

From Data to Semantics: Automated Aspect Model Creation and Knowledge Graph Integration for Digital Twin

Nazanin Mashhaditafreshi

06.02.2025

Supervisors:

Dr.-Ing. habil. Achim Wagner,

M.Sc. Pascal Rübel,

Dr. Andreas Textor,

M.Sc. Nastaran Moarefvand

smartFactory^{KL}

dfki
ai
Innovative
Fabriksysteme



BOSCH



Overview

- Introduction
- Motivation & Research Questions
- Methodology
- Results
- Limitations & Future Work
- Conclusion

Introduction

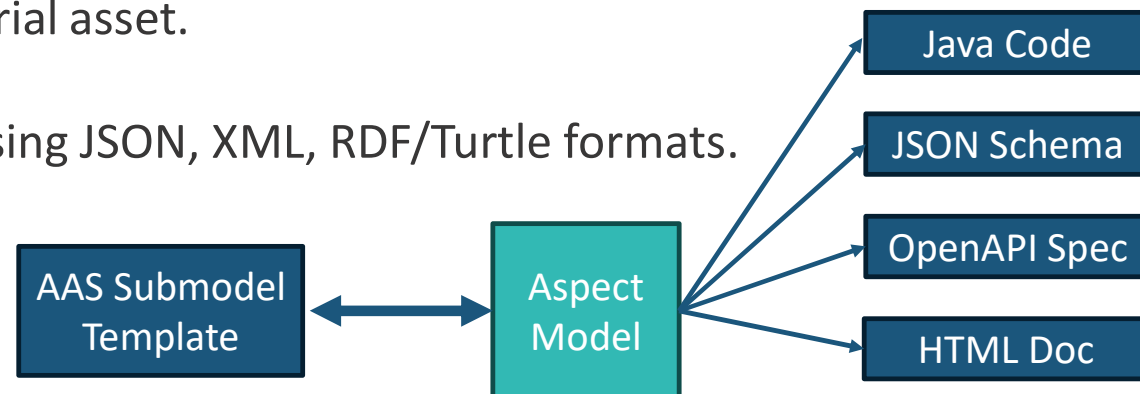
Semantic Aspect Meta Model (SAMM) and **Asset Administration Shell (AAS)** are designed to standardize and enhance data interoperability, particularly within Industry 4.0.

SAMM:

- A core component of the Eclipse Semantic Modelling Framework.
- RDF-based metamodel that semantically describes data within the digital twin.
- Explicitly supports semantic definitions, including units of measurement, and models constraints (ranges, regular expressions, etc.).
- Rich tooling for generating additional artifacts.

AAS:

- The digital representation of an industrial asset.
- Describes semantic via semantic ID
- Enables standardized data exchange using JSON, XML, RDF/Turtle formats.



Example of SAMM Aspect Model

```
{
  "uuidV4Property" : "urn:uuid:48878d48-6f1d-47f5-8ded-a441d0d879df"
}
```

JSON Payload



```
@prefix samm: <urn:samm:org.eclipse.esmf.samm:meta-model:2.1.0#>.
@prefix samm-c: <urn:samm:org.eclipse.esmf.samm:characteristic:2.1.0#>.
@prefix samm-e: <urn:samm:org.eclipse.esmf.samm:entity:2.1.0#>.
@prefix unit: <urn:samm:org.eclipse.esmf.samm:unit:2.1.0#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix : <urn:samm:io.catenax.shared.uuid:2.0.0#>.

:Uuid a samm:Aspect;
  samm:preferredName "Shared Aspect for UUIDs v4"@en;
  samm:description "This is a shared aspect for UUIDs with a regex."@en;
  samm:properties (:uuidV4Property);
  samm:operations ();
  samm:events ().
:uuidV4Property a samm:Property;
  samm:preferredName "UUID v4 Property"@en;
  samm:description "Property based on a UUID v4."@en;
  samm:characteristic :UuidV4Trait;
  samm:exampleValue "urn:uuid:48878d48-6f1d-47f5-8ded-a441d0d879df".
:UuidV4Trait a samm-c:Trait;
  samm:preferredName "Trait for UUIDs v4"@en;
  samm:description "Trait to ensure UUID v4 data format."@en;
  samm-c:baseCharacteristic :Uuidv4Characteristic;
  samm-c:constraint :Uuidv4RegularExpression.
:Uuidv4Characteristic a samm:Characteristic;
  samm:preferredName "UUID v4"@en;
  samm:description "A version 4 UUID is a universally unique identifier that is generated using random 32
  samm:dataType xsd:string;
  samm:see <https://tools.ietf.org/html/rfc4122>.
:Uuidv4RegularExpression a samm-c:RegularExpressionConstraint;
  samm:preferredName "UUID v4 Regular Expression"@en;
  samm:description "The provided regular expression ensures that the UUID is composed of five groups of c
  samm:see <https://datatracker.ietf.org/doc/html/rfc4122>;
  samm:value "^(^([0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})$)|(^urn:uuid:"
```

