EMF-Kaizen: an intelligent assistant for domain-specific modelling and meta-modelling

Lissette Almonte, Jefferson Iván Rengifo, Esther Guerra, Juan de Lara

MISO - Modelling & Software Engineering Research Group (miso.es)
Universidad Autónoma de Madrid (Spain)

Motivation

- Building models and meta-models is complex
  - expertise on modelling language
  - expertise on domain
  - expertise on modelled system
  - expertise on high-quality modelling
  - ...
- Need for a modelling assistant...
  - akin to programming assistants (GitHub Copilot, Gemini Code Assist...)
  - able to deal with arbitrary modelling languages

State of the Art

- Analysis of approaches for modelling assistance
- Most recent ones based on Large Language Models (LLMs)
- Suffer from some limitations (one or more):
  - target a single language (e.g., UML, Ecore)
  - fixed assistance (e.g., recommending attributes)
  - target either models or meta-models (not both)
  - some limited to textual languages
  - not conversational
  - not integrated within the modelling environment

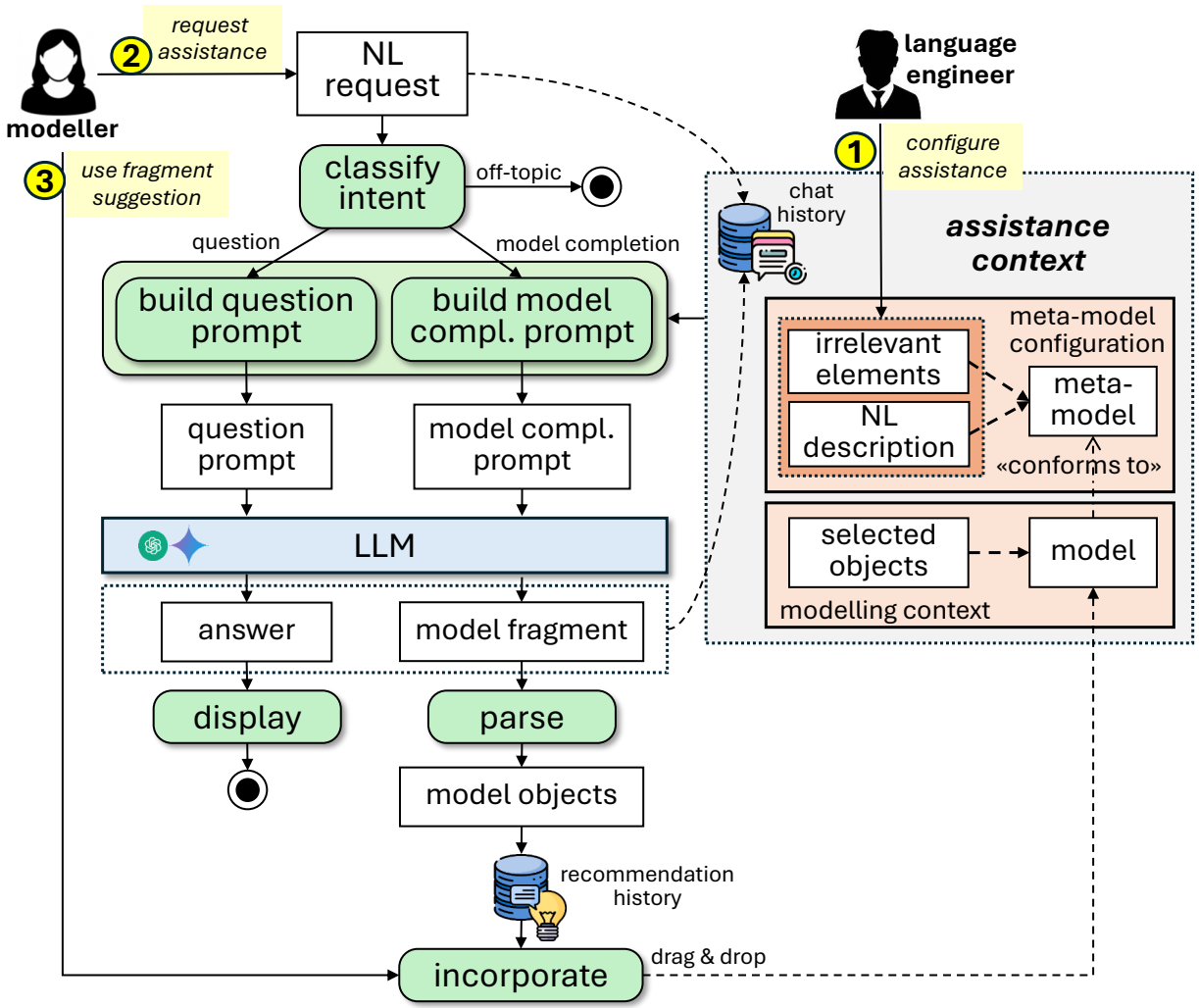
Contribution

- EMF-Kaizen: Novel approach & tool for modelling assistance
  - **LLM-based**
  - **conversational**: flexible assistance requests
  - **meta-model agnostic**: for arbitrary languages
  - **meta-level agnostic**: for models and meta-models
  - **concrete-syntax independent**: for graphical, textual...
  - **integrates** off-the-shelf within EMF tree editors

- Evaluation
  - effectiveness
  - LLM effect
  - prompting strategies

Approach

Working Scheme

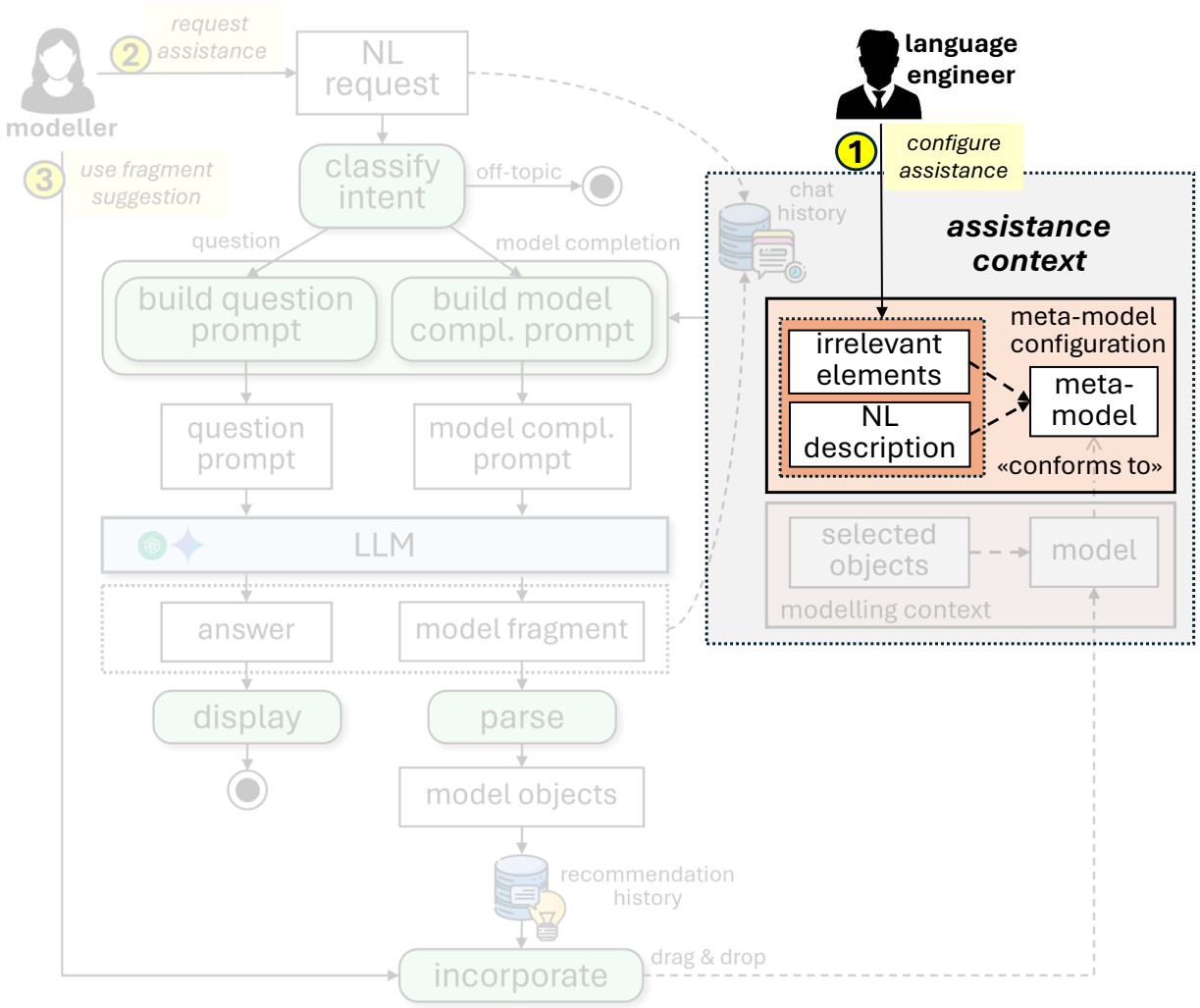


LLM-based conversational assistant

Two phases:

