16th International Model-Driven Requirements Engineering

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

Daniel Calegari[1,2] . Andrea Delgado[2]

[1] Universidad ORT Uruguay | **calegari@ort.edu.uy**

[2] Universidad de la República, Uruguay

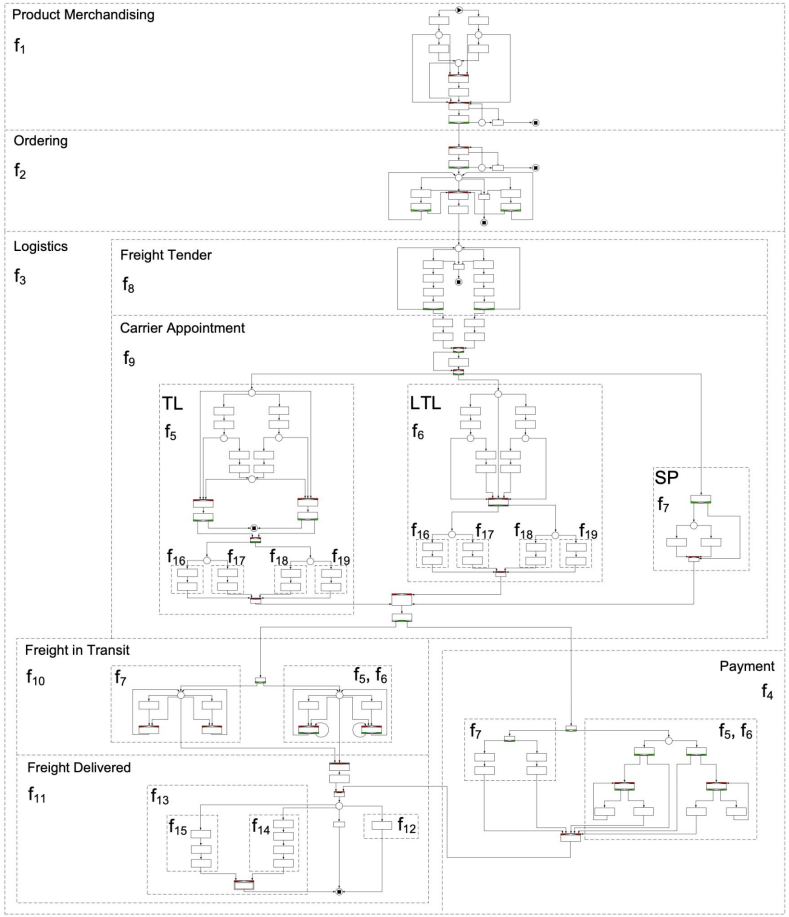
UNIVERSIDAD ORT Uruguay

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

context

Business process variability refers to contextual variations within a business process that lead to the derivation of different process variants belonging to a **business process family**.

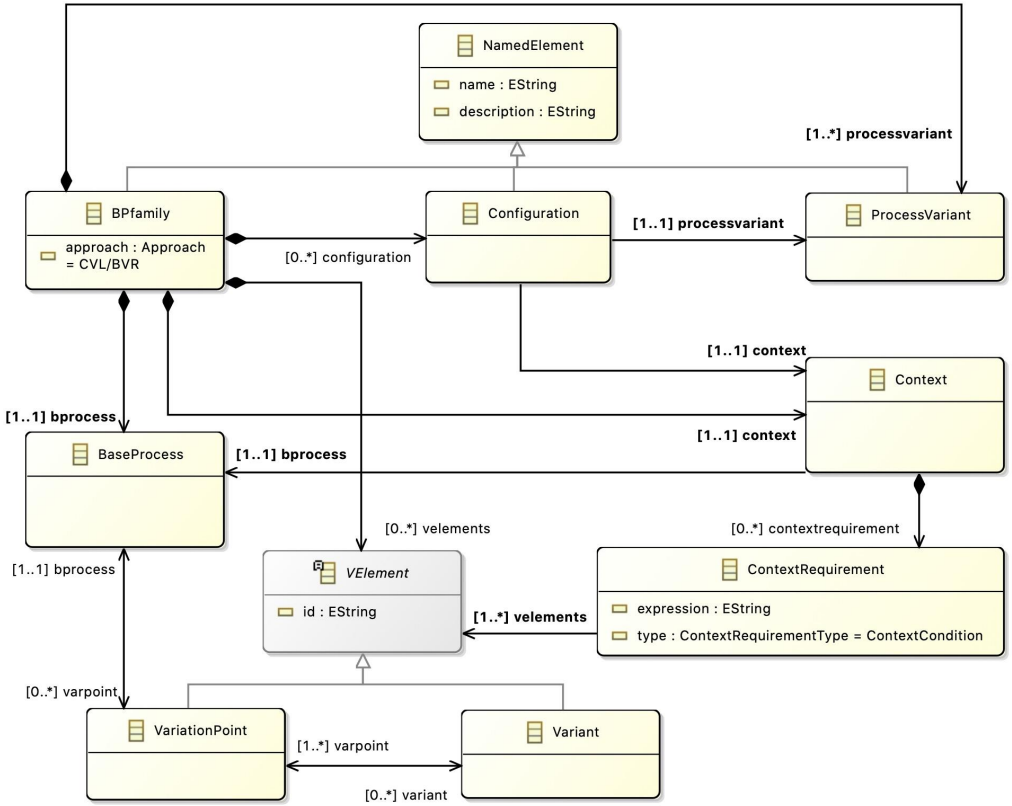
A BP family defines a common base process together with variability points that must be configured to derive a concrete variant.