RDF/Turtle



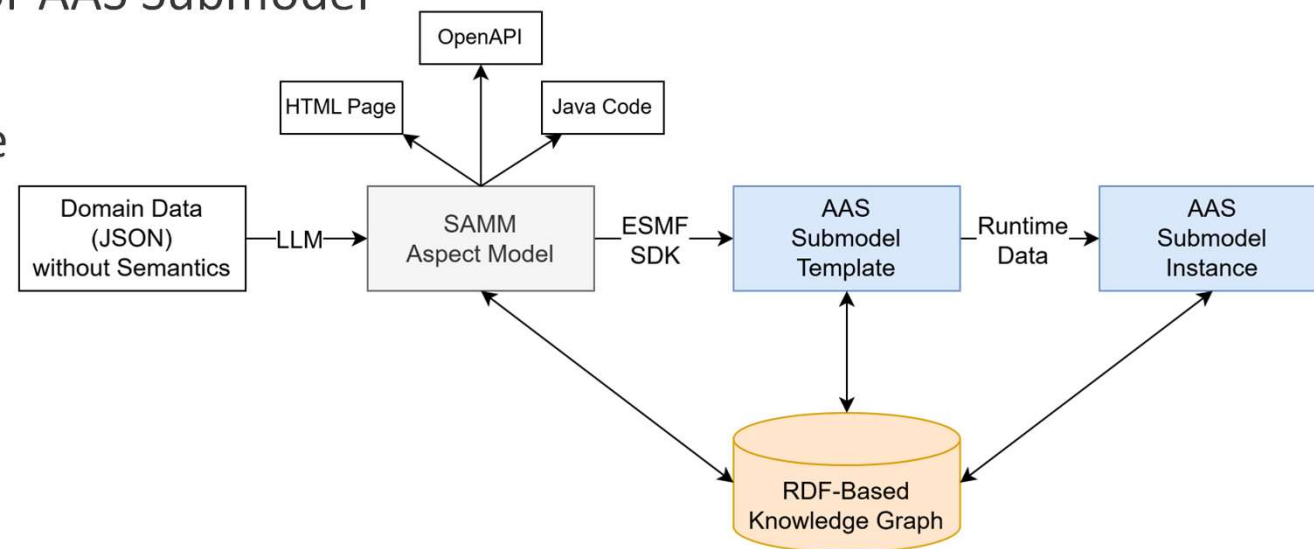
SAMM Elements

A	Aspect The root element of each Aspect Model
AP	Abstract Property Abstract named value
P	Property Named Value
C	Characteristic The meaning of a Property in the conte...
AE	Abstract Entity The abstraction of a logical encapsulati...
E	Entity The logical encapsulation of multiple va...
U	Unit A definition of a physical unit
Co	Constraint A limitation applied to a Characteristic
T	Trait Encapsulates multiple limitations to Ch...
O	Operation An Operation represents an action that ...
Ev	Event A definition of an Event supported by th...

Motivation

Problem

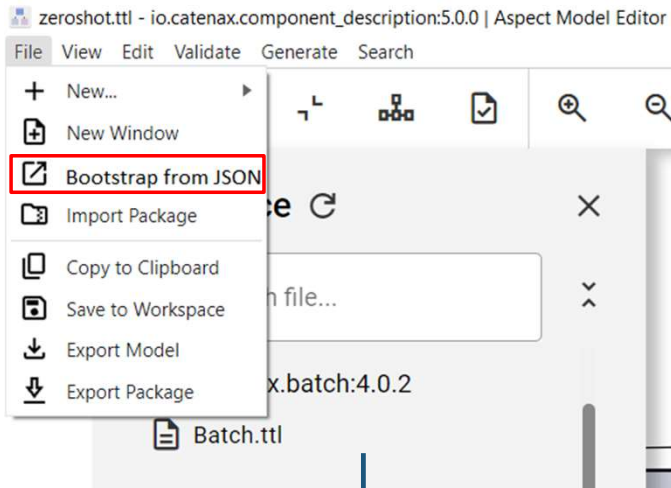
- Creating SAMM Aspect Model or AAS Submodel for existing assets
 - Requires modeling expertise
 - Needs domain knowledge
 - Is hard and time consuming



