

Fixed rules, thinking colleagues

Fixed code wherever possible, and language models where they fit

Summary

- **The question:** what do we use a language model for, and what do we use fixed code for?
- **What we do:** as much work as possible follows fixed rules. Language models think within those rules.
- **What we saw:** the consequential errors occur between the components.
- **What it means:** a rule is a law. It never hesitates, so it needs a brake.
- **The main limitation:** this is one small workshop. It is not representative of SMEs.

Why we are writing this

When people think of AI, they picture a chat window. In our workshop, many AI colleagues work at once, day and night. Fixed code sits between them. Without that context, our examples are easy to misread.

Rule, model or person?

In our workshop, every task has a place. Try it yourself.

Who handles this task?

Task	Who	Why
Check whether an IBAN is valid.	A fixed rule	An IBAN has a defined format and check digits. A rule can check those consistently.
Summarise a client conversation.	A language model	Summarising requires an understanding of ordinary language. That is something a language model does well.
Send the quotation.	A person approves	Sending cannot be undone. A person therefore gives explicit approval.
Count how often a task fails.	A fixed rule	Counting requires precision. A rule counts consistently and can raise an alert at a threshold.
Judge whether an email sounds friendly.	A language model	Tone calls for judgement about language. A model can do that, although it can be wrong.
Pay a new supplier.	A person approves	A payment cannot be undone. A rule runs the checks first, but a person approves it.

This is how we divide the work: rules for precision, models for language and judgement, and a person for irreversible actions.

A rule follows a law; a model thinks

Fixed rule	Language model
• Predictable	• Understands ordinary language
• Can be checked with a test	• Writes and summarises
• You can see why it refuses	• Makes connections
• Cheap and fast	• Can notice its own error
But: It never hesitates.	But: It can be wrong.

Ask both the same question ten times. The rule gives the same answer ten times. The model does not always do so.

The same question, twice

Who	Answer
Question	May this email be sent? It contains an internal name.
The fixed rule, every time	No. It contains an internal name.
The language model, attempt 1	Yes, the tone is friendly.
The language model, attempt 2	Better not: the second sentence contains a name that looks internal.
The language model, attempt 3	Yes, but check the names once more.
The language model, attempt 4	No, it contains something that looks internal.

A simulated example.

That is why a model never decides on an irreversible action on its own in our workshop.

We put rules around the thinking

Think of it as a house. Point to a part.

The house, part by part

- **The rooms.** This is where language models work. They think, write and assess.
- **The walls.** The fixed rules. They define what must always happen and what must never happen.
- **The front door.** A fixed scan. Everything leaving the workshop passes through here.
- **The window.** A second look. Another model reviews the work, because the same model shares the same blind spots.
- **The key.** A person holds it. Sending, paying, deleting and publishing happen only after their approval.

A rule needs a brake too

A rule never hesitates. One night in September, two fixed rules kept each other busy. By morning, the counter had reached 20,478. See whether you can stop the loop yourself.

Stop the loop

Two fixed rules keep sending each other notes. The first asks for room. The second follows a fixed rule: "Is there room? Then yes." Nobody intervenes. By morning, the count is 20,478 failed tasks (measured). With a brake, the rule says: "One retry, then a person." The loop then stops. A simulation of that night. Only the final figure was measured.

Since then, the rule has been: one retry, then a person. The system involved remains switched off until that brake is in place.

Keep the scale in mind

Your coding assistant	Our workshop
• One model	• Many AI colleagues
• One conversation	• Day and night
• You are watching	• Even when nobody is watching
	But: The consequential errors arise between the components.

That is why our articles often focus on the space between those components.

Limitations

- **Not representative.** This describes how one small workshop operates.
- **Not a measurement.** We did not count how much work follows rules and how much uses models.
- **A snapshot.** The capabilities of language models change quickly.

What this means

What does this mean for you?

- **Which task could a fixed rule handle better than your AI?** Tip: Start with anything that requires precision: counting, checking and validating a format.
- **Where does a model make a decision that a rule should check?** Tip: Look at everything that leaves your organisation. Put a fixed scan in front of it.
- **Who stops a rule that gets stuck in a loop?** Tip: Allow each automatic repetition one retry. After that, a person takes a look.

Closing notes

With thanks to the people and AI colleagues at ITARR.

Also read: A night in the workshop and the working policy Stay in control of your AI.

How to cite this: ITARR Lab (2026). *Fixed rules, thinking colleagues*. Version 0.2, as of 17 October 2026. itarr.eu/lab/vaste-regels-denkende-collegas. CC BY 4.0.

Sources: ITARR’s own working rules, as of 17 October 2026; a night in September 2026, shown in the film at itarr.nl/etalage/nacht/ (as of 17 October 2026).

Licence: CC BY 4.0. Use, adapt and share it. Attribute ITARR Lab, link to the licence and indicate any changes.