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.

context

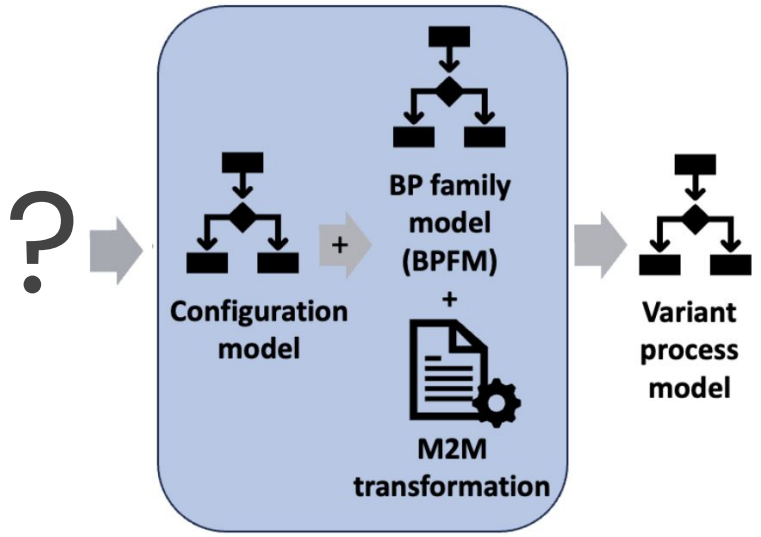
Model-Driven Engineering (**MDE**) has been proposed as a viable approach for **representing BP families** and automating variant derivation through model transformation