Our Solution

- Automate the process of generating Aspect models from existing JSON data

End Goal

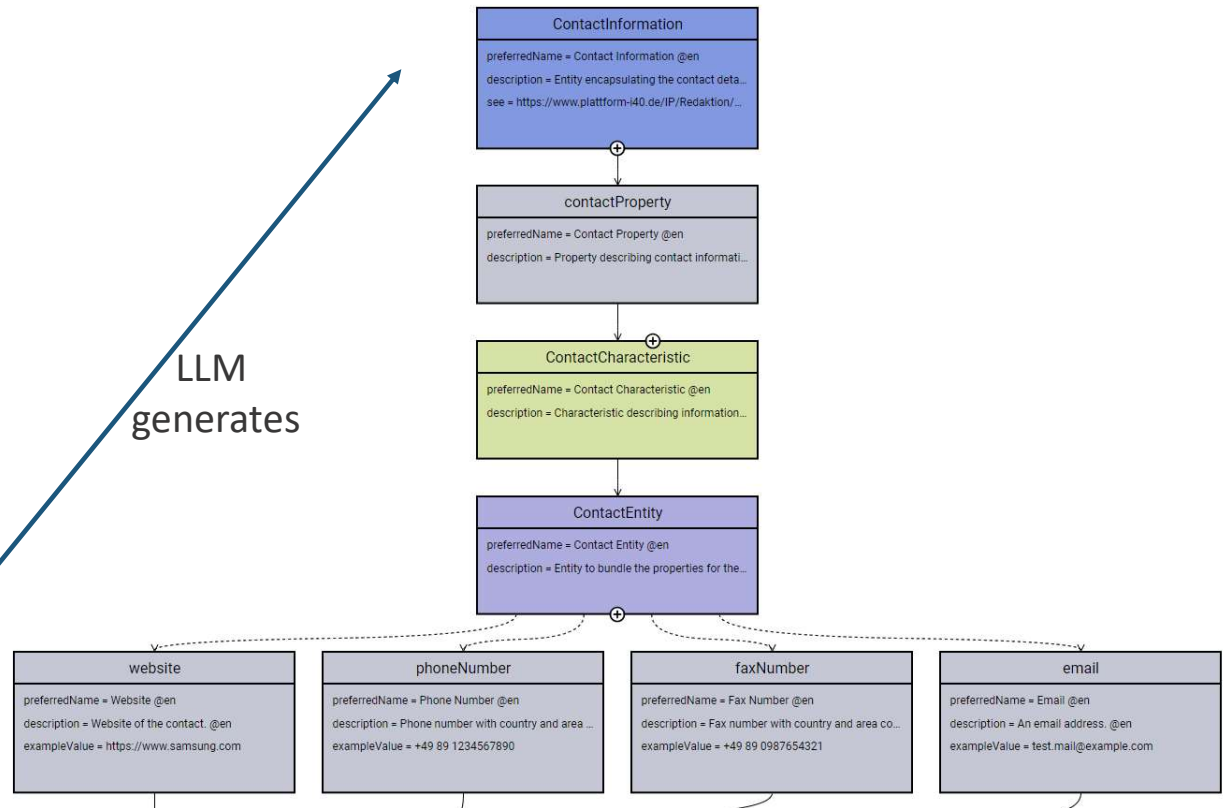


Sample JSON Payload preview

User gives an example

```
{
  "contactProperty": {
    "faxNumber": "+49 89 0987654321",
    "website": "https://www.samsung.com",
    "phoneNumber": "+49 89 1234567890",
    "email": "test.mail@example.com"
  }
}
```

LLM generates

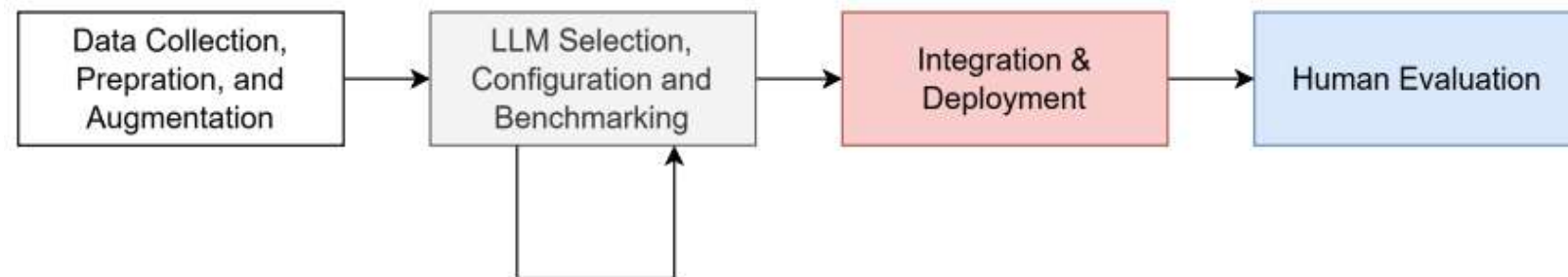


Research Questions

- **RQ1:** How can existing domain data in JSON format be leveraged to automatically or semi-automatically derive the basic structure of Aspect Models within SAMM?
- **RQ2:** How do open-source LLMs compare to commercial solutions, such as OpenAI's models, in generating Aspect Models?
- **RQ3:** What automated methods can be developed to evaluate the Aspect Models generated by LLMs?
- **RQ4:** To what extent can data augmentation techniques improve the accuracy of LLMs when creating Aspect Models from domain data?
- **RQ5:** How can an LLM be integrated into various end-user tools and workflows, including but not limited to an Aspect Model Editor, to help domain experts?
- **RQ6:** How can SAMM and AAS be integrated to form a Knowledge Graph-based digital twin for enhanced data interoperability and decision-making?

Methodology

- Data Preparation
 - Collection: Aspect Models from Eclipse Tractus-X Semantic Layer (SLDT) GitHub repository
 - Cleanup: Fix issues in the repository, remove unnecessary data, etc.
 - Augmentation: Remove and rename properties randomly
 - Split based on the Aspect Model name
- LLM Selection, Configuration, and Benchmarking
- Integration & Deployment
- Human Evaluation



Final Dataset

Test Data = 20% of Data

Validation Data = 10% of Remaining Data

	Original-Cleaned Data	Synthetic Data
Train Data	110	241
Validation Data	13	27
Test Data	32	96
Total	155	364

LLM Models and Fine-Tuning

- **Cloud Fine-Tuning** (Azure AI Studio, OpenAI Platform, AWS Bedrock, ...)

- Low entry cost
- Fully managed with various billing options
- GPT4o-mini

- **Local Fine-Tuning** (PyTorch, HuggingFace, Unsloth, ...)

We used Unsloth library for fine-tuning due to its efficiency and lower resource consumptions.

