A model-driven LLM-assisted approach for generating configuration questionnaires from business process family models

Daniel Calegari*†, Andrea Delgado*
*Instituto de Computación, Facultad de Ingeniería, Universidad de la República
†Universidad ORT Uruguay
Montevideo, Uruguay
calegari@ort.edu.uy, adelgado@fing.edu.uy

***Abstract*—Variability models define configuration spaces in which stakeholders must make decisions before deriving concrete variants. Business process (BP) families constitute a representative application domain in which processes vary according to business requirements and contextual constraints. Model-driven approaches support the specification of BP family models and the derivation of variants through model transformations; however, the increasing number and complexity of variability decisions make configuration difficult for end users. This paper explores how Large Language Models (LLMs) can support the generation of end-user questionnaires from BP family models. In our approach, questionnaires are treated as higher-level configuration models that capture user-oriented decisions and provide configuration information to automate variant derivation. The contribution is threefold: a metamodel-guided LLM generation pipeline for questionnaire-based configuration modeling; an exploratory empirical assessment of the generated models; and lessons on human-in-the-loop validation of AI-generated models. The results show the feasibility of combining model-driven engineering and LLMs for configuration support, while also highlighting current limitations that must be addressed before deployment in industrial settings.**

***Index Terms*—variability, business process family, model-driven engineering, large language models**

I. Introduction

Business process variability [1] refers to contextual variations within a business process that lead to the derivation of different process variants belonging to a business process family (BP family). Typically, a BP family defines a common base process together with variability points that must be configured to derive a concrete variant. In practice, the configuration is often performed manually, requiring users to decide among variants based on informal guidelines and domain expertise.

Model-Driven Engineering (MDE) has been proposed as a viable approach for representing BP families and automating variant derivation through model transformations [2]. In this setting, variability is explicitly represented in a BP family model, while configuration information is used as input to model transformations that automatically generate concrete process variants. This approach improves automation and consistency during process customization.

However, the potentially large number of variants poses significant challenges for end users. Questionnaire-based approaches [3] have been proposed to guide users through configuration by capturing their decisions via user-oriented questions. From a requirements engineering perspective, the questionnaires can be interpreted as user-facing decision models that elicit configuration requirements and maintain traceability to variability elements and constraints in the underlying family model. Nevertheless, constructing such questionnaires remains a complex modeling task. As BP families evolve, maintaining these questionnaires also becomes increasingly costly and error-prone. One might attempt to automate questionnaire construction with a rule-based model transformation, as is done for variant derivation. Yet a questionnaire is not uniquely determined by a family model: many valid questionnaires exist, and the choices that make one usable are design decisions not captured by the underlying BP family. A deterministic generator would thus either fix these choices arbitrarily or require them to be specified by hand in advance.

Recent advances in Large Language Models (LLMs) suggest new opportunities, as they can generate human-like text, reason, and assist with knowledge-intensive activities [4]. Questionnaire generation is thus a natural candidate for LLM assistance, in which a human-in-the-loop refines generated proposals rather than authoring them from scratch. However, fully autonomous generation could be unreliable in modeling scenarios that require consistency and constraint preservation.

This paper explores how LLMs can support the generation of end-user questionnaire models from BP family models. The proposed approach combines LLM-based generation with metamodels to guide questionnaire construction. The approach is grounded in the MDE-based approach defined in [2] and generates questionnaire models in line with the Quaestio proposal introduced in [3]. Rather than replacing human modelers, the approach is framed as AI-enabled collaborative modeling, in which experts validate and refine the generated models.

The contribution is threefold: (i) a metamodel-guided prompting pipeline that maps BP-family variability models to

Partially supported by the project"Ingeniería dirigida por modelos para la especificación y configuración de familias de procesos", funded by project CSIC I+D 2022 "22520220100018UD", Uruguay.

questionnaire-based configuration models; (ii) an exploratory evaluation of how different prompt inputs and scopes affect the generated questionnaire structures; and (iii) lessons learned regarding human-in-the-loop validation of AI-generated models.

The rest of the paper is structured as follows. Section II discusses related work. Section III presents the existing MDE-based approach. Section IV introduces the LLM-assisted pipeline, while Section V presents its experimental evaluation. Section VI discusses the findings. Section VII analyzes threats to validity. Section VIII concludes and outlines future work.

II. Related work

Recent works highlight the growing role of LLMs in model-driven contexts, e.g., identifying open challenges and outlining a research agenda [5], introducing an assistant that accelerates the development of DSL semantics [6], and generating domain models from natural language descriptions [7].

Works, such as [4], explore LLMs in the context of business process management. In [8], the authors introduce the first benchmark evaluating seven closed- and open-source LLMs on BPM problems. They demonstrate that prompt construction is crucial for achieving high accuracy. They tested three prompt engineering strategies: few-shot examples, personas, and step-by-step prompts. Their results indicate that a few-shot example gives the best overall balance of accuracy and token cost. At the same time, persona prompts help in classification tasks by framing the model as a BPM specialist.

The problem addressed in this paper is closely related to research on software product lines (SPL). SPL engineering represents families of related systems using variability models, most commonly feature models. Interactive configuration approaches guide users through a sequence of decisions while ensuring that selected features remain consistent with model constraints [9]. Knowledge-based configuration systems further support users by recommending decisions and explaining constraints during the configuration process [10]. More recently, the Universal Variability Language (UVL) has been proposed as a unified textual language for representing variability models [11]. While these approaches focus on assisting users during configuration, they typically assume that the configuration or questionnaire has been manually designed.

Closer to our work, [12], [13] explore the use of LLMs to generate tailored questionnaires from contextual information and study their impact on user perception. However, these approaches are not connected to model-driven engineering or variability modeling contexts. In the BPM domain, [14] proposes the automatic generation of questionnaires to determine declarative process variants to execute at runtime. Although their approach relies on a configurable model of declarative processes, it differs from the notion of a process family used in [3]. Another line of work derives configurable process models from execution data, e.g., by discovering them from collections of event logs [15]. The variants are inferred from execution traces, and no questionnaire is involved, as the configuration space is predefined by the discovered model. In contrast, our work investigates how LLMs can support the construction of questionnaire structures from BP-family variability models. Rather than supporting users during configuration, the goal is to support the design phase of configurations.