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

context

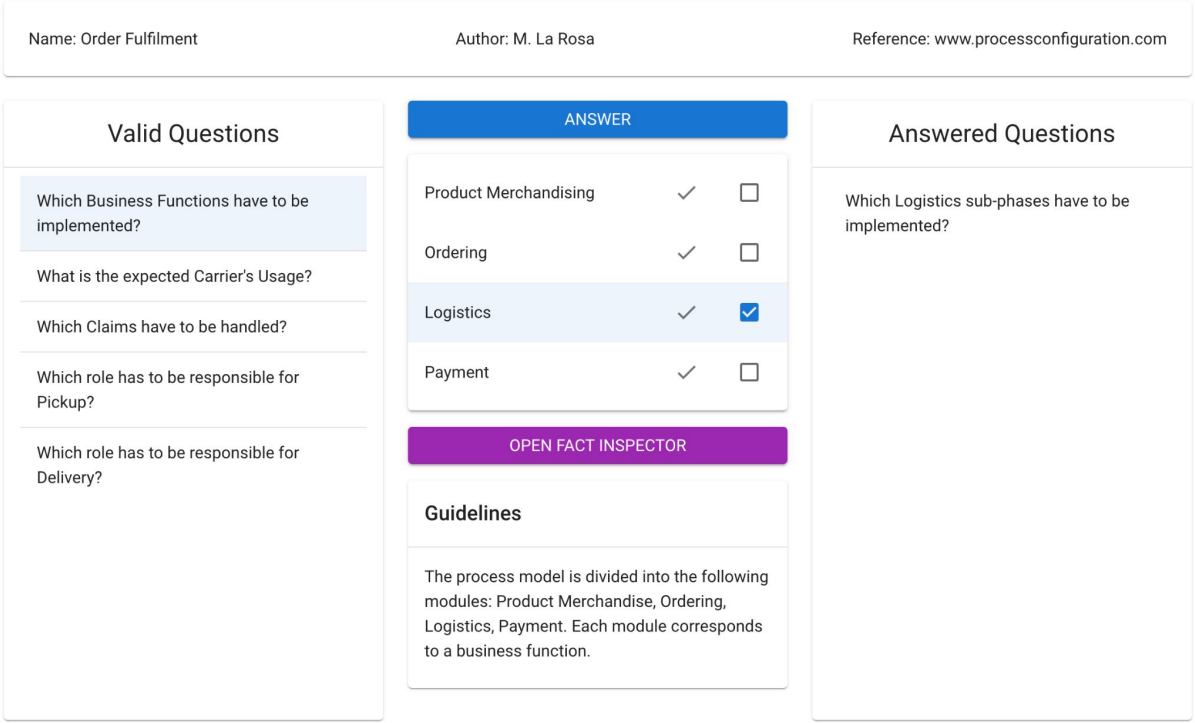
Model-Driven Engineering (**MDE**) has been proposed as a viable approach for representing BP families and **automating variant derivation** through model transformation



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

context

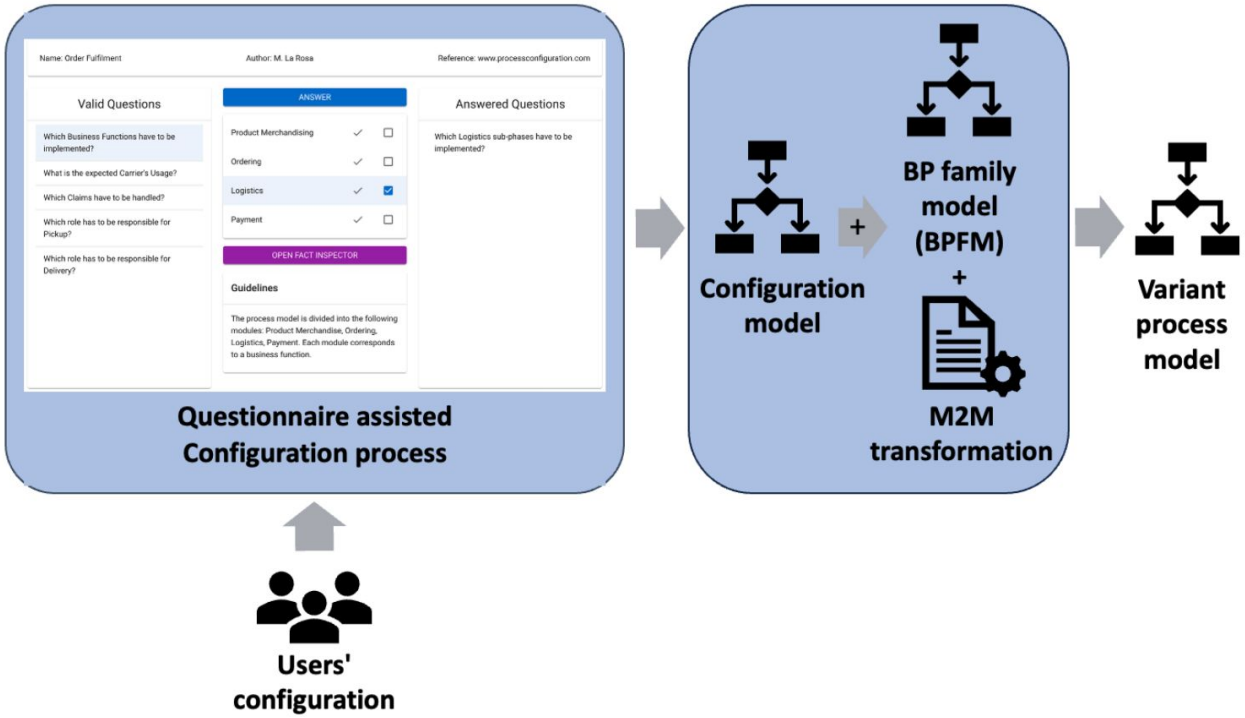
Quaestio has been proposed to **guide users through configuration** by capturing their decisions via user-oriented questions.



