Non-Coexistence, and Perform.

To represent the SBMN models in the dataset used in this work, a textual syntax based on the SBMN models' representations present in [Silva de Castro et al. 2025] was adopted. The models are represented in a textual format, containing two sections: Domain and Situations. The Domain section lists all AFOs sequentially, with each entry organized into three columns. The first column indicates the entity type (e.g., actor, activity, trigger), the second provides a unique identifier, and the third specifies the entity name. Similarly, the Situations section consists of three columns: the second and last columns refer to the entities involved, while the first column represents the execution dependency that links them. Table 3 presents an example of an SBMN model derived from a process description included in the dataset.

3.2. Dataset Construction

To construct the dataset employed in this study from the original corpus presented in [Candido et al. 2024], the annotated text segments and their corresponding labels were extracted and reformatted to generate the SBMN models. The original labels were preserved and used to name entities and situations in each SBMN model, while a unique identifier was assigned to each entity derived from the annotations.

Table 3. Example of business process and its SBMN model.

Business Process:
Once a loan application is received by the loan provider, and before proceeding with its assessment, the application itself needs to be checked for completeness. If the application is incomplete, it is returned to the applicant, so that they can fill out the missing information and send it back to the loan provider. This process is repeated until the application is found complete.

SBMN Model:

```
Domain *****************
activity e1 application is checked for
completeness
activity e2 application is returned to the
applicant
activity e3 fill out the missing information
activity e4 send the application back to loan
provider
actor e5 loan provider
actor e6 applicant
catch e8 loan application is received
catch e9 application is found complete
condition e10 the application is incomplete
Situations *****************
circumstantial_dependence e2 e10
perform e6 e3
strict_dependence e1 e8
strict_dependence e4 e3
```

Because the source dataset followed the Conference on Natural Language Learning (CoNLL) format, originally designed for Named Entity Recognition (NER) tasks, it contained duplicated entities and terms that lacked semantic relevance to SBMN modeling, such as isolated pronouns. Consequently, a preprocessing phase was applied to remove redundant instances and entries that could compromise the semantic consistency of the generated SBMN models. The following preprocessing steps were performed: pronouns were replaced with their corresponding referents; identical or semantically equivalent entities were grouped into a single entity with a unique identifier; and qualifiers were removed, with actors that differed only by qualifiers unified into a single entity without the qualifier.

3.3. Large Language Models Investigated

Six Large Language Models (LLMs) were selected to be evaluated in this study. The selection criteria for the models followed the same approach as in [Pedroso et al. 2025], prioritizing models more accessible to the general public rather than those most commonly used in the academic community. For comparison purposes, this study evaluates

two groups of LLMs with different numbers of parameters.

Medium-scale LLMs comprise architectures with approximately 7B to 30B parameters, including Qwen 2.5 with 7 billion (7B) of parameters, Granite 3.3 8B, and Llama 3.1 8B. In contrast, extra-large LLMs are characterized by substantially higher capacity, ranging from 32B to more than 70B parameters, and are represented in this study by Qwen 3 80B, Claude Sonnet 4.5, and Llama 3.3 70B.

3.4. Comparison Between SBMN Models

The comparison between two SBMN models involves measuring their similarity in terms of both *Active Flow Objects* (AFOs) and *Situations*. In this study, we propose a semantic–structural similarity metric to enable quantitative evaluation of the accuracy of SBMN component extraction from text. Two AFOs are considered equivalent when they share the same type and have high cosine similarity in their names. In this study, a threshold value of 0.7 was adopted. To determine this threshold, an experiment was conducted using a random sample of text-SBMN model pairs from the dataset. The texts in the sample were processed through the extraction pipeline using all LLMs. During the similarity measurement stage done by the word embedding model, it was observed that AFOs and situations generated by the LLMs that exhibited semantic discrepancies compared to the AFOs in the reference models had values that occasionally approached 0.7 but rarely exceeded it. For that reason, this value was adopted as the threshold for determining whether two AFOs or two situations are semantically similar.

An Active Flow Object (E) is defined as a tuple:

$$E = (t, n) \tag{1}$$

where $t \in T$ represents the *type* of the object (e.g., activity, actor, trigger, and others.), and $n \in \Sigma^*$ denotes the *name* represented as a textual description of the entity.