1. Assistant configuration
2. Assistant use

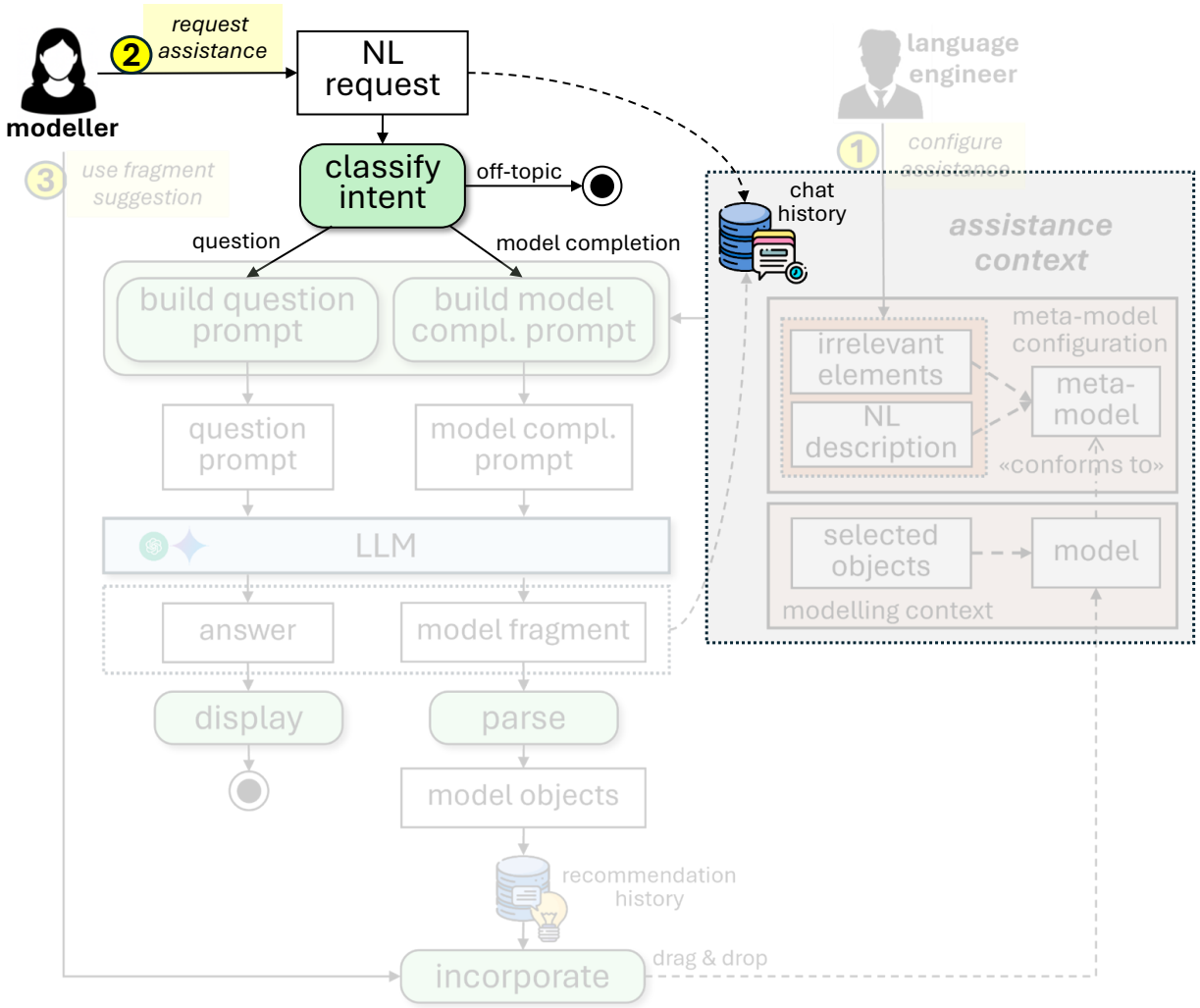
1. Assistant Configuration Phase



- Meta-model of the modelling language
- (o) Filter irrelevant elements
- (o) NL description of meta-model
- (o) Default LLM and parameters

→ EMF-Kaizen generates synthetic example models.

2. Request Assistance

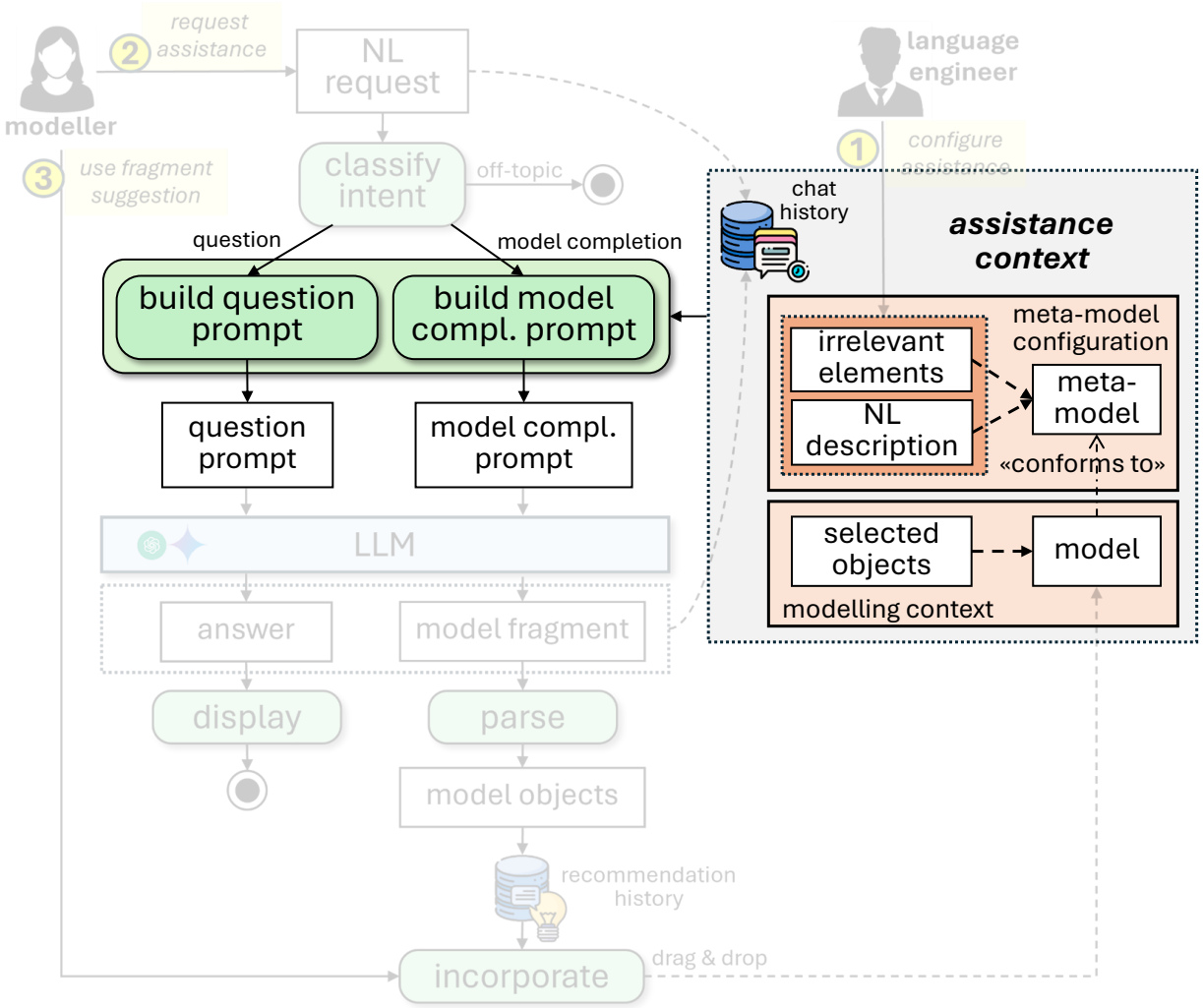


i Request in NL

ii LLM-based classification of modeller intent:

- question
- request to complete current model
- off-topic otherwise

2. Request Assistance

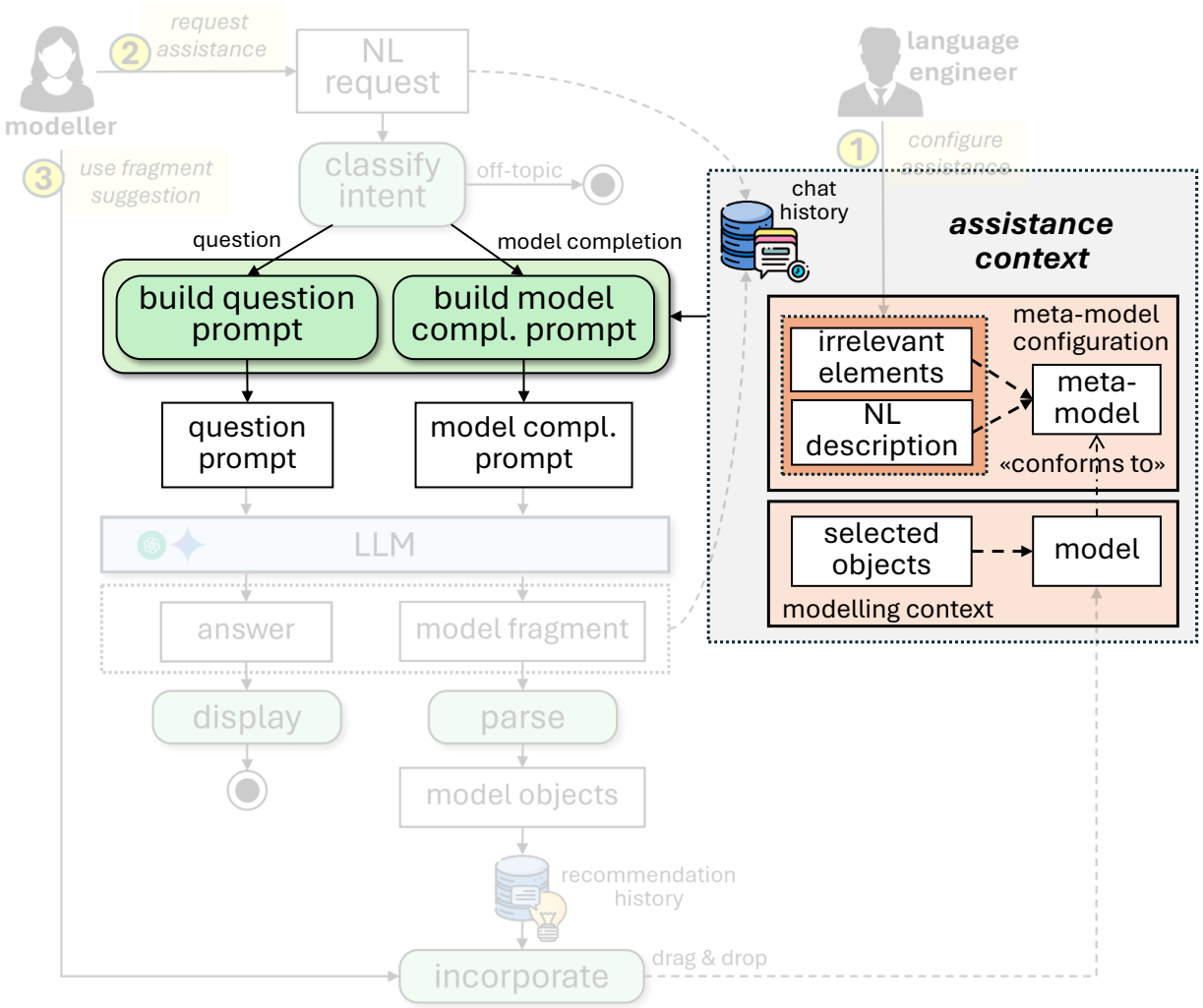


iii Prompt construction

Common part:

- system prompt
- NL request
- selected objects in current model
- current model (JSON, filtered)
- its meta-model (filtered)
- MM description
- conversation history

2. Request Assistance

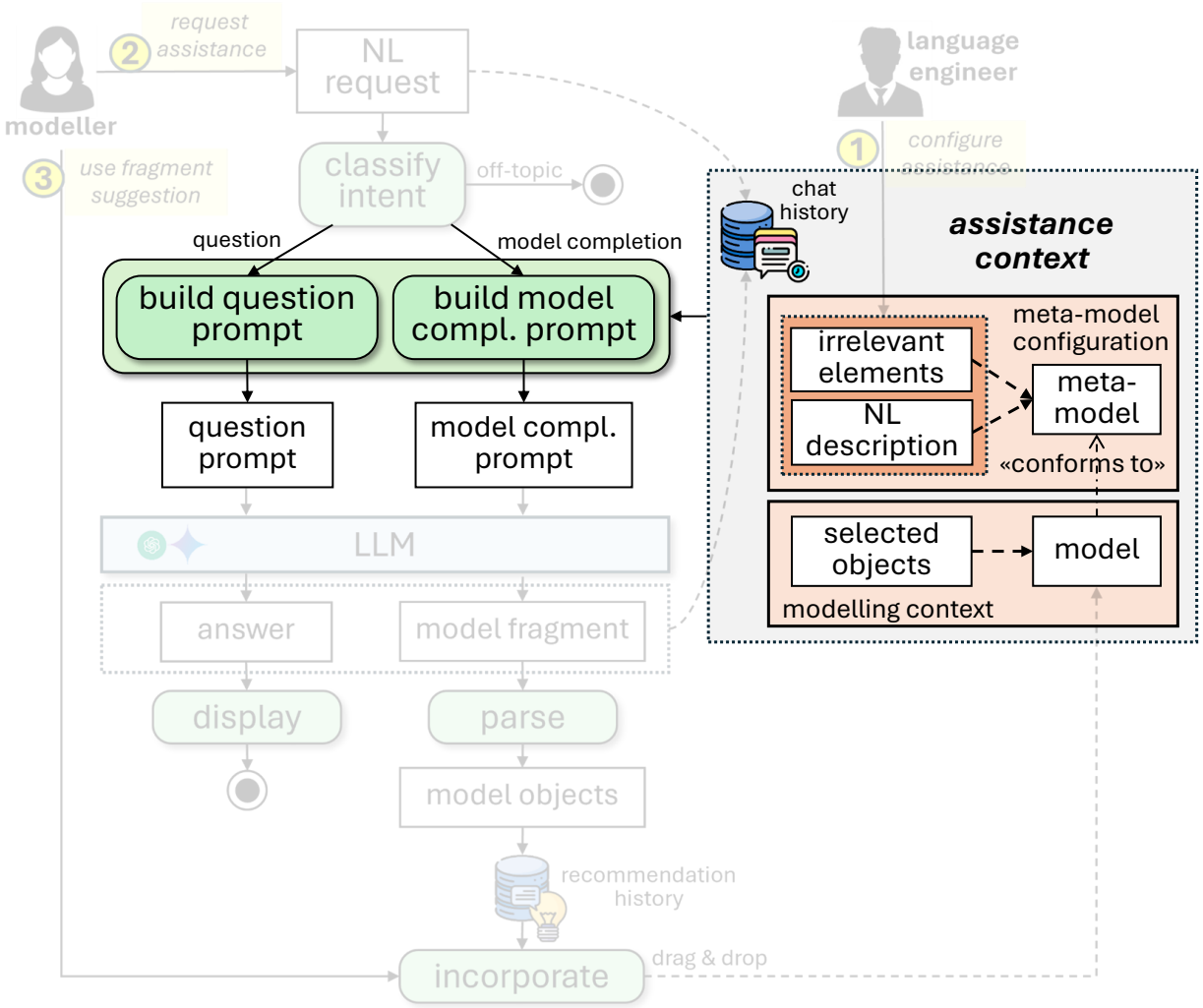


iii Prompt construction

Query-specific:

- question prompt (*NL explanation as response*)
- response guidelines (*forbid JSON or model generation*)

2. Request Assistance

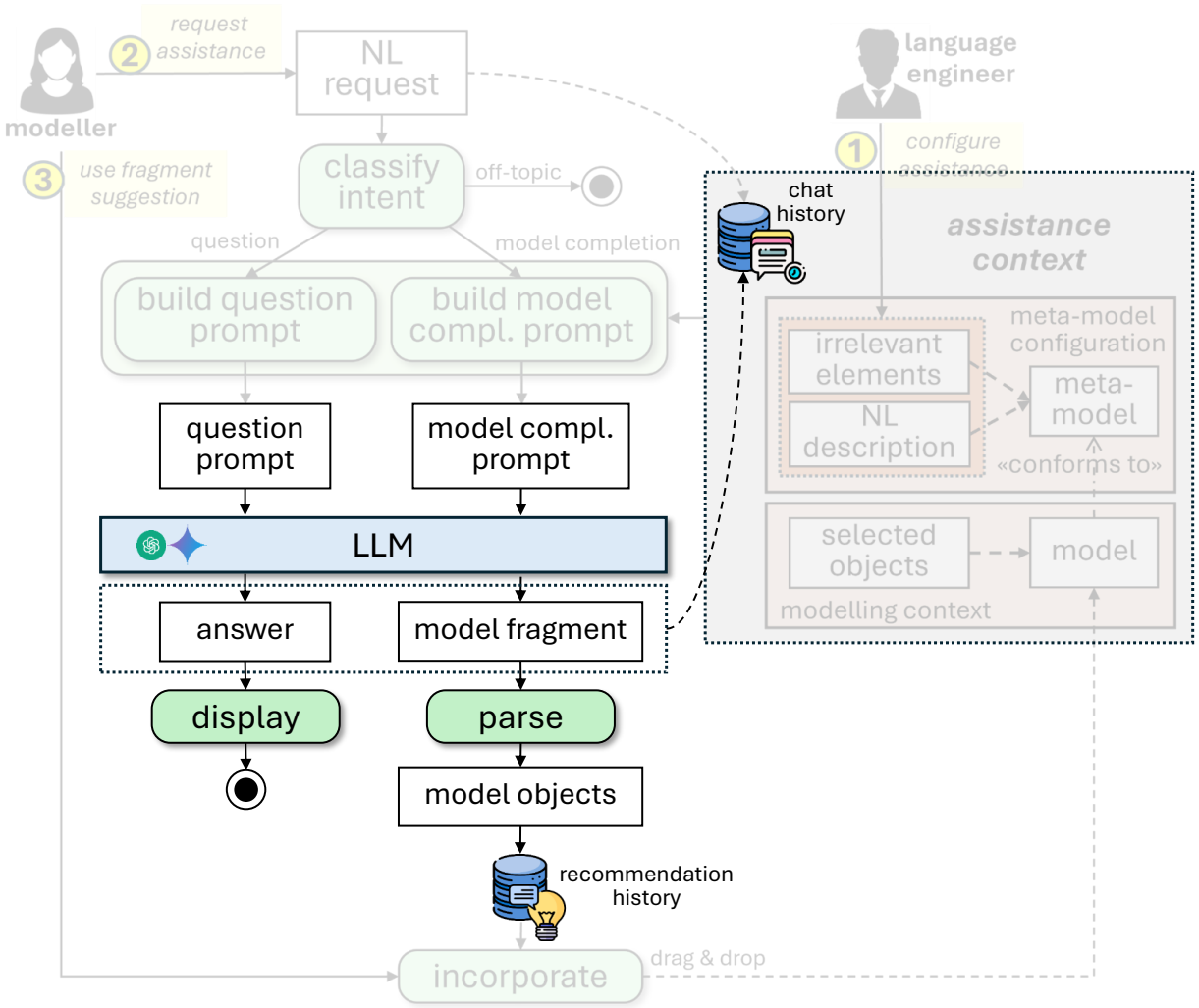


iii Prompt construction

Completion-specific:

- completion prompt (*model fragment conformant to MM*)
- response guidelines (*only a JSON fragment*)
- example models (JSON, synthetic)