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.

context

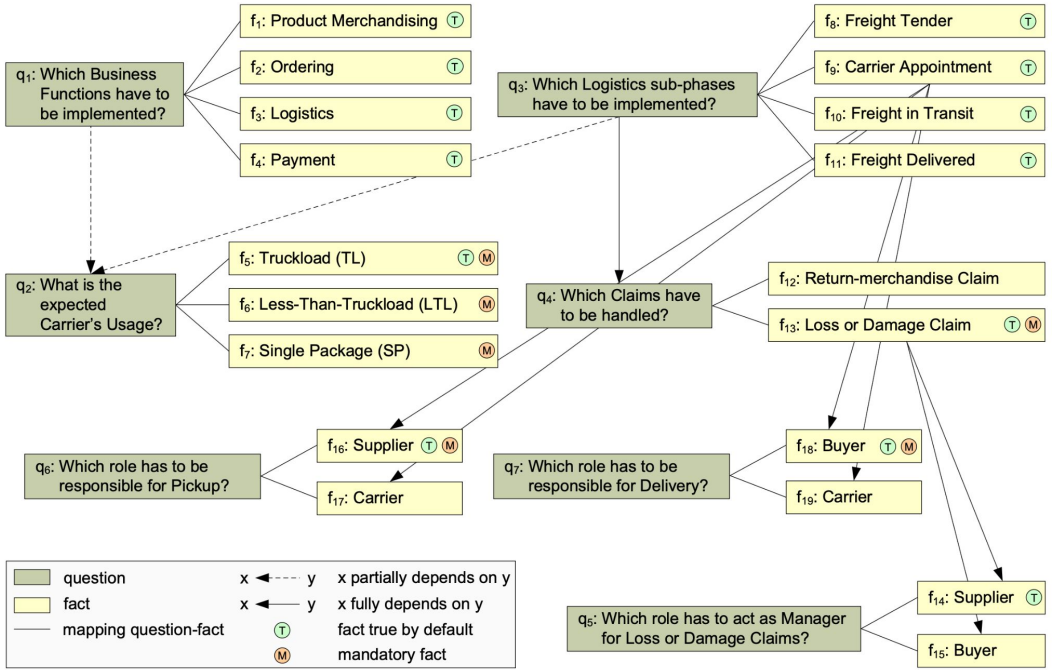
Answering a **questionnaire** generates a **configuration model** that feeds the automatic variant derivation



problem definition

however, constructing and maintaining such questionnaires remains a **complex, costly and error-prone task**.