Two AFOs, $E_i = (t_i, n_i)$ and $E_j = (t_j, n_j)$, are considered equal if:

$$t_i = t_j \ \wedge \ \text{cosine\_sim}(n_i, n_j) \geq 0.7 \tag{2}$$

This equation states that two AFOs are considered equal or equivalent only if they share the same type and the cosine similarity between their names is greater than or equal to 0.7. This condition guarantees that both the structural role of the object (captured by its type) and its semantic meaning (captured by the textual similarity of the name) are preserved in the comparison.

A Situation is defined as a tuple:

$$S = (t, e_1, e_2) \tag{3}$$

where $t \in T_S$ represents the *type* of the situation (e.g., union, perform, strict dependence, and others), and $e_1, e_2 \in E$ are the AFOs involved in the relation.

Given two situations

$$S_i = (t_i, e_{1i}, e_{2i}) \quad \text{and} \quad S_j = (t_j, e_{1j}, e_{2j}),$$

the similarity between them is defined as:

$$\text{sim}(S_i, S_j) = \begin{cases} \max\left(\frac{\text{sim}(e_{1i},e_{1j})+\text{sim}(e_{2i},e_{2j})}{2},\ \frac{\text{sim}(e_{1i},e_{2j})+\text{sim}(e_{2i},e_{1j})}{2}\right), & \text{if } t_i = t_j \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $\text{sim}(e_x, e_y)$ corresponds to the similarity between AFOs as previously defined.

Two situations S_i and S_j are considered *equal or equivalent* if:

$$t_i = t_j \ \wedge\ \text{sim}(S_i, S_j) \geq 0.7 \tag{5}$$

The similarity function $\text{sim}(S_i, S_j)$ is designed to capture the equivalence between two situations. First, situations are only comparable if they share the same type ($t_i = t_j$); otherwise, their similarity is defined as zero.

If the types match, the similarity is computed by comparing the pairs of entities involved. Since a situation is defined over two entities, there are two possible ways to align them: a *direct comparison* ($e_{1i} \leftrightarrow e_{1j},\ e_{2i} \leftrightarrow e_{2j}$) and a *crossed comparison* ($e_{1i} \leftrightarrow e_{2j},\ e_{2i} \leftrightarrow e_{1j}$). For each alignment, the average similarity of the two pairs of Active Flow Objects is calculated, and the final similarity is the maximum of these two averages. This strategy ensures that situations are considered similar even when the order of their entities is reversed. Finally, two situations are considered *equal or equivalent* if the resulting similarity value is greater than or equal to the empirically defined threshold of 0.7.

To measure the similarity of all AFOs and Situations between two SBMN models, a Similarity Formula was applied. This formula computes a normalized score ranging from 0 to 1 that quantifies the similarity between the models, considering their AFOs and Situations separately. For AFOs, the similarity is obtained by matching entities with equivalent types and semantically similar names, as defined in Equation (2). For Situations, the similarity between pairs of relations is computed according to Equations (3)–(5), ensuring that both the constraint type and the semantic proximity of the connected entities are preserved.

The similarity between the set of AFOs and Situations present in two SBMN models M_1 and M_2 is calculated as

$$\text{Sim}(M_1, M_2) = \frac{1}{\max(|M_1|, |M_2|)} \max_{\pi} \sum_{e \in M_1} \left(\tfrac{1}{2}\,\delta_{t_e, t_{\pi(e)}} \ +\ \tfrac{1}{2}\,S_{\text{nome}}(x_e, x_{\pi(e)})\right)$$

In this equation:

- M_1 and M_2 are the sets of entities(AFOs or Situations) in each model; $|M|$ is the number of entities in M.

- π is the optimal one-to-one matching between entities of M_1 and M_2 (e.g., Hungarian algorithm); unmatched entities contribute 0.
- $S_{\text{nome}}(x_e, x_{\pi(e)}) \in [0, 1]$ measures the lexical/semantic similarity between the names of the paired entities (e.g., Jaccard or cosine over embeddings). In this work, the all-mpnet-base-v2 embedding model [Reimers and Gurevych 2019, Song et al. 2020] provides the semantic vectors for computing this similarity. A threshold value of 0.7 was selected based on preliminary experiments that showed it provided a good balance between ignoring spurious matches and capturing semantically meaningful similarities.
- $\delta_{t_e, t_{\pi(e)}}$ is the Kronecker delta: 1 if the types are equal and 0 otherwise (strict type match).
- The two components(type and name) have equal weight ($\frac{1}{2}$ each); the denominator $\max(|M_1|, |M_2|)$ normalizes the final score to $[0, 1]$.