2. Request Assistance

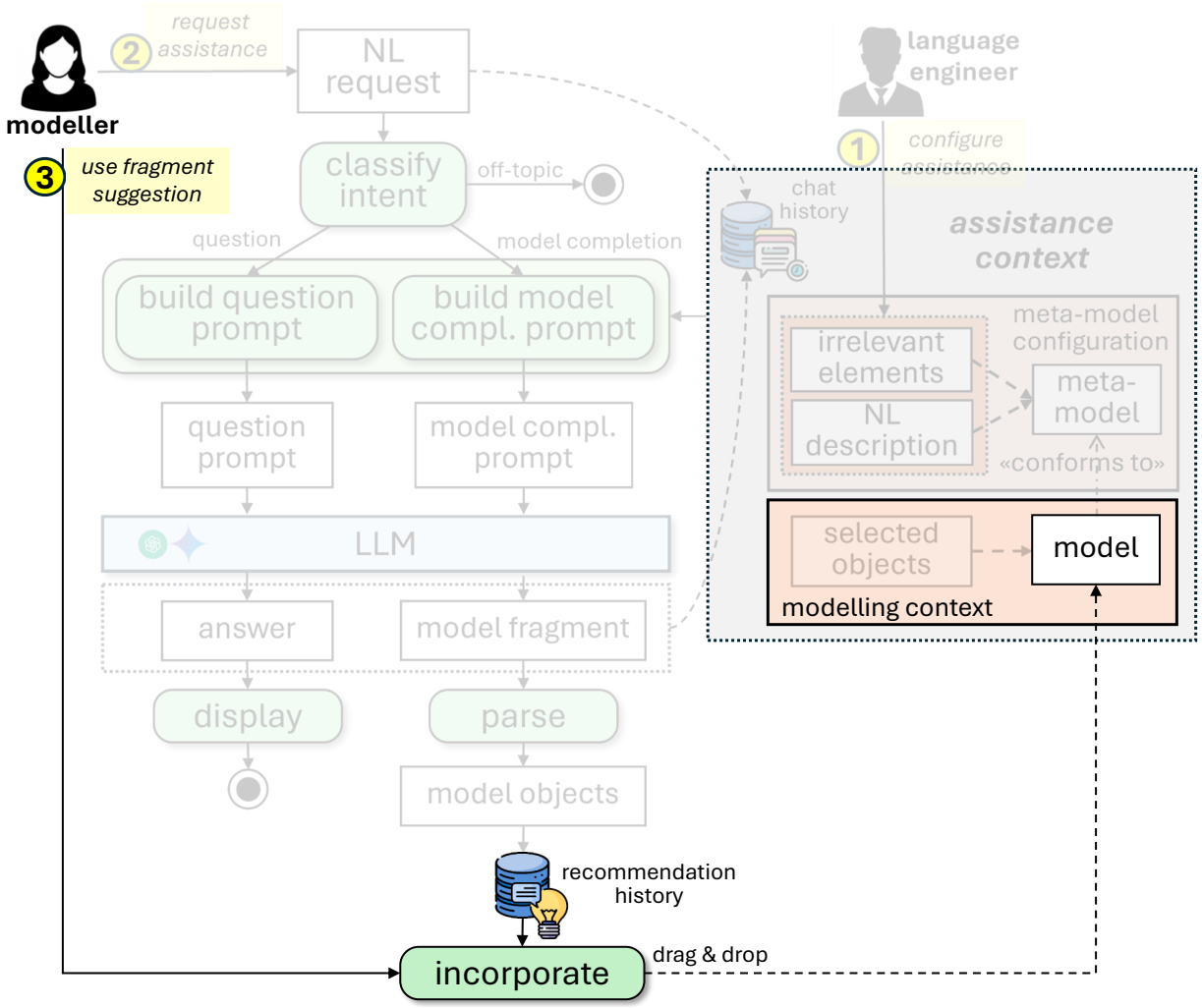


iv LLM is prompted

v Response processing

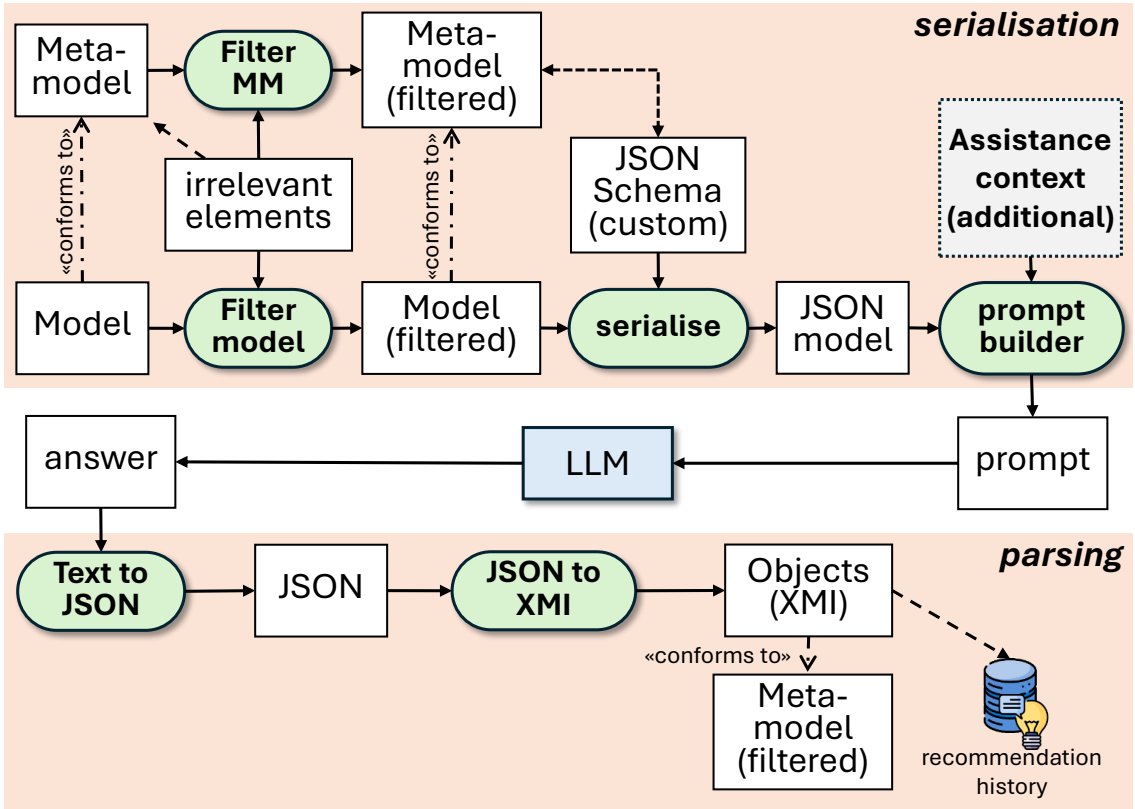
- query: display
- completion request: parse JSON into EMF objects

3. Use Fragment Suggestion



- EMF objects are stored and displayed
- Drag&drop of objects
  - into current model
  - into other models
  - fully / partially

Why JSON as the model exchange format?



Why JSON? Robustness!