This positions our work at the intersection of LLM-assisted modeling, model-driven configuration, and requirements elicitation through structured decision models.

III. Model-driven BP families management

In [2], the authors introduced an MDE-based approach for managing BP families, which defines a metamodel that conceptualizes key elements of BP families, and a high-level management process supported by the Business Process Family Manager (BPFM) tool[1], which allows many variability approaches to coexist in a homogeneous way. The BPFM metamodel (Figure 1) provides an approach-independent conceptualization of a BP family.

A business process family (`BPFamily`) groups multiple process variants (`ProcessVariant`) that share a base process (`BaseProcess`) while allowing variations (`VariationPoint`). These variation points determine where different variants (`Variant`) can be applied based on the context (`Context`), which defines context requirements (`ContextRequirement`) for variant selection. A specific process variant is derived by specifying a `Configuration`.

As an example, consider the order-fulfillment BP family, from [3], derived from the Voluntary Inter-industry Commerce Standard (VICS). The family is designed to support business transactions among suppliers, buyers, and logistics providers. The process supports multiple business functions, including Product Merchandising, Ordering, Logistics, and Payment. Logistics is divided into sub-phases that cover the entire shipping lifecycle, from negotiating freight quotes to handling post-delivery claims. In this context, business functions, sub-phases, and roles can be considered `Variants`, and the model incorporates `VariationPoints` that enable organizations to tailor the process into a concrete `ProcessVariant`. Moreover, there are several `ContextRequirement` items; for example, if the process manages loss or damage claims, there must be one role selected to act as the manager for them.

The general approach is depicted in Figure 2. It involves a process engineer creating a BPFM-based model for the BP family, along with a questionnaire that conforms to a questionnaire's metamodel (QML) to guide users through the configuration process. As will be explored, LLMs could assist in constructing and refining the questionnaires. A final user configures a process variant assisted by the questionnaire, producing a configuration model. Variant generation could be partially or fully automated via model transformations, using the former BP family and the configuration models.

The questionnaire-based Quaestio approach [3] is used to capture variability independently of specific languages. A questionnaire structures the space of possible alternatives as a set of questions, guiding users in defining process variants.

[1]Business Process Family Manager (BPFM): https://gitlab.fing.edu.uy/open-coal/bpfm

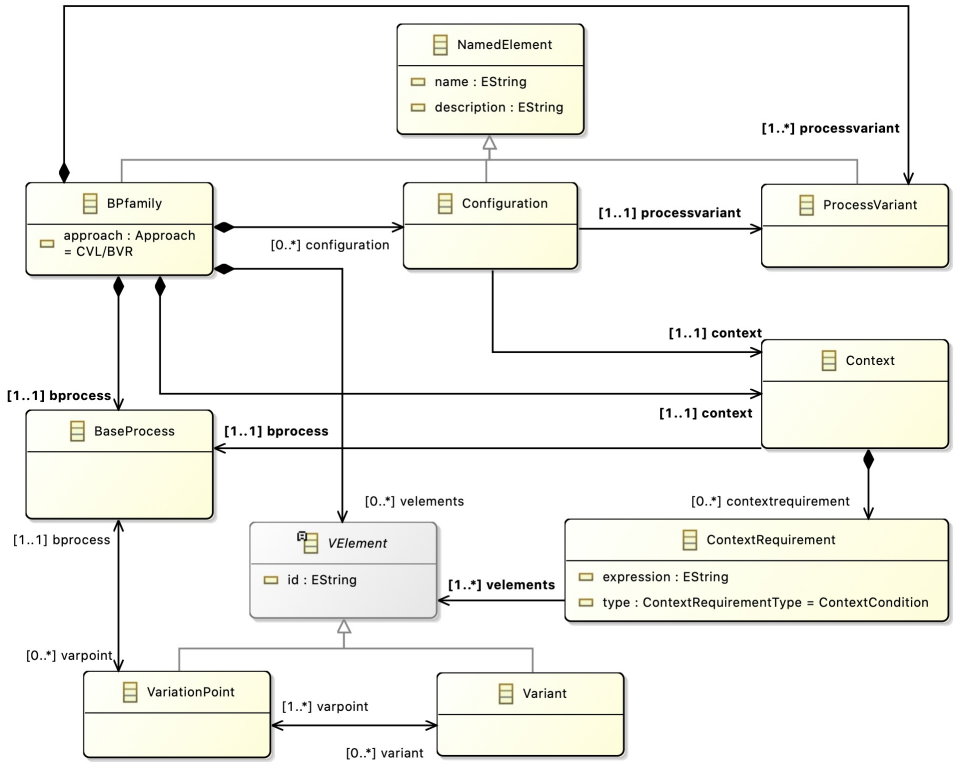

Figure 1: BP families metamodel (BPFM [2])

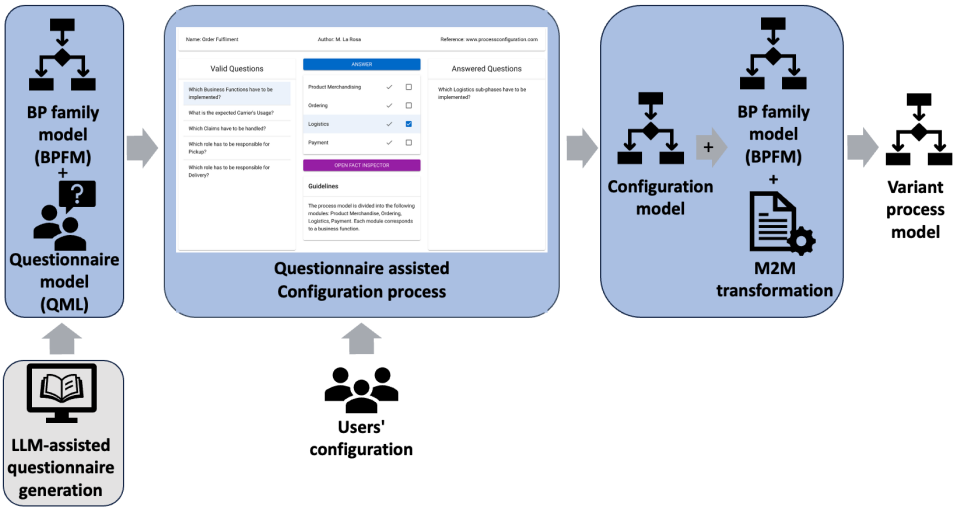

Figure 2: Approach for LLM-supported MDE-based process variant generation

Questions refer to facts that capture the system's variability, and both can be connected through dependencies and subject to domain constraints. Constraints are propositional logic expressions over a domain-specific language. Dependencies can be defined at the question level to control the questionnaire flow. A full dependency means that a question is only relevant after another has been answered. In contrast, a partial dependency means that a question conditions only part of another, for example, only some of its facts or answer options.