- Deployment locally via Ollama
- Chat interface via LibreChat, ...
- Llama3.1-8b, Qwen2.5-Coder-7b, Llama3.2-3b, and CodeLlama-7b

Create a fine-tuned model

Method
Specify the method to be used for fine-tuning.
Supervised

Base Model
gpt-4o-mini-2024-07-18

Training data
Add a jsonl file to use for training. By providing the file, you confirm that you have the rights to use the data.
☐ Upload new ☒ Select existing [Browse files](#)
file-QqIFROInqO6EcW3agq6b7HXU

Validation data
Add a jsonl file to use for validation metrics.
☐ Upload new ☒ Select existing ☐ None [Browse files](#)
file-ffOjzPz7a6hJaFlaxBsZ9KOW

Suffix
Add a custom suffix that will be appended to the output model name.
samm-augmented-v1

Seed
The seed controls the reproducibility of the job. Passing in the same seed and job parameters should produce the same results, but may differ in rare cases. If a seed is not specified, one will be generated for you.
31

Configure hyperparameters
☐ Batch size [?](#)
☐ Learning rate multiplier [?](#)
☐ Number of epochs [?](#)

[Learn about fine-tuning](#) [Cancel](#) [Create](#)

OpenAI's dashboard to create a fine-tuning job.

Zero Shot Prompt Template

- You are a bot to help people create Semantic Aspect Meta Model (SAMM) from given JSON data. Create SAMM model based on the following JSON:
- JSON:
<Domain Data (JSON)>
- Provide only the SAMM model without explanation. Make sure you always give a valid RDF turtle as the SAMM model.

Few Shot Prompt Template (One Example)

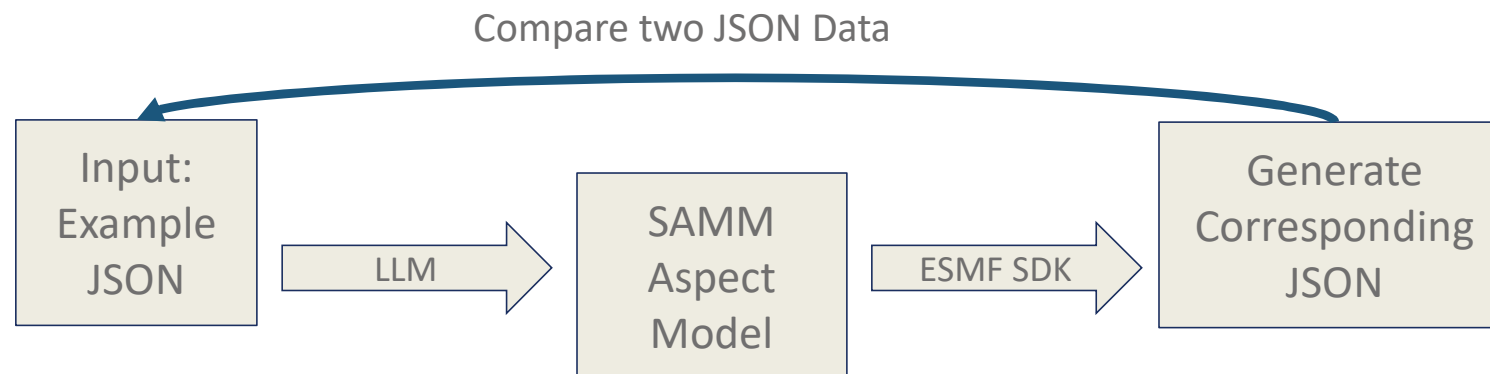
- This is an example SAMM model:
<SAMM Aspect Model (turtle)>
- This is its corresponding JSON example:
<JSON Payload of Aspect Model>
- Your task is to create a SAMM model from a JSON Example.
- Json Example:
<Domain Data (JSON)>
- Provide only the SAMM model without explanation, make sure that the output is a valid RDF Turtle format.

Few Shot Prompt Template (Two Examples)

- This is an example SAMM model:
 <SAMM Aspect Model (turtle) #1>
- This is its corresponding JSON example:
 <JSON Payload of Aspect Mode #1>
- This is an example SAMM model:
 <SAMM Aspect Model (turtle) #2>
- This is its corresponding JSON example:
 <JSON Payload of Aspect Mode #2>
- Your task is to create a SAMM model from a JSON Example.
- Json Example:
 <Domain Data (JSON)>
- Provide only the SAMM model without explanation, make sure that the output is a valid RDF Turtle format.

Evaluation

- Produce Valid Turtle (Valid Turtle)
- Produce Valid SAMM (Valid SAMM)
- Generated JSON structurally similar to the input JSON (Correct)
- Human Evaluation (Complete)