4. Experiments and Results

To evaluate the capacity of LLMs to generate SBMN models equivalent to human-generated models, a prompt was built using a zero-shot approach with reasoning steps [Kojima et al. 2023], along with additional explanations of the output format and the types of entities (Active Flow Objects) and relations (Situations) to be considered. The prompt instructs the model to insert the final SBMN models within [answer] tags to make them easier to separate from the step-by-step reasoning section of the text.

A temperature of zero was applied to all LLMs to reduce randomness and ensure a more deterministic response. The maximum number of tokens (max_tokens) was set to 10,024 to ensure the models have sufficient token space for step-by-step reasoning. Each business process description in the dataset was inserted into the prompt, which was then executed across the six LLMs investigated here. The SBMN models generated for each LLM were extracted from the output, and their entities and relations similarities with the corresponding human-generated model present in the dataset were computed using the Similarity Formula detailed in the Subsection 3.4.

4.1. Similarities Between AFOs

Table 4 summarizes the AFO similarity statistics for each LLM, including the mean, standard deviation, median, maximum, and minimum similarity scores. The best mean result is highlighted in bold. It can be observed that Claude-Sonnet-4.5 achieved the highest mean entity similarity, followed by Qwen3-Next-80B and Llama-3.3-70B. Meta-Llama-3.1-8B and Qwen2.5-7B obtained intermediate results, whereas Granite-3.3-8B performed the worst.

To further illustrate the performance of the LLMs, Figure 3 presents the similarity score distribution of the models generated by the LLMs. The boxplots for Claude and Qwen-80B are centered on higher values with lower dispersion, while Granite exhibits a wide spread and numerous low outliers. Minimum scores further illustrate the performance gap; Claude never falls below 0.35, whereas Granite reaches as low as 0.03.

Among the medium-scale LLMs, Granite demonstrated the weakest performance, achieving a minimum similarity of zero for both AFOs and Situations. This outcome can be attributed to invalid SBMN structures generated by the model, such as AFOs lacking

Table 4. AFO similarity statistics between models.

LLM	Mean	Median	Min	Max
Qwen3-Next-80B-A3B-Instruct	0.61 ± 0.11	0.60	0.32	0.92
Claude-sonnet-4.5	**0.66** ± 0.11	0.68	0.35	0.91
Llama-3.3-70B-Instruct-Turbo	0.55 ± 0.12	0.55	0.16	0.83
Qwen2.5-7B-Instruct-Turbo	0.45 ± 0.10	0.44	0.21	0.67
Meta-Llama-3.1-8B-Instruct-Turbo	0.50 ± 0.09	0.49	0.27	0.72
Granite-3.3-8b-instruct	0.29 ± 0.16	0.29	0.00	0.66

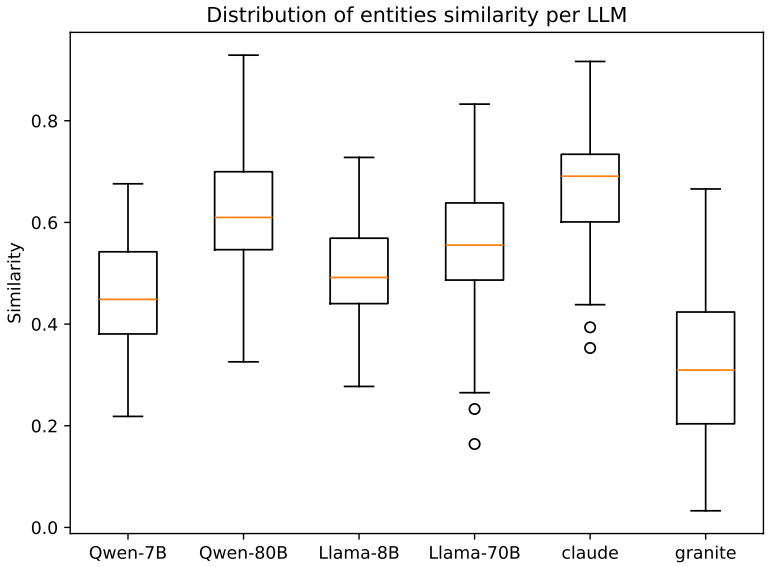

Figure 3. Distribution of AFO similarity for each LLM.

proper identifiers, which prevented the corresponding entities from being counted. Although most AFOs and Situations produced by Granite were correctly categorized, comparison with the human-generated models revealed that a substantial number of relevant elements were missing from its outputs.