For example, Figure 3 depicts a questionnaire for the VICS BP family. The questionnaire's default settings implement all business functions and logistics sub-phases, support Truckload (TL) shipments, and assign Pickup and Delivery responsibilities. There are several constraints, such as `f13 = xor(f14, f15)`, which expresses that there must be one role selected to act as manager for loss or damage claims (q5), if and only if such claims were selected in q4.

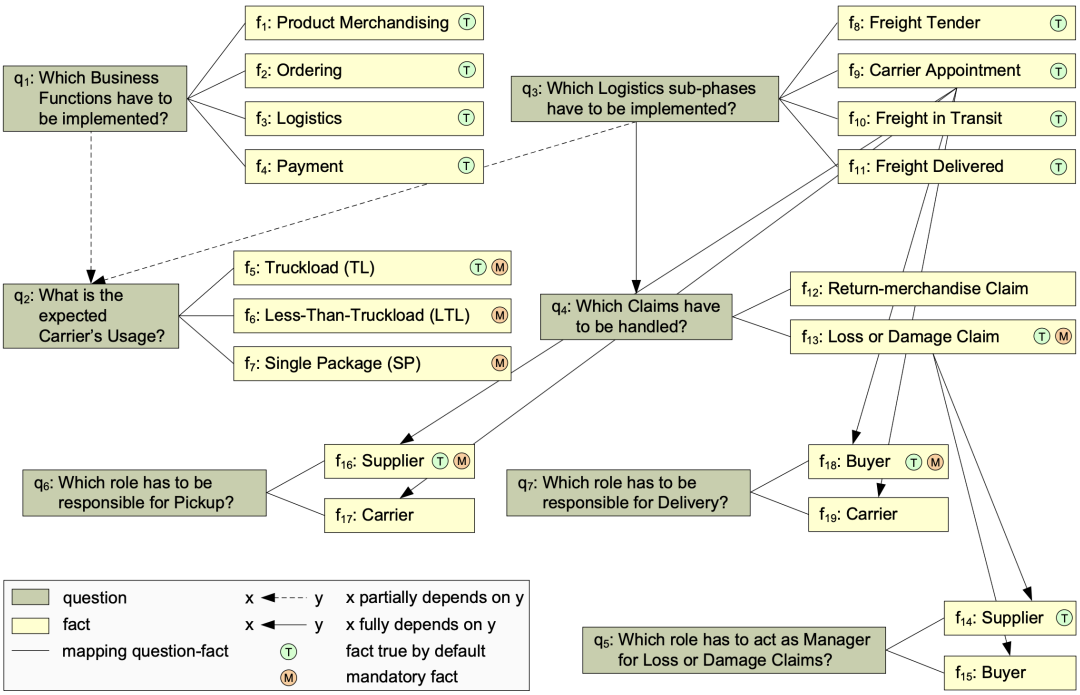

Figure 3: VICS questionnaire (from [3])

A reengineered version of the Quaestio tool[2], compatible with the MDE-based approach, enables manual graphical modeling of questionnaires, conforming to the QML metamodel in Figure 4, and guides users through a questionnaire to produce a configuration model. It uses a logical expression manipulation tool to verify in real time that answers comply with constraints, preventing inconsistent or invalid configurations.

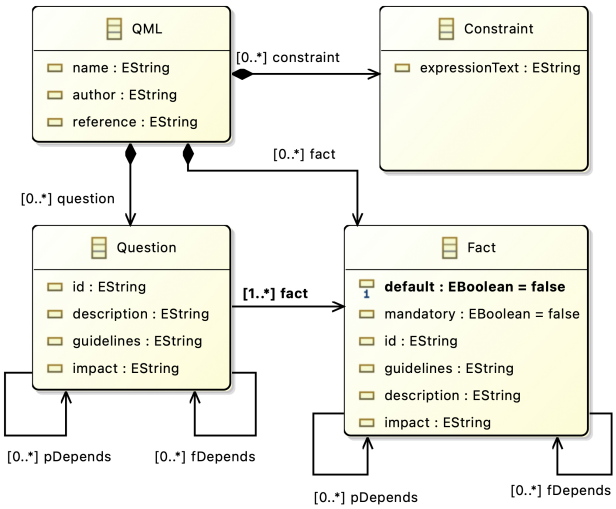

Figure 4: QML metamodel

One might expect the questionnaire itself to be produced by a deterministic model-to-model transformation. However, the same family admits many structurally different but equally valid questionnaires. A rule-based transformation can mechanically project structural elements, e.g., deriving facts from variants and constraints from context requirements, but it cannot by itself fix the decisions that distinguish a usable questionnaire from a merely valid one: how variation points and facts are grouped into the fewest sensible questions, how questions and facts are phrased for non-technical users, and which question-level dependencies should hold. These are design decisions that are not encoded in the family metamodel. A deterministic generator would therefore either commit to a single fixed policy or require the modeler to annotate the family model with these decisions in advance, thereby reintroducing the questionnaire-authoring effort the approach aims to reduce. This motivates exploring an LLM-assisted alternative that proposes the underspecified content for expert validation, rather than computing a single canonical questionnaire.

IV. LLM-ASSISTED CONSTRUCTION OF QUESTIONNAIRES

The use of LLMs to support the construction of end-user questionnaires for BP-family configuration is explored. Questionnaires are generated following the general schema shown in Figure 5, which takes a BP family abstractly specified as a BPFM conforming model and derives a QML questionnaire.