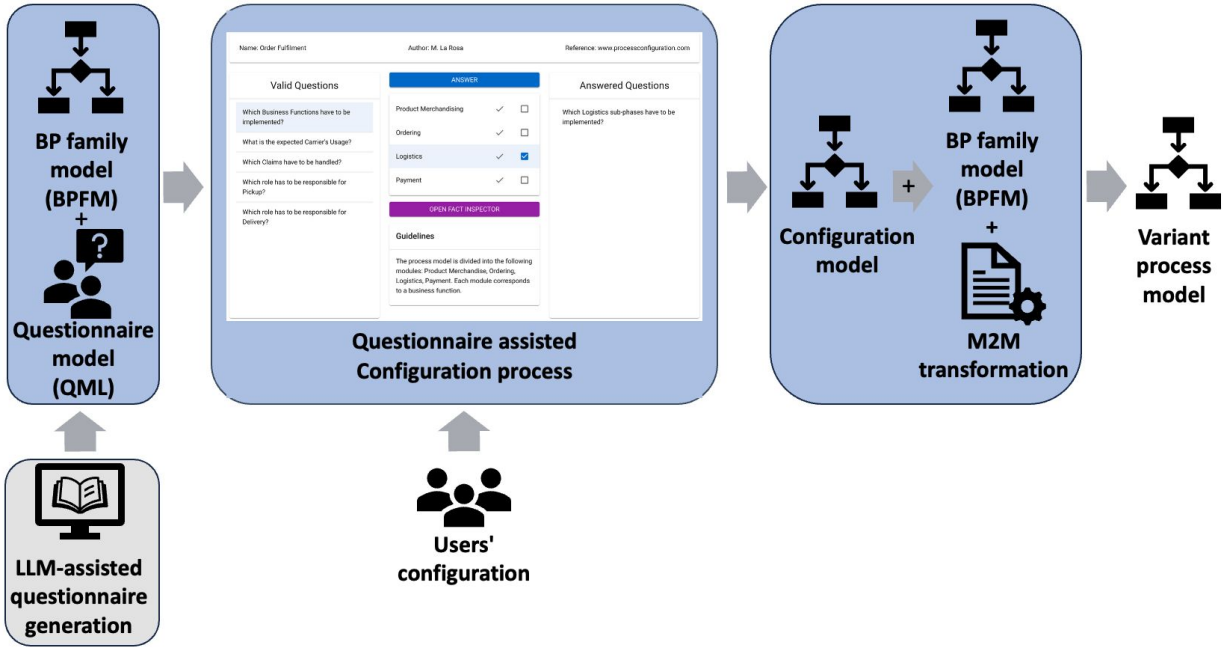
Many valid questionnaires exist, but design decisions are not captured by the underlying BP family.



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.

how LLMs could assist?

LLMs suggest new opportunities, as they can generate human-like text, reason, and assist with knowledge-intensive activities.



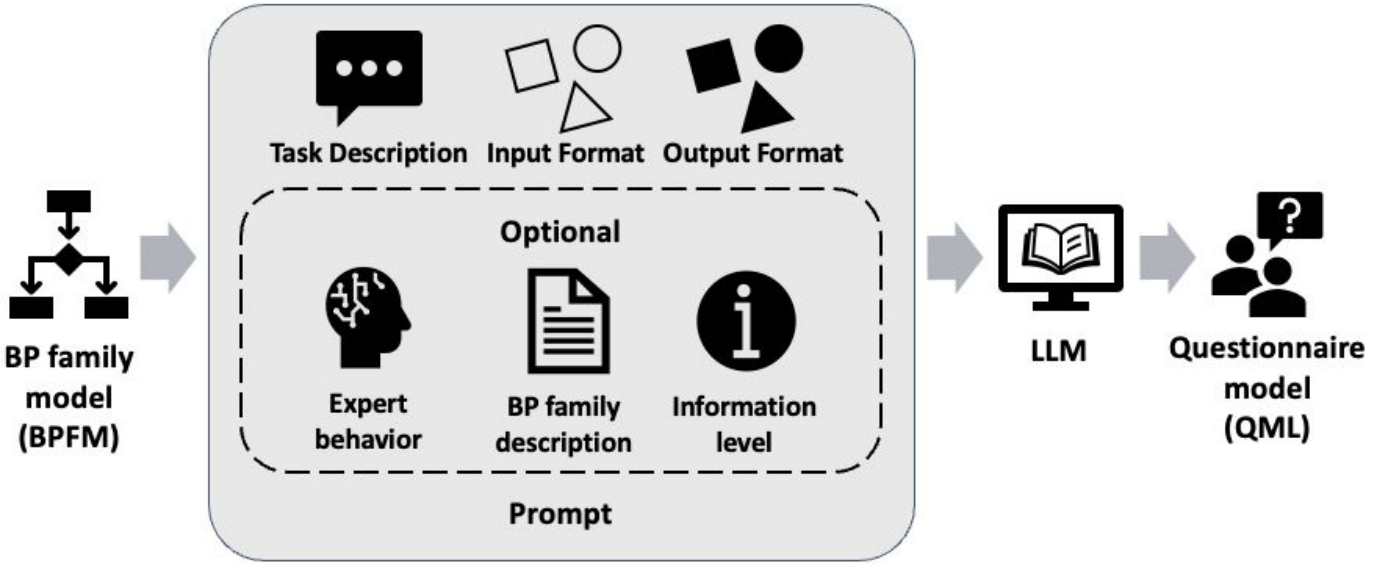
objectives

This paper explores **how LLMs can support the generation of end-user questionnaire models from BP family models**.

