Text In, Model Out? Five Levels Where the AI Fails

17 October 2026 · 12 min

> You describe the process in the chat, and the AI draws the BPMN model. I measured that expectation: five language models, three kinds of input documents, more than 170 generated models. The result is a yardstick with five levels, and a hypothesis that turned out to be wrong.

Text in, model out. That is how most people imagine process modelling with AI, and so did I at first. You describe the process in a few sentences in the chat, and the language model delivers the finished BPMN model. If that is true, why would we still need a method, style rules and standards?

I did not debate this question, I measured it: with five language models, three kinds of input documents and more than 170 generated models, every single one checked automatically. This article sums up the experiment and its results. The video tells the same story with the real models from the lab (English version, German original). And whoever wants to measure it themselves will find the material at the end.

The counter-position, in its strongest form

Before I get to the experiment, the counter-position. It deserves to be taken seriously.

First: language models know BPMN. They have read the specification and seen thousands of models. **Second:** they get better every year; what is stuck today will be solved tomorrow. **Third:** whoever provides more documents gets a better model; more context means more knowledge. **Fourth:** method and style rules are ballast from a time when humans still drew models themselves.

Some of this is true. The models really do know the notation, and they are getting better. The only question is whether that is enough.

Why the chat alone is not enough

Whoever tries it out usually experiences the same thing. A strong model delivers a diagram that opens and looks plausible. Roughly right. Often it is not usable.

The reason is not a lack of intelligence, but a missing frame. The chat does not know the company, its terms or its rules. So it does what language models do: it looks for the most likely pattern, it interprets, and where something is missing, it fills the gap. Plausible, but invented. Research calls this hallucination, and it is not a flaw of individual models but a property of the technology (Ji et al., 2023).

It becomes dangerous because the result looks good. Automation research has described for decades that people follow a machine's result all the more readily the more reliable it appears (Parasuraman & Riley, 1997). A neatly drawn but incomplete model is therefore riskier than an obviously broken one.

And whoever then uploads the existing documents, the Word document, the training slides, a note, does not necessarily make it better. Documents contradict each other, as in every real project. The AI then silently decides which one to believe.

So everyone talks past each other. Some say: it works. Others: it is all hallucinated. Both have an example. What is missing is a yardstick.

A yardstick: five levels

I propose five levels that build on each other. Each one requires the previous one.

Level	Name	The question
1	**It loads**	Is the result valid BPMN, with a diagram, and does it open in any tool?
2	**Complete**	Are all described cases there, not only the normal case, but also the exceptions?
3	**Stable**	Whoever asks twice, do they get the same flow twice?
4	**Faithful**	Is nothing invented, and do the terms remain the terms of the domain experts?
5	**Honest**	Where the sources contradict each other: does the AI ask instead of deciding silently?

Only from level 4 on can you trust a model. And only with level 5 can you take responsibility for it.

The experiment

The process is deliberately ordinary: a return in online retail. The customer announces a return, it is checked, the goods come back, are inspected, and the money is refunded. Or not: the return is rejected, or the goods are not in order.

Three variants served as input: a clean text; a note written from memory, the way such notes come about in practice; and three documents that deliberately contradict each other a little (a Word document, training slides and that note). Each variant went to each language model ten times, always with the same simple request: «Erstelle aus dieser Beschreibung ein BPMN-2.0-Modell als XML.» ("Create a BPMN 2.0 model as XML from this description.") One message, no system prompt, no follow-up corrections. Every result was saved and checked automatically against the levels.

Level 1: Does it load?

With the note as input, three of five models did not deliver a single model that opens, in any run.

Model	valid, with diagram (of 10)
gpt-oss (120b)	0
DeepSeek V4 Flash	0
GPT-4.1	0
GPT-5 mini	3
GLM-5	10

The errors are instructive. Sometimes the XML is simply broken, for example because a comment contains a double hyphen, which XML forbids. More often the diagram part is missing entirely: the model describes a process, but it does not draw one. A viewer then only shows "no diagram to display".

So the first level cannot be taken for granted. And it is pure technology; it says nothing yet about the process itself.

Levels 2 and 3: Complete and stable?

The note clearly says that the warehouse checks whether the goods are correct and in order. A complete model therefore needs a path for the case that they are not.

GLM-5, the model that always drew valid models at level 1, modelled this path in one of ten runs. The other nine show the normal case, neatly drawn. That is exactly what is dangerous: a model that looks good but only knows the normal case.