The prompt, shown in Table I, considers recommended aspects of prompt engineering [4]. Although few-shot prompting is often effective, input–output examples are intentionally avoided. Since the domain knowledge is entirely within the source model, and the prompt only specifies the genera-

[2]QuaestioMDE: https://gitlab.fing.edu.uy/open-coal/quaestiomde

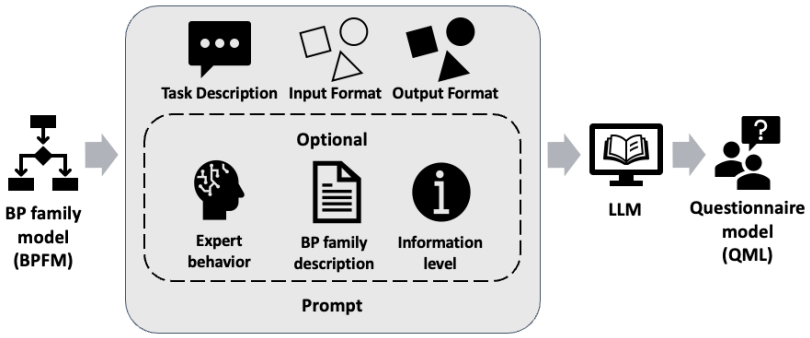

Figure 5: General schema for LLM-assisted questionnaire-model construction

tion procedure, including examples would introduce domain-specific patterns that could bias the generated questionnaires. Therefore, a zero-shot prompt design is adopted to ensure that the questionnaire content is derived solely from the inspected model. The prompt is composed of three parts:

1) **Task Description.** A clear statement of the task to be performed includes setting the context, i.e., business process variability, and optionally, asking the LLM to act as an expert on the application domain.
2) **Input Requirements.** A textual description of the documents from which the information is extracted, i.e., a model conforming to the BP families metamodel, together with the full metamodel as a reference. Adding an optional textual description of the BP family could also improve the input.
3) **Output Requirements.** A textual description of the expected output, which is a model that conforms to the QML metamodel, and the metamodel. Optionally, different levels of information to derive can be defined, such as the complete questionnaire, only the facts, or only the questions.

V. Experimentation

This section evaluates the feasibility of using LLMs to support the construction of QML questionnaire models using various prompt strategies and different levels of input information and expected output complexity. All models, prompts, and results are available online[3]. We organize the evaluation around four research questions.

- RQ1 How does enriching the prompt input affect the completeness and domain adequacy of the elicited questions and facts?
- RQ2 How does the targeted output complexity affect generation quality?
- RQ3 How do two state-of-the-art reasoning LLMs compare for the richest input configuration?
- RQ4 Does decomposing questionnaire construction into a sequence of steps improve the quality and controllability of the output?

[3]Supplementary material. https://doi.org/10.5281/zenodo.15838559

A. Experimental setup

As a first experiment, a **zero-shot strategy** is used across different test dimensions, without any feedback loop. An incremental test case is defined that focuses on the richness of the input and the required complexity of the output. These two dimensions (Table II) are already considered as optional fragments of the general prompt structure in Table I. Input requirements range from considering only the input model to evaluating a model enhanced with a textual description of the BP family, in which the LLM acts as a domain expert. For each input requirement, a second dimension, provided by the output requirements, concerns the information to be generated. These requirements require different levels of reasoning, ranging from proposing structural elements to producing an initial, complete questionnaire model for expert review. GPT 5.1 Thinking[4] model was used, OpenAI's most advanced reasoning model as of November 2025.

As a second experiment within the zero-shot strategy, **LLMs are compared based on their reasoning capabilities**. Advanced tests of I4 input requirements (Model + Description + Expert) were conducted with Gemini 3 Pro[5], Google's most advanced reasoning model as of November 2025.

As a final experiment, a **prompt chaining strategy** was employed. The GPT 5.1 Thinking model was used to determine whether it is possible to improve results through conversational interaction with the LLM, more closely resembling the usual way of working with domain experts in real environments.

For the three experiments the same input models were used: the family model of the VICS example introduced in Section III, and the "Unborn child acknowledgment" process, described in [16], enacted when a man wants to register that he will be the father of a child still to be born while he is not married to his pregnant partner. The model representing the BP family conforms to the BPFM metamodel (Figure 1) and to the textual descriptions of the BP family in the papers introducing these examples. As ground truth, the existing human-built questionnaires extracted from the Quaestio tool and validated with stakeholders were used. Although the Unborn BP family has a simpler structural definition (7 variation points and 13

[4]OpenAI GPT 5 System Card: https://openai.com/index/gpt-5-system-card/
[5]Gemini model family: https://deepmind.google/models/gemini/

Table I: General prompt structure (excerpt)

Task Description:

[*OPTIONAL*: Act as an expert on business process variability and the `XXX` application domain.] You are provided with an XMI file (conforming to bpfm.ecore) describing a business process family. Your task is to generate an XMI document (conforming to qml.ecore) containing questions to help non-technical users select process variants for the available variation points.

Input Requirements:

The input contains `<velements>` elements with the following attributes:

- `xsi:type`: variation point `bpfm:VariationPoint` or variant `bpfm:Variant`,
- `id`: the name of the variation point or variant,
- `variant` (optional): the variants that can be selected for such a variation point,
- `varpoint` (optional): the variation points to whom the variant applies.

It also contains `<contextrequirement>` elements defining a context requirement that limits the possible combinations of variation points and variants, with attributes:

- `velements`: a reference to the variation points and variants involved,
- `expression`: the context requirement expression [...]
- `type` (optional): a context condition, a constraint, or a guideline [...]

[*OPTIONAL*: Consider the following description of the business process family...]

Output Requirements:

The output should contain a set of questions and facts that represent the space of possible answers to those questions. Each `<question>` has the following attributes:

- `id`: a unique identification among the questions,
- `description`: it is a concrete question to ask,
- `fact`: it refers to the set of facts that correspond to the question [...]
- `guideline` (optional): it explains the question and its related facts.
- `pDepends` (optional): it references other questions [...] (partial dependency) [...],
- `fDepends` (optional): it references other questions [...] (full dependency) [...].