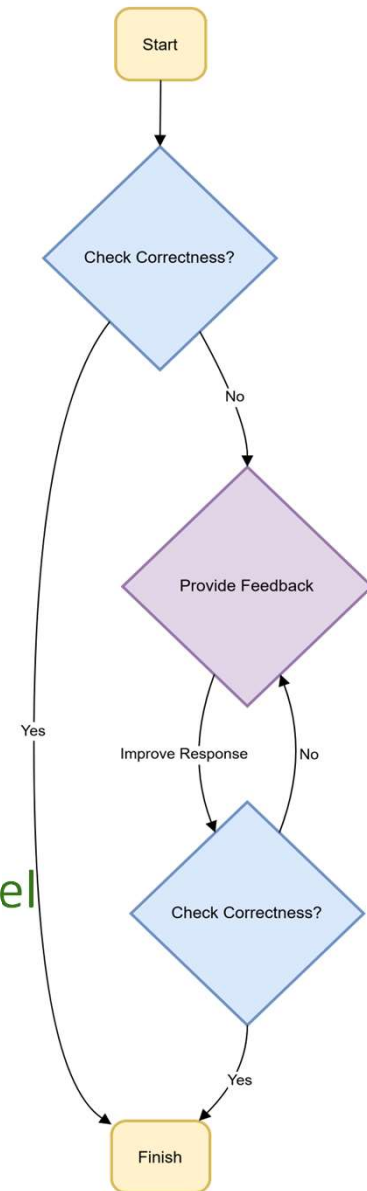


Iterative Prompting

Ask the model to correct itself by providing feedbacks.

Iterative Prompting Template:

- In your previous attempt you created this Semantic Aspect Meta Model (SAMM Aspect Model):
<Previous SAMM Aspect Model generated by the LLM>
- But it has the following error:
<Previous Error Messages>
- Try to fix the error and generate the whole corrected SAMM Aspect Model without any extra explanation.
<Extra Hints related to the Exception>



Iterative Prompting Example:

2-result.txt

```

1 @prefix samm: <urn:samm:org.eclipse.esmf.samm:meta-model:2.1.0#> .
2 @prefix samm-c: <urn:samm:org.eclipse.esmf.samm:characteristic:2.1.0#> .
3 @prefix samm-e: <urn:samm:org.eclipse.esmf.samm:entity:2.1.0#> .
4 @prefix unit: <urn:samm:org.eclipse.esmf.samm:unit:2.1.0#> .
5 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
6 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
7 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
8 @prefix : <urn:samm:io.catenax.determination_load_data_source:1.0.0#> .
9
10 :DeterminationLoadDataSource a samm:Aspect ;
11     samm:preferredName "Determination Load Data Source"@en ;
12     samm:description "The determination of the load data source is based
13     samm:properties ( :sourceLoadSpectrum ) ;
14     samm:operations ( ) ;
15     samm:events ( ) .
16
17 :sourceLoadSpectrum a samm:Property ;
18     samm:preferredName "Source Load Spectrum"@en ;
19     samm:description "The source load spectrum is the URI to the load sp
20     samm:characteristic samm-c:ResourcePath .

```

3-prompt.txt - Notepad

File Edit Format View Help

java.lang.Exception: JSONs are not similar:
 JSONs are not similar: ground truth has key "DETERMINATIONLOADDATASOURCE" but it is missing
 Add a SAMM "DETERMINATIONLOADDATASOURCE" as a Property or Entity to the Aspect Model.
 Required Ground truth JSON should be like:

```

{
  "DETERMINATIONLOADDATASOURCE" : {
    "sourceLoadSpectrum" : "telnet://192.0.2.16:80/"
  }
}

```

However, the generated Aspect Model corresponds to the following JSON structure:

```

{
  "sourceLoadSpectrum" : "telnet://192.0.2.16:80/"
}

```

Try to fix the error and generate the whole corrected SAMM Aspect Model without any extra exp
 For understanding the construction rules, we define the following terms:

3-result.txt

```

1 @prefix samm: <urn:samm:org.eclipse.esmf.samm:meta-model:2.1.0#> .
2 @prefix samm-c: <urn:samm:org.eclipse.esmf.samm:characteristic:2.1.0#> .
3 @prefix samm-e: <urn:samm:org.eclipse.esmf.samm:entity:2.1.0#> .
4 @prefix unit: <urn:samm:org.eclipse.esmf.samm:unit:2.1.0#> .
5 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
6 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
7 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
8 @prefix : <urn:samm:io.catenax.determination_load_data_source:1.0.0#> .
9
10 :DeterminationLoadDataSource a samm:Aspect ;
11     samm:preferredName "Determination Load Data Source"@en ;
12     samm:description "The determination of the load data source is based
13     samm:properties ( :DETERMINATIONLOADDATASOURCE ) ;
14     samm:operations ( ) ;
15     samm:events ( ) .
16
17 :DETERMINATIONLOADDATASOURCE a samm:Property ;
18     samm:preferredName "Determination Load Data Source"@en ;
19     samm:description "The determination load data source is the aspect t
20     samm:characteristic :DeterminationLoadDataSourceCharacteristic .
21
22 :DeterminationLoadDataSourceCharacteristic a samm:Characteristic ;
23     samm:preferredName "Determination Load Data Source Characteristic"@e
24     samm:description "The determination load data source characteristic
25     samm:datatype :DeterminationLoadDataSourceEntity .
26
27 :DeterminationLoadDataSourceEntity a samm:Entity ;
28     samm:preferredName "Determination Load Data Source Entity"@en ;
29     samm:description "The determination load data source entity represen
30     samm:properties ( :sourceLoadSpectrum ) .
31
32 :sourceLoadSpectrum a samm:Property ;
33     samm:preferredName "Source Load Spectrum"@en ;
34     samm:description "The source load spectrum is the URI to the load sp
35     samm:characteristic samm-c:ResourcePath .
36

```

Experiment 1: Effect of Examples and Number of Attempts

- Counting the number of metamodel elements used in the Aspect Model, such as Property, Characteristics, Trait, Enumeration, etc.
- A more complete example improved the result by 4%.