In several SBMN models generated by Llama-8B, some AFO texts explicitly expressing conditions (e.g., those beginning with subordinating conjunctions) were incorrectly labeled as activities. This behaviour suggests that certain LLMs may conflate the conceptual distinctions among AFO categories as defined in the prompt, leading to misclassification of entities.

Interestingly, in the outputs of larger LLMs, such as Claude and Qwen-80B, some correctly identified AFOs appeared in the generated SBMN models but were absent in the human references. This finding suggests that the original dataset may contain incomplete or inconsistently annotated SBMN models, particularly regarding the categorization of AFOs and Situations.

4.2. Similarities Between Situations

Table 5 summarizes the Situation similarity statistics for each LLM and Figure 4 presents the score distribution. The Qwen3-Next-80B, Llama-3.3-70B, and Claude-Sonnet-4.5

achieved the highest performance, with mean similarity scores ranging from 0.45 to 0.48 and median values above 0.45, indicating consistent alignment across Situations. Qwen2.5-7B and Meta-Llama-3.1-8B produced intermediate results, characterized by lower central tendencies and greater dispersion. In contrast, Granite-3.3-8B exhibited the weakest performance, with extremely low means (0.15) and medians (0.02) and high variability.

Table 5. Situation similarity statistics between models.

LLM	Mean	Median	Min	Max
Qwen3-Next-80B-A3B-Instruct	0.48 ± 0.17	0.51	0.11	0.84
Claude-sonnet-4.5	0.45 ± 0.15	0.46	0.09	0.81
Llama-3.3-70B-Instruct-Turbo	0.47 ± 0.16	0.48	0.02	0.82
Qwen2.5-7B-Instruct-Turbo	0.36 ± 0.16	0.39	0.00	0.82
Meta-Llama-3.1-8B-Instruct-Turbo	0.35 ± 0.14	0.35	0.03	0.71
Granite-3.3-8b-instruct	0.15 ± 0.17	0.02	0.00	0.56

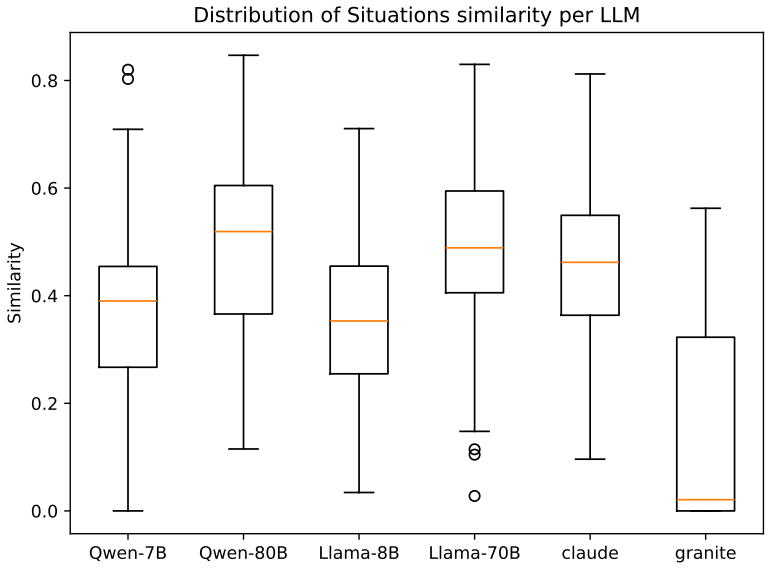

Figure 4. Distribution of Situation Similarity for each LLM.

In contrast to the results observed in the AFO experiments, the Qwen-7B model showed Situation statistics with a minimum score of zero. This outcome resulted from syntactic errors in the Situations section of some generated SBMN models, which prevented valid Situations from being counted, even when the LLM correctly identified them.