Each `<fact>` element has the following attributes:

- `id`: a unique identification among the facts,
- `description`: it gives a name to the fact,
- ...

The resulting model must be targeted at non-technical end users. Ensure that the output is well-formed XMI, conforms to qml.ecore, and preserves the input's structure.

Table II: Test dimensions for validating questionnaire generation

#	Input requirements	Purpose
I1	Model	Evaluate if the questionnaire strictly follows the model without adding external information.
I2	Enhanced (Model + Description)	Evaluate if the description improves the clarity and context of the questionnaire.
I3	Model + Expert	Evaluate if the LLM acting as an expert enriches the questionnaire with relevant additional information.
I4	Enhanced Model + Expert	Evaluate the combined effect of structured data, textual description, and expert knowledge.

#	Output requirements	Description
O1	Structural elements	Involves defining facts and questions with their information, but without dependencies between them.
O2	Logical connections	Involves building logical connections, i.e., to define the facts, and the dependencies between them.
O3	Full discovery	Involves building the complete questionnaire, including dependencies among questions.

variants), and these elements do not depend on one another, unlike VICS (19 variation points and 20 variants), it has more complex context requirements that include information on possible combinations of gateways, without direct connections to dependencies or mandatory facts as found in VICS.

To evaluate the results, quantitative metrics were calculated for precision, recall, F1, and accuracy. Accuracy denotes a positive-class agreement metric[6], excluding true negatives; it is therefore equivalent to the Jaccard index. This is appropriate since the context does not provide meaningful true negatives, because the universe of potentially generated questionnaire elements is unbounded. Metrics are based on the semantic equivalence between the ground-truth model's questions, facts, and dependencies and those of the derived models. Semantic equivalence is determined by assessing whether two structural elements have similar meanings, i.e., whether their names refer to the same concept and their descriptions indicate the same role within the model, while allowing only minor lexical variations (e.g., synonyms). To reduce subjectivity, the evaluation was performed independently by two reviewers, and disagreements were resolved through discussion. A partial-credit scheme was used, which explains the fractional scores:

- O1: +1 for each element (question and fact) that is present in both models.
- O2: +0.5 for each fact that is present in both models, plus 0.5 if their dependencies match those in the ground-truth.
- O3: +0.5 for each element that is present in both models, plus 0.5 if their dependencies match those in the ground-truth.

Regarding O2 and O3, additional dependencies that were judged logically consistent but absent from the ground truth were not counted as true positives, but treated as neutral qualitative observations. Generated dependencies that were neither present in the ground truth nor judged as logically consistent were counted as false positives.

All reported results correspond to a single execution for each combination. The experiments were designed as an initial exploratory validation to assess feasibility and identify

[6]Accuracy is computed as $TP/(TP + FP + FN)$

typical weaknesses in LLM-assisted questionnaire generation, rather than as a statistically robust evaluation of output stability. The models were used under their default generation settings, without controlling generation parameters such as seed or temperature. Future replications should execute each configuration multiple times and report controlled decoding parameters whenever supported by the underlying API.

B. Zero-shot experiment

Table III shows the quantitative evaluation, which is summarized next.

1) ***Structural elements (O1)****:* In the reported executions, this was the least challenging task, with outputs generally close to the ground truth. The structural simplicity of the "Unborn" family yielded a perfect match in these executions. For VICS, relying solely on the model (I1-O1) already gave balanced precision and recall, and adding the BP-family description (I2-O1) or combining it with the expert persona (I4-O1) reproduced the reference questions and facts exactly. In contrast, enabling expert knowledge alone (I3-O1) introduced additional, unspecified elements (e.g., a default question), lowering precision and accuracy. When combined with textual information, expert knowledge yielded the highest scores for this task and introduced domain-specific terminology.

2) ***Logical connections (O2)****:* When focusing exclusively on dependencies among facts, the reported executions achieved high scores across several configurations, although these results should not be read as evidence of statistical stability. For "Unborn", there were two instances in which the LLM generated facts tied to the variation points, but not when asked to differentiate facts from questions. For VICS, the simplest setting (I1-O2) was already strong, while the variants with description and/or expert knowledge (I2-O2, I3-O2, I4-O2) remained high but slightly lower. This suggests that textual descriptions and expert knowledge can help infer candidate dependencies, but may also introduce spurious or reversed links, explaining the observed drops in precision and accuracy.

3) ***Full discovery (O3)****:* As this task requires reconstructing the complete questionnaire, it is the most challenging. For VICS without additional context (I1-O3), the model obtained moderate yet balanced scores. Adding the description (I2-O3) or the expert knowledge (I3-O3) improved both structure and relations, and combining model, description, and expert knowledge (I4-O3) reached the highest scores among the reported configurations. This suggests that a richer context may help reconstruct structure and dependencies more accurately, though additional runs would be needed to assess stability. For "Unborn, however, no question dependencies were discovered: in VICS, the family describes variant dependencies that induce question dependencies, whereas in "Unborn" these do not exist, so question dependencies are a design decision rather than a context requirement. Some of these surface only when asking for improvement suggestions (see Section V-D).

C. Exploratory LLM comparison

Table III (bottom) presents quantitative results of Gemini 3 Pro on advanced I4 input tests, using the same models as before, so it can be compared to GPT 5.1 Thinking results.

For structural element identification (I4-O1), Gemini 3 Pro obtained perfect scores in the reported execution, preserving the original elements and the relations between questions and facts in both models. Also, in the fact description, the action to which the fact refers is correctly added, e.g., in VICS, f14="Supplier acts as Claim Manager" instead of only "Supplier", f17="Carrier arranges Pickup" instead of only "Carrier". This makes the labels more explicit and prevents duplicate labels for entities such as "Supplier" and "Carrier". In the reasoning explanation, although not explicitly asked in the input prompt, the context requirements and constraints were analyzed to examine the logical relations among elements.