✗ LLMs can produce XMI with errors

✗ EMF parser fails

✓ JSON serialiser

comps → nested dicts
refs → unique XMI:ids

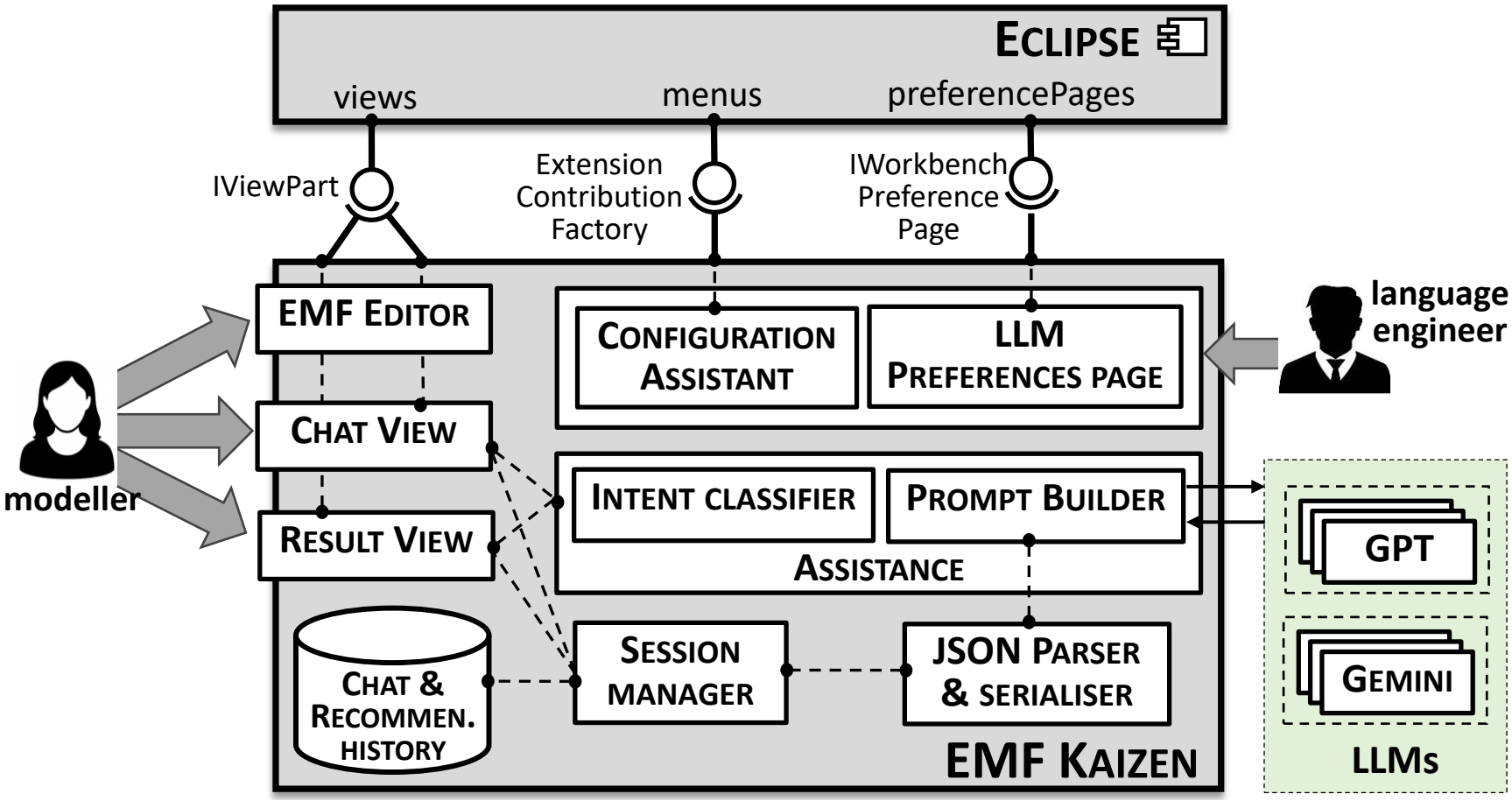
✓ Flexible JSON parser

fix mistakes, like non-existing attributes

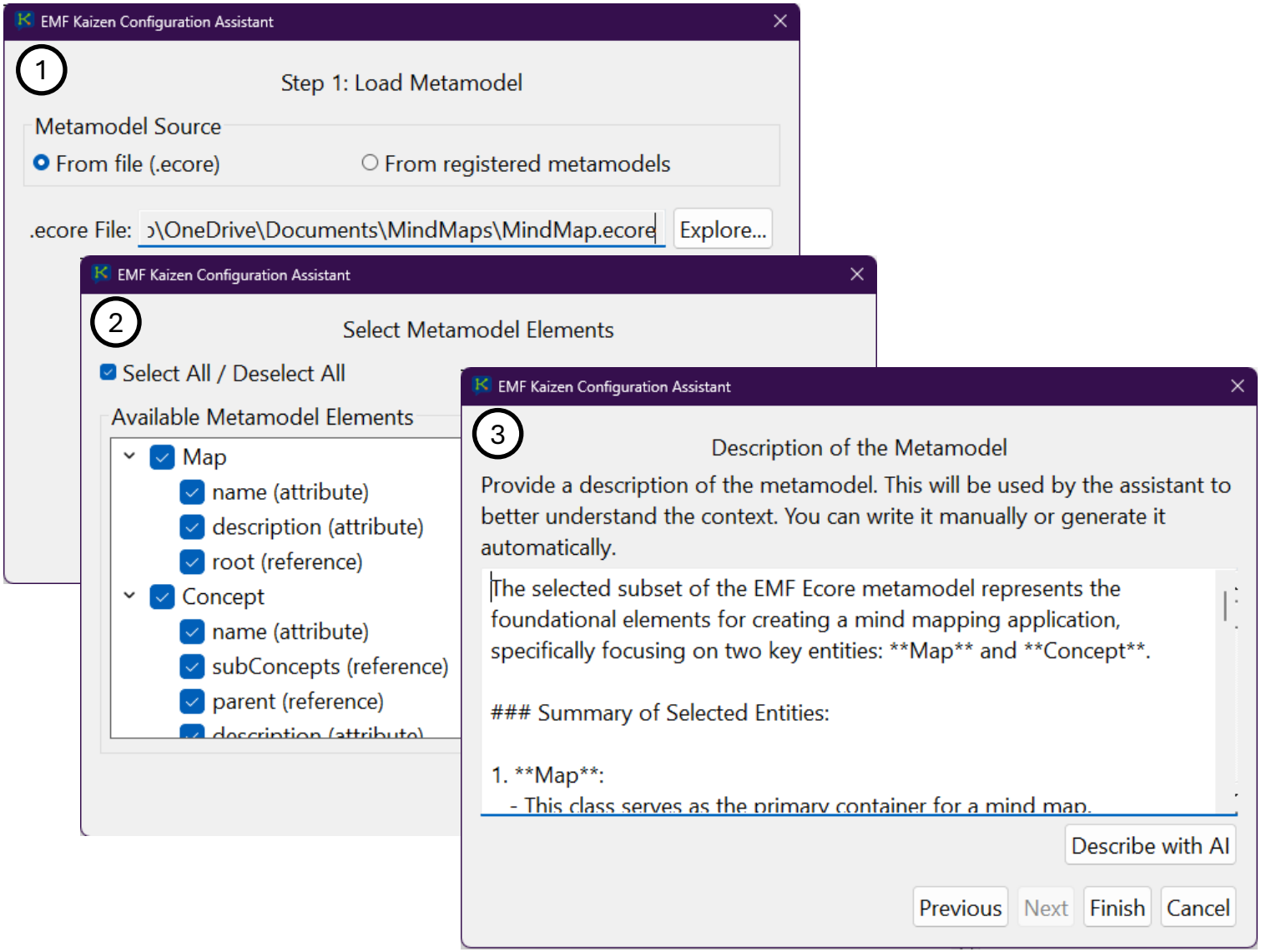
Tool

EMF-Kaizen

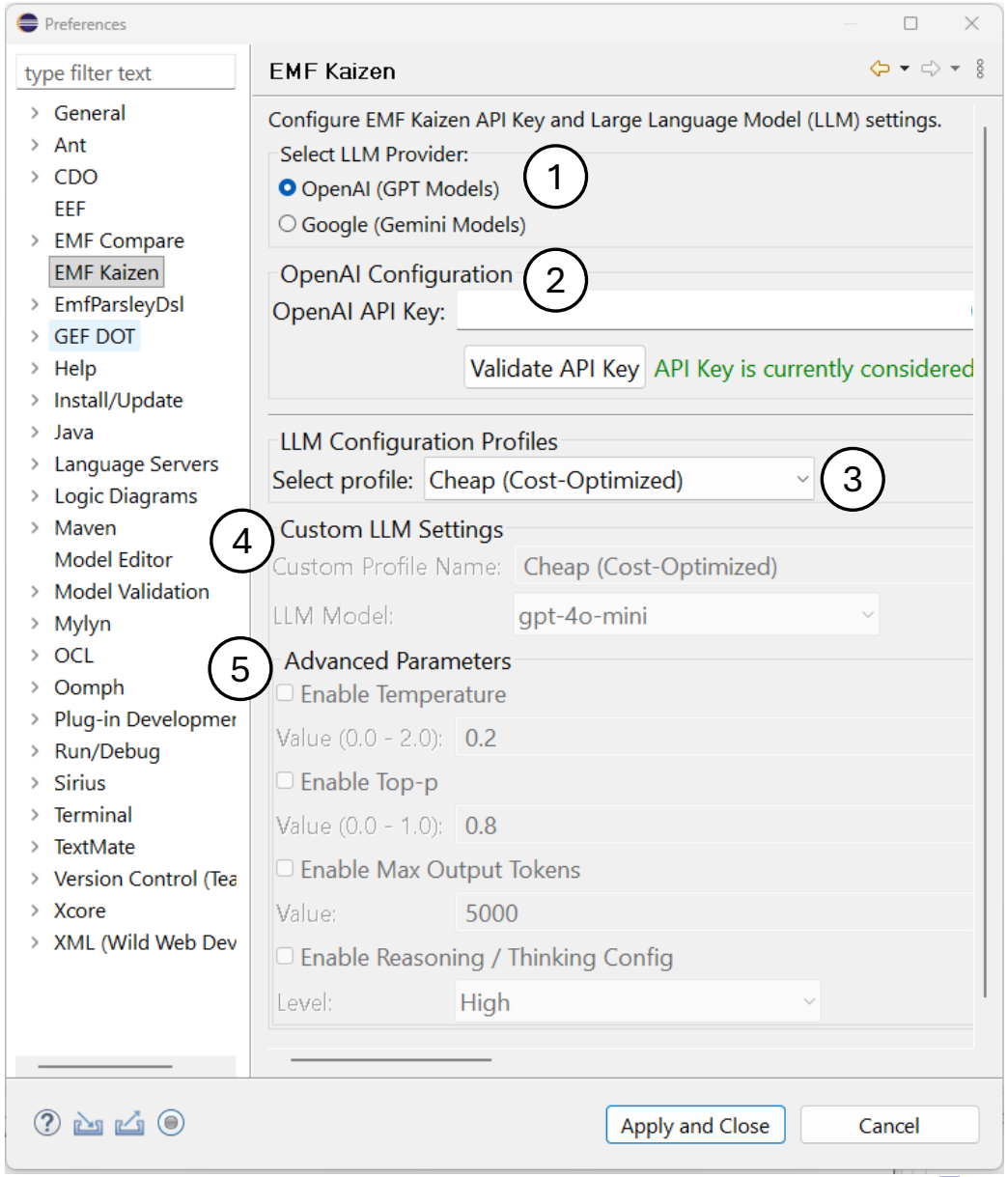
Architecture



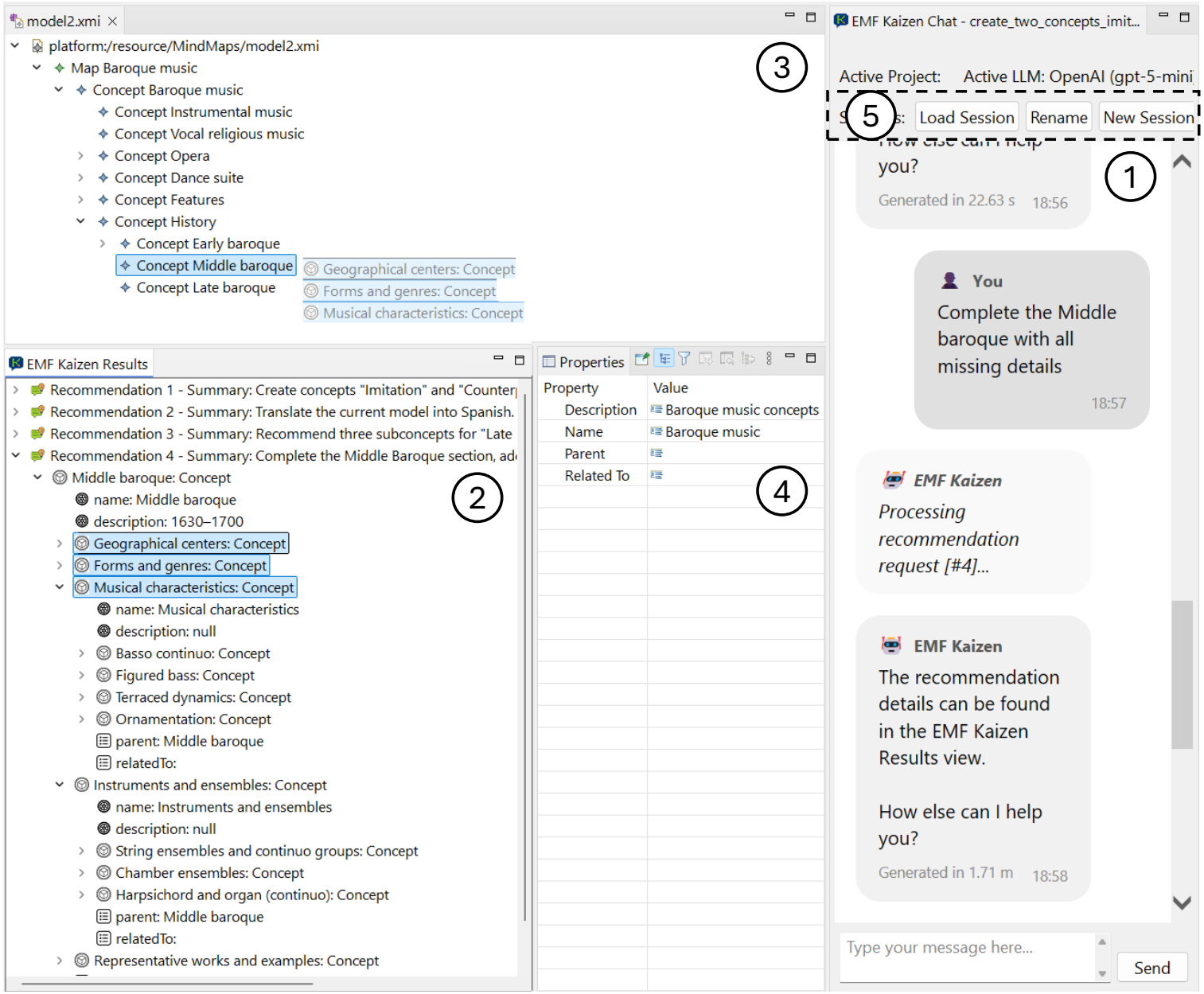
Configuration



LLM Preferences



Assistance


model2.xmi
platform:/resource/MindMaps/model2.xmi
Map Baroque music
Concept Baroque music
Concept Instrumental music
Concept Vocal religious music
Concept Opera
Concept Dance suite
Concept Features
Concept History
Concept Early baroque
Concept Middle baroque
Concept Late baroque
Geographical centers: Concept
Forms and genres: Concept
Musical characteristics: Concept
3
EMF Kaizen Results
Recommendation 1 - Summary: Create concepts "Imitation" and "Counter
Recommendation 2 - Summary: Translate the current model into Spanish.
Recommendation 3 - Summary: Recommend three subconcepts for "Late
Recommendation 4 - Summary: Complete the Middle Baroque section, ad
Middle baroque: Concept
name: Middle baroque
description: 1630–1700
Geographical centers: Concept
Forms and genres: Concept
Musical characteristics: Concept
name: Musical characteristics
description: null
Basso continuo: Concept
Figured bass: Concept
Terraced dynamics: Concept
Ornamentation: Concept
parent: Middle baroque
relatedTo:
Instruments and ensembles: Concept
name: Instruments and ensembles
description: null
String ensembles and continuo groups: Concept
Chamber ensembles: Concept
Harpsichord and organ (continuo): Concept
parent: Middle baroque
relatedTo:
Representative works and examples: Concept
2
Properties
Property
Value
Description
Baroque music concepts
Name
Baroque music
Parent
Related To
4
EMF Kaizen Chat - create_two_concepts_imit...
Active Project:
Active LLM: OpenAI (gpt-5-mini
Load Session
Rename
New Session
5
you?
Generated in 22.63 s
18:56
1
You
Complete the Middle baroque with all missing details
18:57
EMF Kaizen
Processing recommendation request [#4]...
EMF Kaizen
The recommendation details can be found in the EMF Kaizen Results view.
How else can I help you?
Generated in 1.71 m
18:58
Type your message here...
Send

Evaluation:

Effectiveness of EMF-Kaizen

Effectiveness: Experiment Design

- Dataset of meta-models, models, and NL requests
- EMF-Kaizen to obtain model fragments out of the requests

  4 LLMs:
  - Google: gemini-2.5-flash-lite
  - OpenAI: gpt-4o-mini
  - OpenAI: gpt-4.1-mini
  - OpenAI: gpt-5-mini
- Analysis of produced model fragments:
  - syntactical correctness (once converted into XMI)
  - semantical meaningfulness
  - redundancy (unnecessary objects repeated from current model)
  - size
  - response time

Effectiveness: Dataset

5 meta-models

- Ecore filtered to discard elements absent from Ecore editor
- Rest of meta-models are hand-made to avoid data contamination
- NL description generated automatically, adjusted if needed

	Original Size			Configuration			
Meta-model	**#Class**	**#Atts**	**#Refs**	**#Class**	**#Atts**	**#Refs**	**#Desc. Words**
Ecore	20	33	48	13	15	9	371
Mind-maps	2	5	3	2	5	3	266