Model	Valid Turtle	Valid SAMM	Correct
Simple example	84 (87%)	34 (35%)	28 (29%)
Complex example	85 (88%)	41 (42%)	32 (33%)

- Increasing the number of attempts improved performance.

Model	Attempt 1	Attempt 2	Attempt 3
Simple example	21 (21%)	25 (26%) (+4)	28 (29%) (+3)
Complex example	27 (28%)	31 (32%) (+4)	32 (33%) (+1)

Simple example: io.catenax.waste

Complex example: io.catenax.secondary_material_content

Experiment 2: Comparison of Llama3.1-8b, Qwen2.5-Coder-7b, Llama3.2-3b, and CodeLlama-7b without Fine-Tuning

- Code-based LLMs are more capable of generating valid Turtle Aspect Models.
- In the absence of SAMM knowledge, they generate arbitrary RDF Turtle statements, but not a valid SAMM Aspect Model.

Model	Valid Turtle	Valid SAMM	Correct
Qwen2.5-Coder 7B	39 (40%)	0	0
CodeLlama 7B	31 (32%)	0	0
Llama 3.1 8B	29 (30%)	0	0
Llama 3.2 3B	5 (5%)	0	0

Experiment 3: Effect of More Shots on Qwen2.5-coder-7b and Llama3.1-8b

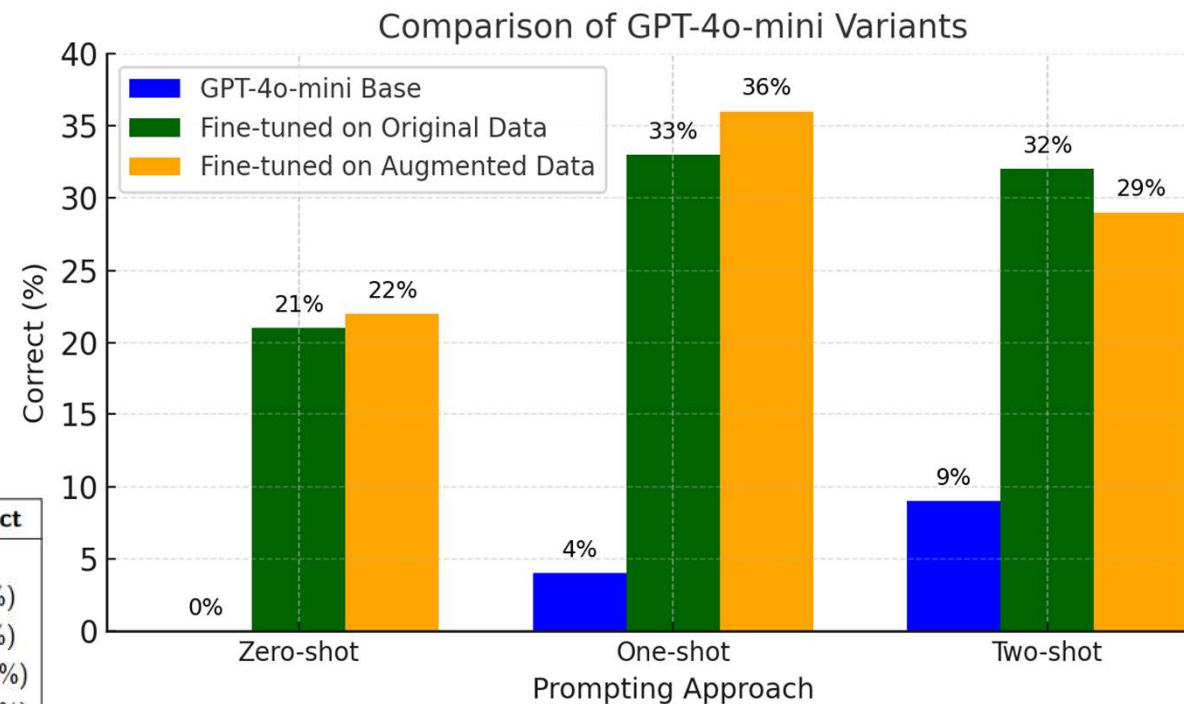
- Two-shot prompting improves the performance of the base LLM compared to one-shot.

Model	Valid Turtle	Valid SAMM	Correct
Qwen2.5-Coder 7B (two-shot)	38 (39%)	12 (12%)	3
Qwen2.5-Coder 7B (one-shot)	39 (40%)	0	0
Llama 3.1 8B (two-shot)	24 (25%)	4 (%)	0
Llama 3.1 8B (one-shot)	29 (30%)	0	0

Experiment 4: Comparison of GPT-4o-mini Base, Fine-Tuned on Original Data, and Fine-Tuned on Augmented Data

- The augmented dataset improved results by 3% (one-shot) but doubled the training cost.
- Our data augmentation introduces minimal variations with simplifying the model, this does not teach new concepts to the LLM.