Moreover, in the reported Gemini 3 Pro execution, logical connections (I4-O2) preserved the correct identification of facts from the previous task in both models and correctly identified the evaluated dependencies between facts (in VICS, since there are no dependencies in "Unborn"), yielding perfect metric values in this execution. Regarding the reasoning, it added an analysis of the process logic, based on business rules, to convert constraints into dependencies.

For the full discovery (I4-O3), Gemini 3 Pro results were very similar to those of GPT 5.1, in the reported execution. They maintained the correct identification of questions and facts as before, but missed dependencies in the case of "Unborn". Although it asked about "mapping out any partial or full dependencies (pDepends/fDepends) based on the underlying constraints", which could indicate an understanding of the missing elements, they were not included in the answer, which drops all measures for "Unborn". In the case of VICS, it combines two questions into one regarding the role responsible for pickup and delivery, while also integrating the associated facts and dependencies, leaving only one missing dependency, so the metrics remain close to GPT 5.1 Thinking. It also changed question IDs from numbers to semantic grouping, e.g., "q_functions" or "q_shipment_size", which may improve their understandability.

In summary, in this exploratory comparison, Gemini 3 Pro produced similar or higher scores than GPT 5.1 Thinking on the selected advanced tasks, with the same dependency drawbacks but more explicit, domain-specific labeling. It should therefore be read as an exploratory contrast under the selected configuration, not as a benchmark.

D. Prompt chaining evaluation

In the prompt chaining evaluation, the LLM is prompted to generate one aspect at a time, and each response is then reused as input to the next prompt. The chain of prompts requires, in order: (1) fact definition, (2) question definition, (3) fact dependencies, (4) question dependencies derived from facts, (5) constraints, and (6) suggestions for improvements based on expert knowledge. Each prompt is a refined version of the general prompt in Table I, tailored to the task at hand, particularly by making explicit the information to be generated and adjusting the expected use of expert knowledge

Table III: Test metrics for the evaluation

	test	model	#Q	#F	#Q+F	TP	FP	FN	precision	recall	F1	accuracy
GPT 5.1 Thinking [structural elements]	I1-O1	vics	7	19	26	24	2	2	0.92	0.92	0.92	0.86
		unborn	6	13	19	19	0	0	1.00	1.00	1.00	1.00
	I2-O1	vics	7	19	26	26	0	0	1.00	1.00	1.00	1.00
		unborn	6	13	19	19	0	0	1.00	1.00	1.00	1.00
	I3-O1	vics	7	20	27	23	4	3	0.85	0.88	0.87	0.77
		unborn	6	13	19	19	0	0	1.00	1.00	1.00	1.00
	I4-O1	vics	7	19	26	26	0	0	1.00	1.00	1.00	1.00
		unborn	6	13	19	19	0	0	1.00	1.00	1.00	1.00
GPT 5.1 Thinking [logical connections]	I1-O2	vics	-	20	20	19	1	0	0.95	1.00	0.97	0.95
		unborn	-	19	19	13	6	0	0.68	1.00	0.81	0.68
	I2-O2	vics	-	20	20	17.5	2.5	1.5	0.88	0.92	0.90	0.81
		unborn	-	13	13	13	0	0	1.00	1.00	1.00	1.00
	I3-O2	vics	-	20	20	18	2	1	0.90	0.95	0.92	0.86
		unborn	-	13	13	13	0	0	1.00	1.00	1.00	1.00
	I4-O2	vics	-	20	20	18.5	1.5	0.5	0.93	0.97	0.95	0.90
		unborn	-	19	19	13	6	0	0.68	1.00	0.81	0.68
GPT 5.1 Thinking [full discovery]	I1-O3	vics	6	20	26	22.5	3.5	3.5	0.87	0.87	0.87	0.76
		unborn	3	13	16	16	0	3	1.00	0.84	0.91	0.84
	I2-O3	vics	8	20	28	25	3	1	0.89	0.96	0.93	0.86
		unborn	3	13	16	16	0	3	1.00	0.84	0.91	0.84
	I3-O3	vics	6	19	25	23.5	1.5	2.5	0.94	0.90	0.92	0.85
		unborn	3	13	16	16	0	3	1.00	0.84	0.91	0.84
	I4-O3	vics	7	19	26	25.5	0.5	0.5	0.98	0.98	0.98	0.96
		unborn	3.5	13	16.5	16.5	0	2.5	1.00	0.87	0.93	0.87
ground-truth		vics	7	19	26	-	-	-	-	-	-	-
		unborn	6	13	19	-	-	-	-	-	-	-
Gemini 3 Pro	I4-O1	vics	7	19	26	26	0	0	1.00	1.00	1.00	1.00
		unborn	6	13	19	19	0	0	1.00	1.00	1.00	1.00
	I4-O2	vics	-	19	19	19	0	0	1.00	1.00	1.00	1.00
		unborn	-	13	13	13	0	0	1.00	1.00	1.00	1.00
	I4-O3	vics	6	19	25	25	0	1	1.00	0.96	0.98	0.96
		unborn	3.5	13	16.5	16.5	0	2.5	1.00	0.87	0.93	0.87

in the generation process. All prompts and results are available online within the supplementary material repository.

Across both case studies, prompt chaining provided useful results, but its benefits were not uniform. When requesting exclusive identification of facts, allowing expert knowledge only for disambiguation and description yielded a perfect match with the reference facts. Question generation also matched most reference elements, although with a deviation in the VICS case study. There, the model returned a reduced set by combining the claim type and role-handling decision into a single entry, and combining the other two role questions. Although this loses precision regarding the potential combinations a final user could select, it is not entirely incorrect, since the BP family does not induce it entirely.

Dependencies were the most problematic aspect of the chain. In VICS, the model provided the complete set of full dependencies within the ground-truth model and also added reasonable new ones. However, it also introduces partial dependencies in the reverse direction, e.g., logistics facts depend partially on their sub-phases. This may be because the definition of what constitutes a dependency is currently very weak. In Unborn, the opposite problem occurred: the dependency-generation steps failed to reconstruct the questionnaire's intended sequential structure. These results suggest that dependencies are among the most difficult elements to infer because they not only encode structural relations among variability elements but also reflect design decisions about question visibility and answer filtering. This motivated the explicit distinction between full and partial dependencies in the prompt. Nevertheless, expert validation remains necessary to confirm dependency semantics and to avoid incorrect links.