Overall, the experiments indicate that larger and more recent LLMs (Claude, Qwen-80B, and Llama-70B) exhibit greater overlap with human references, suggesting superior capabilities for business process understanding and abstraction. These findings suggest that some LLMs can approach or even surpass human-level SBMN modeling quality, whereas medium-sized models still tend to omit relevant structural elements.

5. Conclusion

This work investigated the use of Large Language Models (LLMs) for automating the identification of entities and constraints defined in the Situation-Based Modeling Notation (SBMN) from textual business process descriptions. The experiments demonstrated that larger and more recent LLMs, such as Claude Sonnet 4.5 and Qwen-3-80B, achieved higher accuracy in identifying Active Flow Objects and Situations, approaching human-level quality. These results indicate the feasibility of using LLMs as intelligent assistants for business process analysts, reducing manual effort in model construction and enabling more flexible, declarative representations compared to purely imperative approaches.

Despite the promising results, this study has certain limitations. The evaluation relied on a human-annotated dataset that contains inconsistencies and does not fully ensure that the reference models accurately capture all aspects of the original business domains, potentially leading to inaccuracies in the similarity measures. As discussed in Section 4, some LLMs generated semantically valid entities and situations that were absent from the reference dataset, highlighting the need for additional work to improve its overall quality and completeness. Future research should therefore include methods for assessing whether an SBMN model accurately represents the business context described in the textual input.

The contributions of this research include: (i) the extension of an existing human-annotated dataset by associating each of its 133 business process descriptions with a corresponding SBMN model derived from the original annotations; (ii) empirical evidence highlighting the strengths and limitations of current LLMs in the process modeling domain; (iii) a methodology for semantically comparing the similarity between two SBMN models; and (iv) practical insights into prompt design strategies to enhance model extraction accuracy. Collectively, these contributions create opportunities to develop semi-automated tools that assist analysts in designing, validating, and refining process models with reduced cognitive effort. Furthermore, this work aligns with the challenges identified in the Open World perspective [Araujo 2016], by advancing approaches that facilitate the modeling and understanding of business processes in increasingly open and collaborative organizational ecosystems.

For future research, several directions are suggested. First, the development of validation mechanisms for SBMN models to ensure both their internal consistency and their alignment with the domain requirements described in the textual input. Such mechanisms could be used to enhance the quality of the dataset employed in this study and to validate the SBMN models generated by LLMs. Second, the definition of a formal syntax or Domain-specific Language (DSL) for SBMN, with explicit grammar and rules, so that LLMs can produce well-formed models that are easier to validate and integrate into modeling tools. Third, the exploration of alternative prompt engineering strategies (e.g., few-shot learning, chain-of-thought prompting) and the investigation of more robust similarity measures beyond general-purpose embeddings, since current results remain sensitive to the choice of embedding model. Finally, integrating LLM-based SBMN extraction into end-to-end business process management toolchains enables organizations to automatically generate, validate, and adapt process models more efficiently in real-world scenarios.

6. Supplementary Materials

All datasets and source codes used in this work are available at: `http://dx.doi.org/10.6084/m9.figshare.30300499`.

Acknowledgments

The authors thank FAPES/UnAC (Nº FAPES 1228/2022 P 2022-CD0RQ, Nº SIAFEM 2022-CD0RQ) for the financial support provided through the UniversidaES System.

The authors declare the use of ChatGPT to support the following activities: text translation and grammatical corrections; LaTeX formatting of text, tables, and equations; and assistance in developing the Python source code used in the experiments.

References

Achour, C. B. (1999). Guiding scenario authoring[1]. *Information Modelling and Knowledge Bases X*, 51:152.

Araujo, R. (2016). Information systems and the open world challenges. In *I GranDSI-BR: Grand Research Challenges in Information Systems in Brazil (2016–2026)*, chapter 4, pages 42–51. Brazilian Computer Society (SBC), Porto Alegre, Brazil.

Avila, D. T., de Moura, V. C., and Thom, L. H. (2023). Using machine learning to classify process model elements for process infrastructure analysis. In *Proceedings of the XIX Brazilian Symposium on Information Systems*, pages 45–52.

Balabko, P., Atkinson, C., and Tucci, C. (2005). *Situation-Based Modeling Framework for Enterprise Architecture*. PhD thesis, École Polytechnique Fédérale de Lausanne (EPFL).