Model	Valid Turtle	Valid SAMM	Correct
GPT-4o-mini (zero-shot)	75 (78%)	0	0
GPT-4o-mini (one-shot)	88 (91%)	24 (25%)	4 (4%)
GPT-4o-mini (two-shot)	83 (86%)	34 (35%)	9 (9%)
GPT-4o-mini fine-tuned on original data (zero-shot)	79 (82%)	20 (20%)	21 (21%)
GPT-4o-mini fine-tuned on original data (one-shot)	85 (88%)	41 (42%)	32 (33%)
GPT-4o-mini fine-tuned on original data (two-shot)	80 (83%)	39 (40%)	31 (32%)
GPT-4o-mini fine-tuned on augmented data (zero-shot)	67 (69%)	25 (26%)	22 (22%)
GPT-4o-mini fine-tuned on augmented data (one-shot)	82 (85%)	43 (44%)	35 (36%)
GPT-4o-mini fine-tuned on augmented data (two-shot)	75 (78%)	36 (37%)	28 (29%)



Experiment 5: Iterative Prompting on Fine-tuned GPT-4o-mini

- Iterative prompting is more effective than simple retry.
- Providing extra information enhances the LLM's performance in iterative prompting.

Model	Attempt 1	Attempt 2	Attempt 3
GPT-4o-mini (one-shot) using simple retry	27 (28%)	31 (32%) (+4)	32 (33%) (+1)
GPT-4o-mini (one-shot) using iterative feedback	33 (34%)	41 (42%) (+8)	42 (43%) (+1)

Summary of the Results

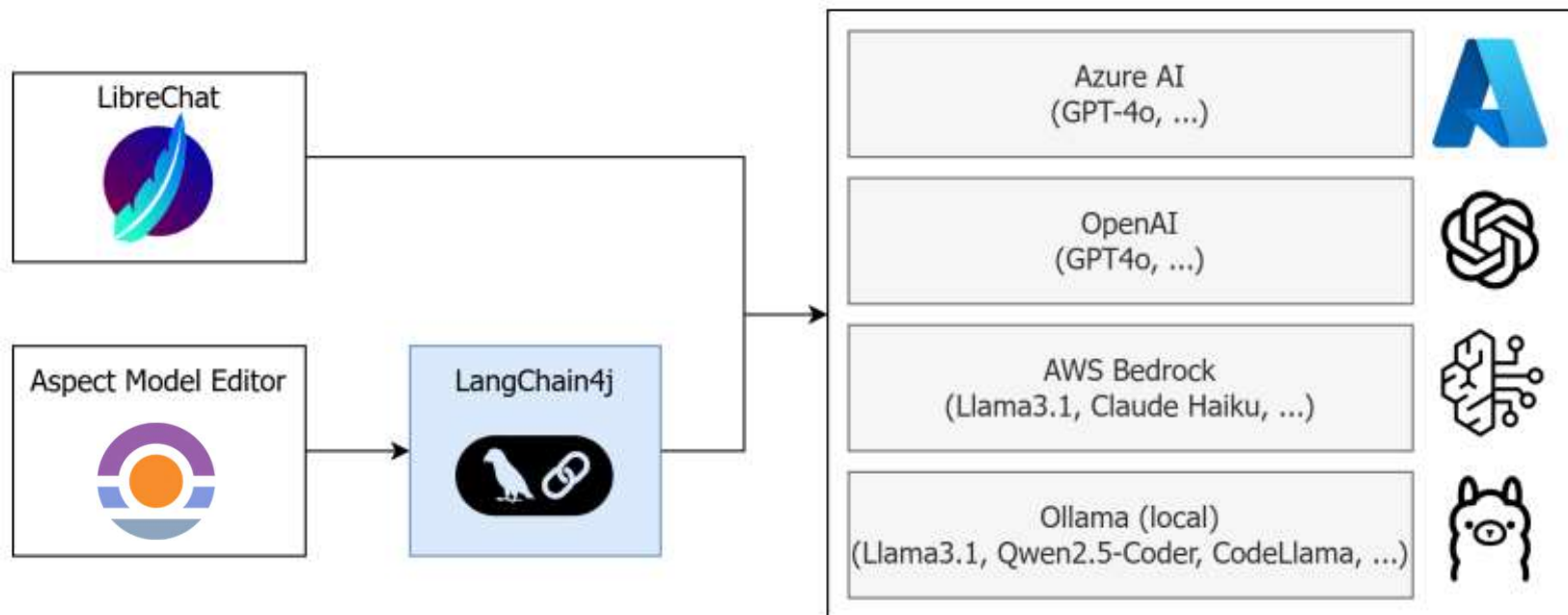
Model	Valid Turtle	Valid SAMM	Correct
Fine-tuned GPT-4o-mini (3 iterations)	86 (89%)	63 (65%)	42 (43%)
Fine-tuned GPT-4o-mini (one-shot)	85 (88%)	41 (42%)	32 (33%)
Fine-tuned GPT-4o-mini (two-shots)	80 (83%)	39 (40%)	31 (32%)
Fine-tuned GPT-4o-mini (zero-shot)	79 (82%)	20 (20%)	21 (21%)
Base GPT-4o-mini (two-shot)	83 (86%)	34 (35%)	9 (9%)
Fine-tuned Qwen2.5-Coder 7B (one-shot)	43 (44%)	10 (10%)	6 (6%)
Base GPT-4o-mini (one-shot)	88 (91%)	24 (25%)	4 (4%)
Base Qwen2.5-Coder 7B (two-shots)	38 (39%)	12 (12%)	3 (3%)
Fine-tuned Qwen2.5-Coder 7B (two-shots)	38 (39%)	6 (6%)	2 (2%)
Fine-tuned Qwen2.5-Coder 7B (zero-shot)	53 (55%)	5 (5%)	1 (1%)
Base Llama 3.1 8B (two-shots)	24 (25%)	4 (%)	0
Base GPT-4o-mini (zero-shot)	75 (78%)	0	0
Base Qwen2.5-Coder 7B (zero-shot)	40 (41%)	0	0
Base Qwen2.5-Coder 7B (one-shot)	39 (40%)	0	0
Base CodeLlama 7B (one-shot)	31 (32%)	0	0
Base Llama 3.1 8B (one-shot)	29 (30%)	0	0
Base Llama 3.2 3B (one-shot)	5 (5%)	0	0