Constraint generation showed a similar contrast. In VICS, the translation of context requirements into propositional constraints closely matched the reference in the reported execution. In Unborn, however, the model produced a much larger set of expressions than expected. Instead of recovering only the minimal set of constraints, it also translated a broader range of BP-family context requirements into propositional logic. This made the result richer in process semantics, but also noisier and less economical as a questionnaire model. Hence, while prompt chaining supports formalization well, it does not always control the abstraction level of constraints.

The final improvement step offered useful observations by exposing weaknesses accumulated throughout the chain and providing a basis for refinement. In VICS, the suggestions addressed issues such as mandatory flags, question granularity, ordering, dependency cleanup, and the interpretation of strong equivalences. Several of these recommendations brought the generated model closer to the ground truth, while others arguably improved the questionnaire beyond what the manually designed version did. In Unborn, the suggestions focused on aligning defaults and mandatory values with the source

model, improving terminology, adding explicit dependencies, and consolidating repeated constraints, among others. After these recommendations were applied, the final model became more faithful in several respects, although some structural limitations remained, especially regarding dependencies and the excessive number of constraints.

VI. Discussion

The results presented show that an LLM can assist in creating questionnaires for configuring BP families. However, several aspects must be addressed before such assistance can be considered production-ready.

Beyond BP-family configuration, the study suggests several lessons for LLM-assisted model-driven configuration tasks more generally. First, metamodel conformance is necessary but not sufficient: a model may conform syntactically while still containing incorrect, missing, or over-inferred semantic relations. Second, adding contextual and expert information improves readability and domain adequacy, but also increases the risk of generating elements not supported by the source model, e.g., questions inferred from textual descriptions that are not present in the BP-family specification. Third, dependencies and constraints are especially difficult, since they do not always follow from structural variability elements but may encode design decisions about question ordering, visibility, and answer filtering. As a result, the LLM tends to omit dependency edges or produce incomplete or reversed ones, and to miss questions for variability elements that are weakly described in the model, so that logical completeness cannot be guaranteed, and post-processing is often required to repair or complete the generated dependencies. Finally, generated models should be treated as intermediate design artifacts requiring both automated validation and expert review before use in configuration workflows. While some over-inferred elements may represent reasonable interpretations, others can yield inconsistent or overly complex configurations, which motivates stronger prompt guidance and validation mechanisms.

Requests about which aspects should be generated must be explicit; for example, since guidelines are optional, the reasoner may choose not to include them, even though they can be useful to experts during questionnaire design. It is also important to add guardrails that prevent the LLM from overstepping its bounds. For example, experiments I1-O3 and I3-O3 added constraints that were not required by the prompt but were valid elements of the output metamodel. Since no instructions were added for building well-formed constraints, these results are not compliant. This behavior can be avoided by adding a guardrail specifying that the LLM not generate any elements other than those requested. In general, the generated models conformed to the QML metamodel, with minor errors in handling XMI references but no other syntactical errors. These errors can be easily corrected by applying a fixing loop based on feedback from an XMI compliance checker.

Combining the BP-family description with expert input improves question discovery and the quality of descriptions. As the complexity of the family model increases, this contextual information becomes even more important. However, overly rich contextual knowledge may overspecify the task and lead the LLM to generate facts or questions not present in the BP-family specification, as observed with GPT 5.1 Thinking. This issue could be mitigated by more precisely defining the scope of expert reasoning or by more carefully structuring prompts. More systematic prompt engineering techniques, such as chain-of-thought scaffolding or RAG, could further improve the quality of generation.

The BPFM metamodel maximizes generality, but it may also deprive the LLM of the richer semantics available in concrete modeling notations such as BPMN. Hard-coding a specific syntax would fragment the approach or require an expanded hybrid metamodel capable of representing imperative, declarative, and event-driven constructs.

Since questionnaires may also include elements not explicitly present in the BP-family model (for example, dependencies between questions that do not directly correspond to dependencies between facts), human-in-the-loop validation plays a key role. In practice, the prompt chaining strategy could be viewed as a structured elicitation and refinement process in which the LLM acts as a design assistant, accelerating questionnaire construction while leaving responsibility for correctness and final approval to domain experts, ensuring correct dependency semantics, appropriate constraint abstraction, and usable questionnaire design. Corrections can then be incorporated before the questionnaire is finalized.

From a scalability perspective, our experiments inject the entire BP-family model into a single prompt. Large industrial repositories may exceed current context-window limits, which could be mitigated by decomposing models into smaller segments and applying prompt chaining to process them incrementally. Similarly, the questionnaire design problem can be decomposed into smaller subtasks, rather than delegating the entire reasoning chain to the LLM.

Finally, current inference latency (several minutes per prompt with GPT 5.1 Thinking) remains acceptable for offline design but not for interactive configuration scenarios. Until faster generation and more reliable automated validation become available, human validation and automated conformance checks will remain necessary within the overall workflow.

VII. Threats to validity

Construct validity: The evaluation relies on manually created questionnaires that serve as the ground truth. However, questionnaire design is inherently subjective, and multiple structurally different questionnaires may represent valid configurations of the same BP family. The reference questionnaire should therefore be interpreted as an expert-designed baseline rather than the only correct solution. Consequently, some elements generated by the LLM that do not match the reference may still represent reasonable alternative formulations. The reported metrics measure conformance to this particular reference design rather than the entire space of valid questionnaires.

Internal validity: LLM outputs are sensitive to prompt design. Our evaluation relies on a specific prompt structure

and ordering of information. During the study's development, a limited number of prompt refinements were made to improve clarity and reduce obvious inconsistencies in the results. However, these adjustments were not conducted through a systematic prompt optimization or sensitivity analysis. As a result, the reported results may depend to some extent on the chosen prompt formulation. A more rigorous investigation of prompt robustness through controlled variations in wording, structure, and information ordering is left for future work.

External validity: The study relies on simple industrial cases recovered from the literature. While these cases provide realistic scenarios, they limit the generalizability of the results. Different domains, modeling paradigms, or model sizes may present different challenges for questionnaire generation. Future studies should therefore evaluate the approach across multiple BP families and application domains.