Wizards	9	12	7	9	12	7	367
State machines	8	8	7	8	8	7	366
Process models	11	5	4	11	5	4	316

Effectiveness: Dataset

15 models (3 per meta-model)

- Hand-made to avoid data contamination
- 3 levels of completion: initial, medium, (almost) complete

Model	Description	Size
Ecore-0	Meta-model of a Petri net (initial)	2
Ecore-1	Meta-model for goal models (incomplete)	10
Ecore-2	A meta-model for a tourism application (complete)	55
Mind-map-0	Mind-map for AI and LLMs for education (initial)	2
Mind-map-1	Mind-map for healthy living (mid-completion)	12
Mind-map-2	Mind-map for baroque music (some details missing)	29
Wizard-0	Wizard to customise LLM requests (initial)	2
Wizard-1	Wizard to set up a user account (mid-completion)	10
Wizard-2	Onboarding data collection (some details missing)	21
State-machine-0	State machine for traffic light (empty)	1
State-machine-1	State machine for media player (mid-completion)	10
State-machine-2	State machine for an ATM (some details missing)	22
Process-model-0	Paper publication process (initial)	2
Process-model-1	CI/CD pipeline (mid-completion)	10
Process-model-2	Order fulfilment (some details missing)	20

Effectiveness: Dataset

120 assistance requests of 4 categories (2 requests per category per model)

- **command-like**: requested elements explicitly described
  - *"create two classes with names Country and City, where Country has..."*
  - *"create two states connected by a time-triggered transition"*
- **model-based**: model modifications
  - *"translate the current model into Spanish"*
  - *"create a composite state that contains the selected elements"*
- **recommendation**: model completions
  - *"recommend fields for each selected page"*
  - *"recommend more substates for Operational"*
- **domain-descriptive**: high-level domain descriptions
  - *"finish the metamodel for Petri nets, considering time in transitions"*
  - *"create the states and transitions for a traffic light that has both lights for cars and for pedestrians, and that has time-triggered transitions"*

Effectiveness: Syntactic Correctness

	OK	Miss att	Miss ref	KO
	gpt-5-mini			
Ecore	75.00	0.00	25.00	0.00
Mind-maps	100.00	0.00	0.00	0.00
Wizards	83.33	8.33	8.33	0.00
State machines	100.00	0.00	0.00	0.00
Process models	95.83	0.00	4.17	0.00
Total	90.83	1.67	7.50	0.00
	gpt-4.1-mini			
Ecore	95.83	0.00	4.17	0.00
Mind-maps	83.33	0.00	16.67	0.00
Wizards	79.17	0.00	20.83	0.00
State machines	100.00	0.00	0.00	0.00
Process models	91.67	4.17	4.17	0.00
Total	90.00	0.83	9.17	0.00
	gpt-4o-mini			
Ecore	79.17	0.00	8.33	12.50
Mind-maps	95.83	0.00	0.00	4.17
Wizards	87.50	0.00	0.00	12.50
State machines	87.50	0.00	0.00	12.50
Process models	70.83	0.00	12.50	16.67
Total	84.17	0.00	4.17	11.67
	gemini-2.5-flash-lite			
Ecore	58.33	0.00	41.67	0.00
Mind-maps	100.00	0.00	0.00	0.00
Wizards	91.67	4.17	4.17	0.00
State machines	91.67	0.00	8.33	0.00
Process models	70.83	0.00	29.17	0.00
Total	82.50	0.83	16.67	0.00

- Most fragments are syntactically correct [82.5–90.83%]
- A few have syntactic errors [4.17–17.5%]
  - missing attributes
  - missing references
- Only gpt-4o-mini had KO's (*"translate into Spanish"* considered off-topic)