And as for stability: ten times the same question gave six different flows with GLM-5, seven with GPT-5 mini. Not different names, but different steps, different orders, different outcomes.

This is where the standard tools get stuck. A better model helps with level 1. With levels 2 and 3 it does not.

What the method changes

In the next step, the same experiment ran through a method, implemented in my lab tool MESTRO. Deliberately with the weakest model of the experiment, gpt-oss, the one that had not managed a single valid model on its own.

Method means four things here. The AI fills a fixed **schema** instead of writing XML freely. It follows **style rules** from *BPMN Method and Style* (Silver, 2011): activities as verb and object, named end states, decisions as questions. The layout is done by a **convention**, not by the language model. And every result is **checked** before a human sees it; whatever violates the rules goes back to the AI. None of these are new ideas: empirical modelling research has long shown such guidelines to drive understandability and fewer errors (Mendling, Reijers & van der Aalst, 2010). Process research that

has language models generate process models also relies on a fixed intermediate structure and a checking loop instead of free output (Kourani et al., 2024).

The result: ten of ten models valid, with a diagram. The path "goods not in order" was there in nine to ten of ten runs.

But honestly: it is not stable. Even with the method, five to seven different flows came out. The method makes sure of what is described. What is open stays open.

Level 4: More documents, more truths

For level 4 I had a hypothesis: the more documents, the more cases the AI finds. The documents contain, for example, a deadline that is missing from the note: the goods must be back within seven days.

The opposite happened. With the three documents, almost all models produced ten different flows in ten runs, even with the method. The seven-day deadline appeared in at most three of ten runs.

And the contradictions? The note says one thing, the slides and the Word document another:

Note	Slides / Word
refund only	or replacement delivery
no formal closure	the case is closed
no supervisor	involve a supervisor

Every model silently decided for one side. The replacement delivery ended up in the model in eight to ten of ten runs, with all models. And the "Refund Request", which the domain experts explicitly call it in the note, became a «Rücksendeanfrage» (German for "return request") in ten of ten runs with the method.

The hypothesis was wrong. More documents do not bring more cases. They bring more truths. And the AI picks one of them without saying so.

Level 5: The AI sees the contradiction and resolves it silently

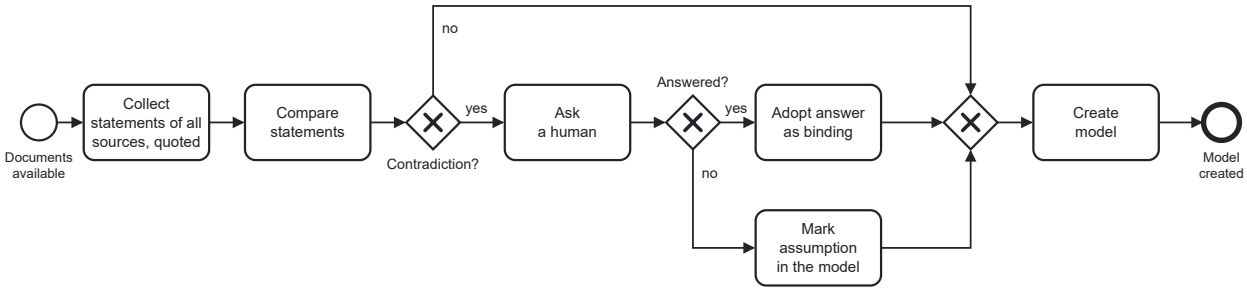
The most revealing finding lies in the raw answers of the plain chats. DeepSeek wrote in the documentation of its model: "Sources differ regarding escalations, replacement deliveries and the use of terms." gpt-oss wrote: "Replacement vs. refund – both options are ... shown; the company can ... decide." (Both translated from the German originals.)

So the language models do see the contradictions. They just resolve them themselves, one way one time, another way the next. That is exactly where the drift of level 4 comes from.

In my lab it was no different. The qualification of the process even named the contradictions literally, for example "inconsistency: replacement delivery vs. refund only". But this knowledge went nowhere; the

generated model contained the replacement delivery anyway.

So we improved it. Not the language model: it stayed the same, nothing was trained. What we changed was the flow.