Conclusion validity: The reproducibility of the results is affected by the characteristics of the LLMs. Proprietary LLMs are non-deterministic and continuously updated. In this study, each configuration was executed once, so the reported metrics should be interpreted as an initial observation of the models' behavior rather than as statistically stable estimates. This limits the strength of conclusions regarding the relative performance of the models and the robustness of specific prompt configurations. In addition, generation parameters such as temperature, sampling settings, and seeds were not systematically varied or controlled. The models were used under default settings to observe the kinds of weaknesses that arise in an unaltered, standard usage scenario. However, this choice may have influenced hallucinations and, consequently, the reported metrics. Another potential threat is the subjectivity involved in the evaluation; this was mitigated by having two reviewers independently assess the generated models and resolve disagreements through discussion.

VIII. Conclusions

This paper explored how LLMs can assist in constructing and refining user-oriented questionnaire models to support the configuration process for BP families. In this setting, questionnaires can be understood as higher-level configuration models that elicit end-user decisions and map them to variability elements in the underlying model. The proposed approach can be extended beyond BPM to other contexts requiring similar configuration mechanisms, particularly those involving variability management and user-guided configuration processes supported by structured questionnaires.

Despite the inherent non-determinism of LLM outputs, our findings suggest that LLMs can serve as useful assistants rather than fully autonomous generators: their main benefit lies in accelerating the initial construction of questionnaire structures and suggesting improvements, while expert review remains necessary for correctness, as discussed in Section VI. In practice, the approach is best suited for offline design phases, where domain experts iteratively refine the generated questionnaire before deployment.

Further evaluation and refinement are needed for industrial reliability. This includes broader benchmarking across LLMs, domains, and model sizes, as well as user-oriented measures such as comprehensibility and perceived usefulness. Future work should also study prompt engineering and knowledge-loading strategies that better constrain the generation process.

Acknowledgment

We thank Marcello La Rosa for providing us with the Quaestio source code and Marcelo Rodríguez for working on the QuaestioMDE implementation.

References

[1] M. La Rosa, W. M. P. van der Aalst, M. Dumas, and F. Milani, "Business process variability modeling: A survey," *ACM Comput. Surv.*, vol. 50, no. 1, pp. 2:1–2:45, 2017.

[2] A. Delgado, D. Calegari, F. García, and B. Weber, "Model-driven management of BPMN-based business process families," *Softw. Syst. Model.*, vol. 21, no. 6, pp. 2517–2553, 2022.

[3] M. La Rosa, W. M. P. van der Aalst, M. Dumas, and A. H. M. ter Hofstede, "Questionnaire-based variability modeling for system configuration," *Softw. Syst. Model.*, vol. 8, no. 2, pp. 251–274, 2009.

[4] M. Grohs, L. Abb, N. Elsayed, and J. Rehse, "Large language models can accomplish business process management tasks," in *Business Process Management Workshops 2023, Revised Selected Papers*, ser. LNBIP, vol. 492. Springer, 2023, pp. 453–465.

[5] J. Di Rocco, D. Di Ruscio, C. Di Sipio, P. Nguyen, and R. Rubei, "On the use of large language models in model-driven engineering," *Softw. Syst. Model.*, vol. 24, no. 3, pp. 923–948, 2025.

[6] C. Di Sipio, R. Rubei, J. Di Rocco, D. Di Ruscio, and L. Iovino, "On the use of LLMs to support the development of domain-specific modeling languages," in *ACM/IEEE 27th Intl. Conf. on Model Driven Eng. Lang. and Sys.* NY: ACM, 2024, p. 596–601.

[7] Y. Yang, B. Chen, K. Chen, G. Mussbacher, and D. Varró, "Multi-step iterative automated domain modeling with large language models," in *ACM/IEEE 27th Intl. Conf. on Model Driven Eng. Lang. and Systems*. NY: ACM, 2024, p. 587–595.

[8] K. Busch and H. Leopold, "Towards a benchmark for large language models for business process management tasks," *CoRR*, vol. abs/2410.03255, 2024.

[9] M. Janota, G. Botterweck, R. Grigore, and J. Marques-Silva, "How to complete an interactive configuration process?" in *SOFSEM 2010: Theory and Practice of Computer Science*. Berlin: Springer, 2010, pp. 528–539.

[10] A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen, *Knowledge-Based Configuration: From Research to Business Cases*. Morgan Kaufmann, 2014.

[11] D. Benavides, C. Sundermann, K. Feichtinger, J. A. Galindo, R. Rabiser, and T. Thüm, "UVL: Feature modelling with the universal variability language," *Journal of Systems and Software*, vol. 225, p. 112326, 2025.

[12] Y. Kim, J. Lee, J. H. Han, M. Kim, H. Lee, and W. H. Lee, "Agentic LLM workflows for personalized user experience questionnaire generation," in *2024 IEEE Intl. Conf. on Consumer Electronics-Asia (ICCE-Asia)*, 2024, pp. 1–4.

[13] Z. Zou, O. Mubin, F. Alnajjar, and L. Ali, "A pilot study of measuring emotional response and perception of LLM-generated questionnaire and human-generated questionnaires," *Scientific Reports*, vol. 14, no. 1, p. 2781, 2024.

[14] A. Jiménez, I. Barba, B. Weber, and C. Del Valle, "Automatic generation of questionnaires for supporting users during the execution of declarative business process models," in *Proc. of 17th Intl. Conf. Business Inf. Systems*, ser. LNBIP, vol. 176. Springer, 2014, pp. 146–158.

[15] J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, "Mining configurable process models from collections of event logs," in *11th Intl. Conf., BPM 2013. Proceedings*, ser. LNCS, vol. 8094. Springer, 2013, pp. 33–48.

[16] F. Gottschalk, T. A. C. Wagemakers, M. H. Jansen-Vullers, W. M. P. van der Aalst, and M. La Rosa, "Configurable process models: Experiences from a municipality case study," in *Advanced Information Systems Engineering, 21st Intl. Conf., CAiSE 2009. Proceedings*, ser. LNCS, vol. 5565. Springer, 2009, pp. 486–500.