Contributions

1. **a pipeline** that maps BP-family variability models to questionnaire-based configuration models
2. **an exploratory evaluation** of how different prompt inputs and scopes affect the generated questionnaire structures
3. **lessons learned** regarding human-in-the-loop validation of AI-generated models.

llm-assisted questionnaire generation



metamodel-guided prompting pipeline

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.

evaluation #1

(RQ1) Does enriching the **prompt input** improve generation?

(RQ2) How does **output complexity** affect quality?

- zero-shot strategy without any feedback loop
- incremental test cases using VICS + “Unborn child acknowledgment”
- GPT 5-1 Thinking model (nov 2025)

#	Input requirements
I1	Model
I2	Enhanced (Model + Description)
I3	Model + Expert
I4	Enhanced Model + Expert

#	Output requirements
O1	Structural elements
O2	Logical connections
O3	Full discovery

evaluation #1 => lessons learned

(RQ1) Does enriching the **prompt input** improve generation?

- ➔ Richer input generally improves completeness and domain adequacy, but may also induce over-inference.
- ➔ Expert knowledge is useful when properly grounded in the source information.
- ➔ Expert knowledge alone may introduce elements not supported by the source model.

(RQ2) How does **output complexity** affect quality?

- ➔ Generation becomes harder as the task requires more semantic reconstruction.

evaluation #2

(RQ3) How do **reasoning LLMs compare**?

- LLMs are compared based on their reasoning capabilities.
- GPT 5-1 Thinking model vs Gemini 3 Pro5
- Advanced tests of I4 input requirements and all output requirements

#	Input requirements
I1	Model
I2	Enhanced (Model + Description)
I3	Model + Expert
I4	Enhanced Model + Expert

#	Output requirements
O1	Structural elements
O2	Logical connections
O3	Full discovery

evaluation #2 => lessons learned

(RQ3) How do **reasoning LLMs compare**?

- ➔ Similar overall performance.
- ➔ Gemini 3 Pro obtained similar or slightly higher scores than GPT-5.1 Thinking in the selected I4 executions.
- ➔ Gemini generated more explicit domain-specific labels.
- ➔ Both models struggled with questionnaire dependencies.

evaluation #3

(RQ4) Does **prompt chaining** help?

GPT 5.1 Thinking model

-> fact definition
-> question definition
-> fact dependencies
-> question dependencies
-> constraints
-> suggestions for improvements

evaluation #3 => lessons learned

(RQ4) Does **prompt chaining** help?

Prompt chaining improves controllability and supports refinement, but its quality benefits are not uniform.

- Fact + questions → very accurate
- Dependencies → still problematic
- Constraints → risk of over-generation
- Improvement step → useful for identifying and correcting accumulated weaknesses

discussions / conclusions

What determines generation quality?

- ➔ Source **semantics matter**.
- ➔ Explicit **structural information** → generally recoverable.
- ➔ **Richer context** → better descriptions and discovery.
- ➔ **Too much expert/context knowledge** → risk of overspecification.
- ➔ **Underspecified design decisions** → difficult to infer reliably.

discussions / conclusions

Implications for an LLM-assisted MDE workflow

- ➔ **Constrain**: explicitly define what the LLM may NOT generate.
- ➔ **Validate**: use metamodel/XMI checkers and semantic checks.
- ➔ **Decompose**: split large models and complex reasoning into smaller generation steps.
- ➔ **Review**: keep domain experts responsible for dependency semantics and final design.
- ➔ **Use offline**: current latency and reliability fit design-time assistance better than interactive configuration.

conclusions

Best suited for AI-assisted modeling, not autonomous generation

Further evaluation and refinement are needed for industrial reliability:

- ➔ broader benchmarking across LLMs, domains, and model sizes
- ➔ user-oriented measures such as comprehensibility and perceived usefulness
- ➔ prompt engineering and knowledge- loading strategies that better constrain the generation process

16th International Model-Driven Requirements Engineering

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

Daniel Calegari[1,2] . Andrea Delgado[2]

[1] Universidad ORT Uruguay | **calegari@ort.edu.uy**

[2] Universidad de la República, Uruguay

UNIVERSIDAD ORT Uruguay

UNIVERSIDAD DE LA REPÚBLICA URUGUAY