Effectiveness: Redundancy

	No	Some	All
	gpt-5-mini		
Ecore	95.80	0.00	4.20
Mind-maps	70.83	12.50	16.67
Wizards	87.50	8.33	4.17
State machines	79.16	0.00	20.83
Process models	87.50	0.00	12.50
Total	84.17	4.17	11.67
	gpt-4.1-mini		
Ecore	95.83	0.00	4.17
Mind-maps	100.00	0.00	0.00
Wizards	100.00	0.00	0.00
State machines	100.00	0.00	0.00
Process models	87.50	0.00	12.50
Total	96.67	0.00	3.33
	gpt-4o-mini		
Ecore	100.00	0.00	0.00
Mind-maps	91.30	4.34	4.34
Wizards	90.47	9.52	0.00
State machines	95.24	4.76	0.00
Process models	90.00	10.00	0.00
Total	93.40	5.66	0.95
	gemini-2.5-flash-lite		
Ecore	100.00	0.00	0.00
Mind-maps	100.00	0.00	0.00
Wizards	100.00	0.00	0.00
State machines	100.00	0.00	0.00
Process models	83.33	0.00	16.67
Total	96.67	0.00	3.33

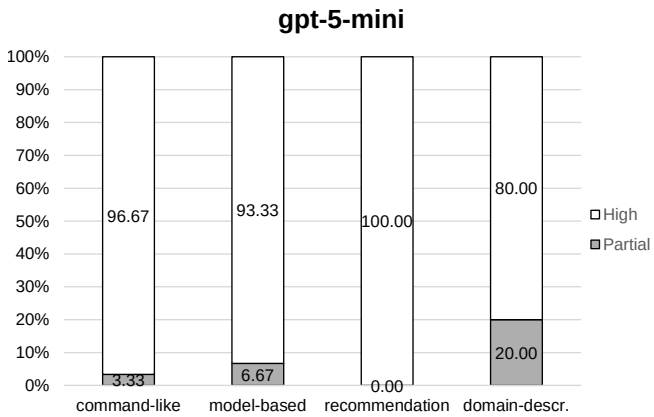
- Most fragments have no redundancy (no unnecesary objects repeated from the model) [84.17–96.67%]

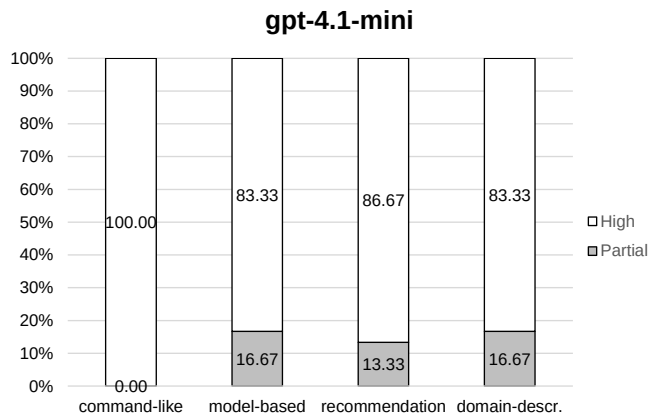
Effectiveness: Semantic Correctness

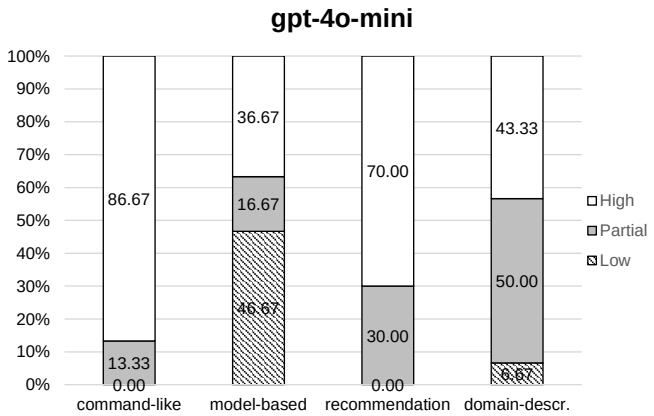
	Low	Partial	High
	gpt-5-mini		
Ecore	0.00	4.17	95.80
Mind-maps	0.00	4.17	95.80
Wizards	0.00	12.50	87.50
State machines	0.00	8.33	91.67
Process models	0.00	8.33	91.67
Total	0.00	7.50	92.5
	gpt-4.1-mini		
Ecore	0.00	33.33	66.67
Mind-maps	0.00	8.33	91.67
Wizards	0.00	4.17	95.83
State machines	0.00	4.17	95.83
Process models	0.00	8.33	91.67
Total	0.00	11.67	88.33
	gpt-4o-mini		
Ecore	12.50	25.00	62.50
Mind-maps	4.17	20.83	75.00
Wizards	12.50	45.83	41.67
State machines	16.67	20.83	62.50
Process models	20.83	25.00	54.17
Total	13.33	27.50	59.17
	gemini-2.5-flash-lite		
Ecore	4.17	37.50	58.33
Mind-maps	0.00	12.50	87.50
Wizards	8.33	25.00	66.67
State machines	16.67	25.00	58.33
Process models	4.17	25.00	70.83
Total	6.67	25.00	68.33

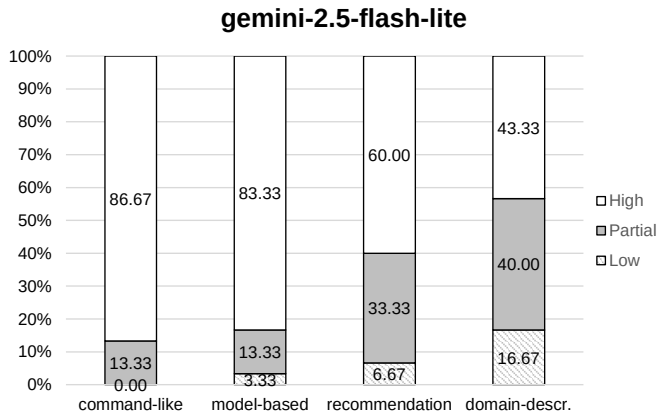
- Many fragments have high-fidelity [59.17–92.5%]
- LLMs with no low-fidelity fragments:
  - gpt-5-mini
  - gpt-4.1-mini
- Meta-models with fewer high-fidelity fragments:
  - Ecore
  - Wizards

Effectiveness: Semantic Correctness



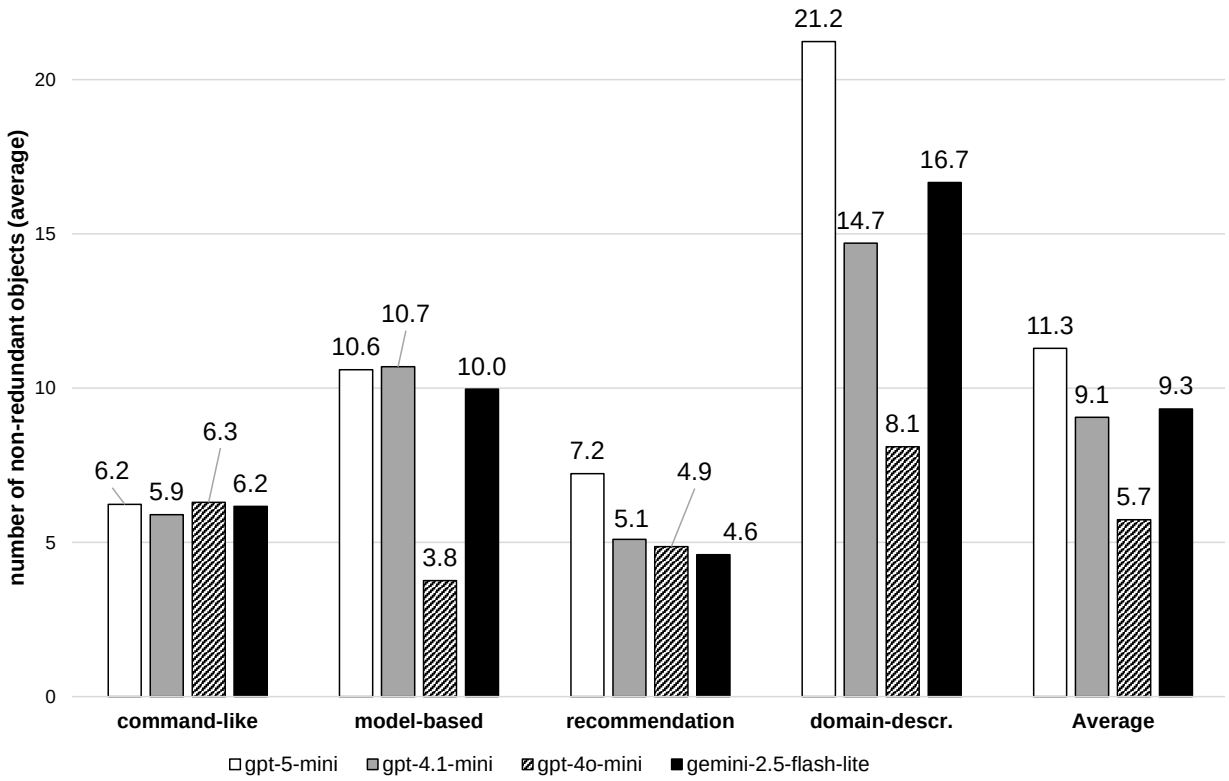






- Command-like: easiest
- Domain-descr.: most difficult

Effectiveness: Size



- Domain-descr.: biggest fragments (up to 92 objects)
- Command-like, recommendation: smallest fragments

Effectiveness: Summary

- Good syntactic correctness [82.5%–90.83%]
- Low redundancy [84.17%–96.67% with no redundancy]
- Good semantic fidelity [59.17%–92.5%]
- Very few fragments with low fidelity [0%–13.33%]
- Fragments of substantial size, especially for domain-descr. reqs

→ EMF-Kaizen is effective for (meta-)modelling assistance

Evaluation:

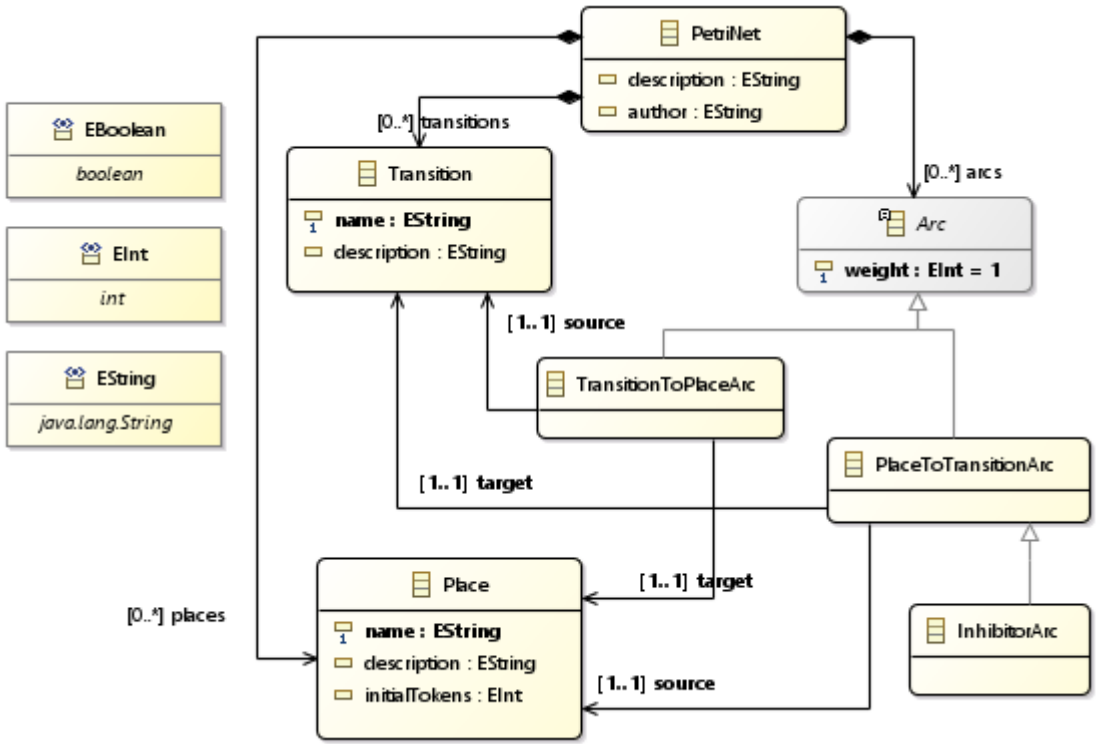
Effect of LLM

Effect of LLM: Summary

- Substantial effect on the LLM used
- gpt-5-mini:
  - best syntactic correctness and semantic accuracy
  - biggest fragments
  - ...but the slowest one (5x): ≈31s
- gpt-4.1-mini:
  - second best performing LLM (by a narrow margin)
  - ...but much faster: ≈6s
  - ...and fewer redundant fragments

Effect of LLM: Example

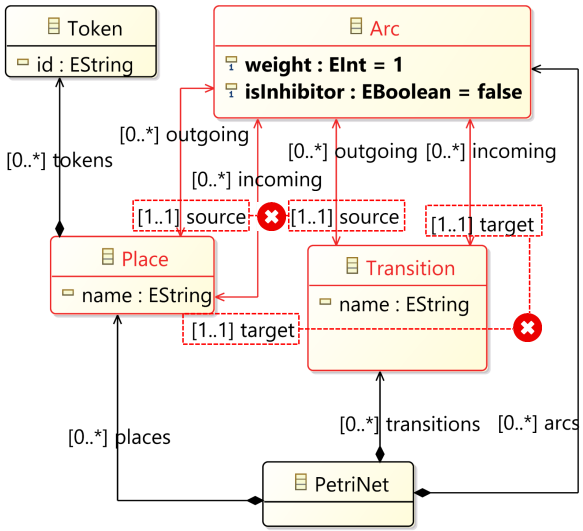
Given Ecore meta-model with 2 elements, *"complete this meta-model of Petri nets, with all missing details, and to consider inhibitor arcs"*



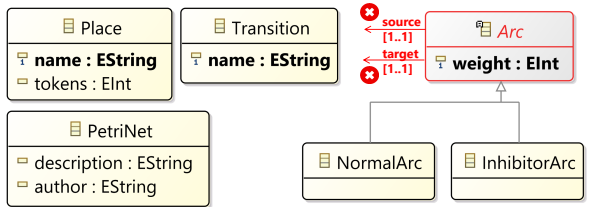
gpt-5-mini (high fidelity)

Effect of LLM: Example

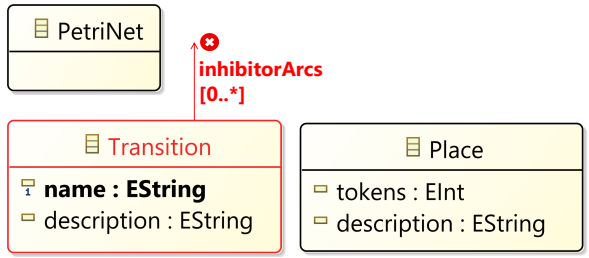
Given Ecore meta-model with 2 elements, *"complete this meta-model of Petri nets, with all missing details, and to consider inhibitor arcs"*



gemini-2.5-flash-lite



gpt-4.1-mini



gpt-4o-mini

The fragments have partial fidelity, and contain references with empty type.

Evaluation:

Effect of prompt design

Effect of Prompt Design: Ablation Study

- LLM: gpt-4o-mini
- 2 meta-models:
  - adapters: low-resource, domain-specific, very specialised
  - UML2: 256 classes, 17700 LoC, focus on class diagrams
- Prompt variations:
  - complete prompt
  - no NL meta-model description
  - no example models
  - no meta-model filtering

Effect of No NL Meta-model Description

	Syntax				Redundancy			Semantic		
	OK	M att	M ref	KO	No	Some	All	Low	Partial	High
	Complete prompt									
Adapters	100.00	0.00	0.00	0.00	100.0	0.00	0.00	8.33	50.00	41.67
UML2	33.33	0.00	66.67	0.00	100.0	0.00	0.00	4.16	33.33	62.5
	No NL meta-model description									
Adapters	100.00	0.00	0.00	0.00	100.0	0.00	0.00	33.33	41.67	25
UML2	33.33	0.00	66.67	0.00	100.0	0.00	0.00	8.33	29.17	62.5

- Adapters: Decrease of high fidelity fragments (from 41.67 to 25%)
- UML2: Small impact

→ **NL meta-model description clarifies domain-specific terms**

Effect of No Example Models

	Syntax				Redundancy			Semantic		
	OK	M att	M ref	KO	No	Some	All	Low	Partial	High
	Complete prompt									
Adapters	100.00	0.00	0.00	0.00	100.0	0.00	0.00	8.33	50.00	41.67
UML2	33.33	0.00	66.67	0.00	100.0	0.00	0.00	4.16	33.33	62.5
	No example models									
Adapters	100.00	0.00	0.00	0.00	100.0	0.00	0.00	8.33	62.50	29.17
UML2	33.33	0.00	41.67	25	77.78	5.55	16.67	37.5	16.67	45.83

- Adapters: Decrease of high fidelity fragments (from 41.67 to 29.17%)
- UML2: Decrease of high fidelity fragments (from 62.5 to 45.83%), more off-topic (from 0 to 25%), more redundancy (from 0 to 22.22%)
- Why? Incorrect encoding of references in JSON (no XMI:id)

→ **Example models illustrate the JSON encoding of models**

Effect of No Meta-model Filtering

	Syntax				Redundancy			Semantic		
	OK	M att	M ref	KO	No	Some	All	Low	Partial	High
	Complete prompt									
Adapters	100.00	0.00	0.00	0.00	100.0	0.00	0.00	8.33	50.00	41.67
UML2	33.33	0.00	66.67	0.00	100.0	0.00	0.00	4.16	33.33	62.5
	No meta-model filtering									
Adapters	100.00	0.00	0.00	0.00	100.0	0.00	0.00	8.33	50.00	41.67
UML2	–	–	–	–	–	–	–	–	–	–

- UML2: the meta-model exceeds the context window of gpt-4o-mini
- Filtering to consider only class diagrams fixes the problem

→ **Filtering reduces large meta-models, yielding smaller prompts**

Conclusions

Summary

- Approach to LLM-based modelling assistance that is:
  - conversational
  - domain-independent (arbitrary modelling languages)
  - meta-level independent (models and meta-models)
  - notation-independent (abstract syntax)
- EMF-Kaizen: realisation for EMF/Eclipse
- Evaluation on several domains and scenarios

Lessons Learnt

- Which LLM to use?
  - faster for simple requests
  - reasoning for complex requests
  - dynamic selection of LLM?
- The request formulation matters
  - ✗ *"suggest 2 tasks for this process"*: task1, task2
  - ✓ *"suggest 2 tasks for this educational process"*: research, presentation
- Identify the application context
  - *"suggest attributes for this class"* → which class???
  - we include the XMI:id of selected objects in the prompt

Future Work

- Trace the recommended model fragments into model (ongoing)
- OCL is added to the prompt, but not validated
- Cross-references between models and fragments (proxy objects?)
- Integration within other editors (Xtex, Sirius...)
- Support for other tasks (e.g., model repair) and artefacts

EMF-Kaizen: an intelligent assistant for domain-specific modelling and meta-modelling

Lissette Almonte, Jefferson Iván Rengifo, Esther Guerra, Juan de Lara
esther.guerra@uam.es

https://emfkaizen.github.io/

Comments? Questions?