Beerepoot, I., Di Ciccio, C., Reijers, H. A., Rinderle-Ma, S., Bandara, W., Burattin, A., Calvanese, D., Chen, T., Cohen, I., Depaire, B., et al. (2023). The biggest business process management problems to solve before we die. *Computers in Industry*, 146:103837.

Candido, D., Lima, J., Oliveira, H., and Costa, M. (2024). An annotated dataset for automatic extraction of entities and restrictions from business process models. In *Anais do XXI Encontro Nacional de Inteligência Artificial e Computacional*, pages 978–989, Porto Alegre, RS, Brasil. SBC.

Costa, M. B. and Tamzalit, D. (2017). Recommendation patterns for business process imperative modeling. In *Proceedings of the Symposium on Applied Computing*, SAC '17, page 735–742, New York, NY, USA. Association for Computing Machinery.

De Marneffe, M.-C. and Manning, C. D. (2008). The stanford typed dependencies representation. In *Coling 2008: proceedings of the workshop on cross-framework and cross-domain parser evaluation*, pages 1–8.

de Santana Candido, D., Alves de Oliveira, H. T., and Costa, M. B. (2025). Unveiling business processes control-flow: Automated extraction of entities and constraint relations from text. In *Proceedings of the 27th International Conference on Enterprise Information Systems - Volume 2: ICEIS*, pages 771–782. INSTICC, SciTePress.

Dumas, M., Rosa, M. L., Mendling, J., and Reijers, H. A. (2018). *Fundamentals of Business Process Management*. Springer Berlin Heidelberg, 2 edition.

Ferreira, R. C. B. and Thom, L. H. (2016). Uma abordagem para gerar texto orientado a processo a partir de texto em linguagem natural. In *Simpósio Brasileiro de Sistemas de Informação (SBSI)*, pages 585–588. SBC.

Friedrich, F., Mendling, J., and Puhlmann, F. (2011). Process model generation from natural language text. In Mouratidis, H. and Rolland, C., editors, *Advanced Information Systems Engineering*, pages 482–496, Berlin, Heidelberg. Springer Berlin Heidelberg.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2023). Large language models are zero-shot reasoners.

Kourani, H., Berti, A., Schuster, D., and van der Aalst, W. M. (2024). Process modeling with large language models. In *International Conference on Business Process Modeling, Development and Support*, pages 229–244. Springer.

Licardo, J. T., Tanković, N., and Etinger, D. (2024). A method for extracting bpmn models from textual descriptions using natural language processing. *Procedia Computer Science*, 239:483–490. CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2023.

Motta, E., Rajan, T., and Eisenstadt, M. (1989). A methodology and tool for knowledge acquisition in keats-2. In *Studies in Computer Science and Artificial Intelligence*, volume 5, pages 297–322. Elsevier.

Nivon, Q. and Salaün, G. (2025). Automated generation of bpmn processes from textual requirements. In Gaaloul, W., Sheng, M., Yu, Q., and Yangui, S., editors, *Service-Oriented Computing*, pages 185–201, Singapore. Springer Nature Singapore.

Pedroso, B., Pereira, M., and Pereira, D. (2025). Performance evaluation of llms in the text-to-sql task in portuguese. In *Anais do XXI Simpósio Brasileiro de Sistemas de Informação*, pages 260–269, Porto Alegre, RS, Brasil. SBC.

Pesic, M., Schonenberg, H., and Van der Aalst, W. M. (2007). Declare: Full support for loosely-structured processes. In *11th IEEE international enterprise distributed object computing conference (EDOC 2007)*, pages 287–287. IEEE.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks.

Silva de Castro, D., Fantinato, M., and Costa, M. B. (2025). Business process design support with automated interviews. In *Proceedings of the 27th International Conference on Enterprise Information Systems - Volume 2: ICEIS*, pages 734–745. INSTICC, SciTePress.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. (2020). Mpnet: Masked and permuted pre-training for language understanding.

van der Aa, H., Di Ciccio, C., Leopold, H., and Reijers, H. A. (2019). Extracting declarative process models from natural language. In Giorgini, P. and Weber, B., editors,

Advanced Information Systems Engineering, pages 365–382, Cham. Springer International Publishing.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.