Before modelling, the statements of all sources are collected with their quotes and compared. Every contradiction becomes a question to a human. Their answer is then binding. And whatever nobody answers appears in the model as a visible assumption instead of being decided silently. How far this source reconciliation actually reaches level 5 is something I am measuring right now; the result will follow.

What remains, level by level

Level	Finding
1 · It loads	A strong language model manages it on its own. With the method, a weak one does too.
2 · Complete	Only the method manages it. The chat draws the normal case.
3 · Stable	Nobody manages it fully, not even the method. Where the description is open, the AI decides anew every time.
4 · Faithful	With several documents, all fail as long as the AI resolves contradictions itself.
5 · Honest	Only when every contradiction becomes a question.

Levels 3, 4 and 5 belong together. A model can only become stable and faithful when someone decides the open points. And that someone is not the AI.

Measure it yourself

The levels are simple enough to measure with any tool, without a program. On bpmn.ch there is a package for it: the note from the experiment, the request, a checklist for levels 1 to 3, a score sheet and the reference values from the video (bpmn.ch/v/1; the guide is in German, the note and the documents are in English). Ask ten times, open every result in a BPMN viewer, tick it off.

Why the checklist stops at level 3. Because up to there, anyone can measure along, with any tool and any model. Does it load? Is the exception path there? Does the same come out twice? You see that in the viewer and tick it off on the score sheet. Nothing more is needed — no access, no programming, no arrangement with me. That is exactly why these three levels are the yardstick: they are the same for everyone.

From level 4 on, the nature of the question changes. Faithfulness can only be checked when there are several sources that contradict each other — so you first have to build the case. And level 5 cannot be read off the model at all. It does not ask what the result looks like, but whether the tool **asked back**. A tool that decided silently delivers a flawless diagram; the error is invisible in the result. You can only see it in the flow before.

With that, it stops being a property of the language model and becomes a property of the frame around it — the source reconciliation with the question to a human that I described above. In the experiment I added this step to MESTRO; whether it really reaches level 5, I am still measuring.

MESTRO addresses all five levels, each at the point where the level is decided: schema and checking loop at level 1, the method at level 2, style rules and layout convention at level 3, source reconciliation at level 4 and the question to a human at level 5. Addressing does not mean reaching — levels 3 and 5 are still open in this experiment. But it shows where the work lies: not in the language model, but in the frame.

Whoever wants to try it anyway: the three contradicting documents are included in the package. That is enough for level 4. For level 5, the question is less what your tool draws than whether it asks you something.

I am interested in which level your tool reaches, and where it breaks for you.

Conclusion

The counter-position is right on one point: language models can do BPMN, and the best of them reliably draw valid models today. But level 1 is technology, not process understanding. Completeness only came with the method in the experiment. And more documents did not make the models better but more arbitrary, because the AI silently decides for itself contradictions it recognises. Reliability does not come from the language model. It comes from the frame it works in, and from the question to a human.

An AI that does not ask decides for you.

Note on the experiment: models called via the API with the default temperature, ten runs per input, as of October 2026. Evaluated automatically; all documents are fictional. MESTRO is my lab tool and is being developed at ELCA, where I work; it serves here as a reference implementation of the method, not as a product comparison.

Sources & further reading

- Video for episode 1 "When AI Models BPMN" on the channel "Stephan Fischli · Agentic Orchestration": English version, German original
- Measurement package for the video on bpmn.ch: bpmn.ch/v/1
- Z. Ji, N. Lee, R. Frieske et al.: *Survey of Hallucination in Natural Language Generation*. ACM Computing Surveys 55(12), 2023.
- H. Kourani, A. Berti, D. Schuster, W. M. P. van der Aalst: *Process Modeling with Large Language Models*. In: Enterprise, Business-Process and Information Systems Modeling (BPMDS/EMMSAD 2024), LNBIP, Springer, 2024.
- J. Mendling, H. A. Reijers, W. M. P. van der Aalst: *Seven process modeling guidelines (7PMG)*. Information and Software Technology 52(2), 2010, pp. 127–136.
- R. Parasuraman, V. Riley: *Humans and Automation: Use, Misuse, Disuse, Abuse*. Human Factors 39(2), 1997, pp. 230–253.
- B. Silver: *BPMN Method and Style*, 2nd edition, Cody-Cassidy Press, 2011. https://methodandstyle.com/
- Object Management Group: *Business Process Model and Notation (BPMN) Version 2.0.2*, 2013. https://www.omg.org/spec/BPMN/2.0.2/