- Iterative prompting improved performance by 10%.
- More shots help base LLMs and confuse fine-tuned LLMs
- Zero-shot works only for fine-tuned LLMs.

Model	Pass@1
 o1-2024-12-17 (temperature=1, reasoning=high) ⭐	35.5
 o3-mini-2025-01-31 (temperature=1, reasoning=medium) ⭐	35.5
 DeepSeek-R1 ⭐	35.1
 o3-mini-2025-01-31 (temperature=1, reasoning=high) ⭐	35.1
 o1-2024-12-17 (temperature=1, reasoning=low) ⭐	34.5

BigCodeBench (hard code generation tasks)
Source: <https://bigcode-bench.github.io/>

Deployment & Integration



Deployment & Integration: SAMM Copilot Platform User Interface

The screenshot displays the SAMM Copilot Platform User Interface. On the left, a sidebar contains a 'New chat' button, a search bar for 'Search messages', and a 'Bookmarks' section. Below these, a 'Previous 7 days' section shows a 'Create SAMM Model from' button. The main chat area shows a conversation with 'GPT-4o'. The user's message, labeled 'test 4', asks the bot to help create a Semantic Aspect Meta Model (SAMM) from given JSON data. The JSON data provided is:

```
{
  "powerConsumption": 3.0,
  "energyConsumption": 7590.0,
  "energyConsumptionStartDate": "2024-10-25T15:25:24Z"
}
```

. The bot's response, also labeled 'GPT-4o', provides the SAMM model in Turtle format. The Turtle code is displayed in a dark-themed code editor with a 'Copy code' button. The code includes prefixes for samm, samm-c, samm-e, unit, rdf, rdfs, xsd, and a specific energy consumption entity. At the bottom of the chat area, there is a 'Message GPT-4o' input field and a microphone icon. A small footer text reads 'SAMM Copilot created by Nazanin Mashhaditafreshi'.

OpenAI ft:gpt-4o-mini-2024-07-18:personal:original-v0:A8wrwtyN

test 4

You are a bot to help people create Semantic Aspect Meta Model (SAMM) from given JSON data. Create SAMM model based on the following JSON:

```
{
  "powerConsumption": 3.0,
  "energyConsumption": 7590.0,
  "energyConsumptionStartDate": "2024-10-25T15:25:24Z"
}
```

Provide only the SAMM model without explanation. Make sure you always give a valid RDF turtle as the SAMM model.

GPT-4o

```
turtle
@prefix samm: <urn:samm:org.eclipse.esmf.samm:meta-model:2.1.0#> .
@prefix samm-c: <urn:samm:org.eclipse.esmf.samm:characteristic:2.1.0#> .
@prefix samm-e: <urn:samm:org.eclipse.esmf.samm:entity:2.1.0#> .
@prefix unit: <urn:samm:org.eclipse.esmf.samm:unit:2.1.0#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-tree#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix : <urn:samm:io.catenax.bi.energy_consumption:2.2.0#> .
```

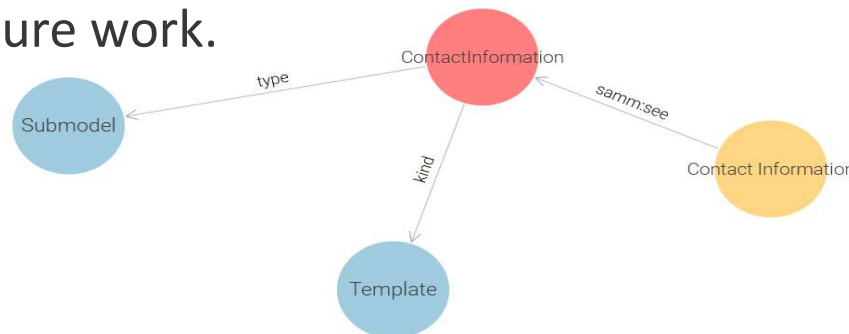
Message GPT-4o

SAMM Copilot created by Nazanin Mashhaditafreshi

<https://sammcopilot.studio/>

SAMM and AAS in an RDF-based Knowledge Graph

- Convert SAMM Aspect Model to AAS Submodel.
- Submodel templates/instances are converted to RDF (according to the AAS specification) using an experimental feature in the Eclipse BaSyx Python SDK.
- This enables the integration of GenAI and Graph Retrieval Augmented Generation (GraphRAG) for applications such as question answering in future work.



Contact Information [🔗](#)

🔗 Contact Information^{en}

Types:

samm:Aspect

RDF Rank:

0

samm:preferredName

Contact Information^{en}

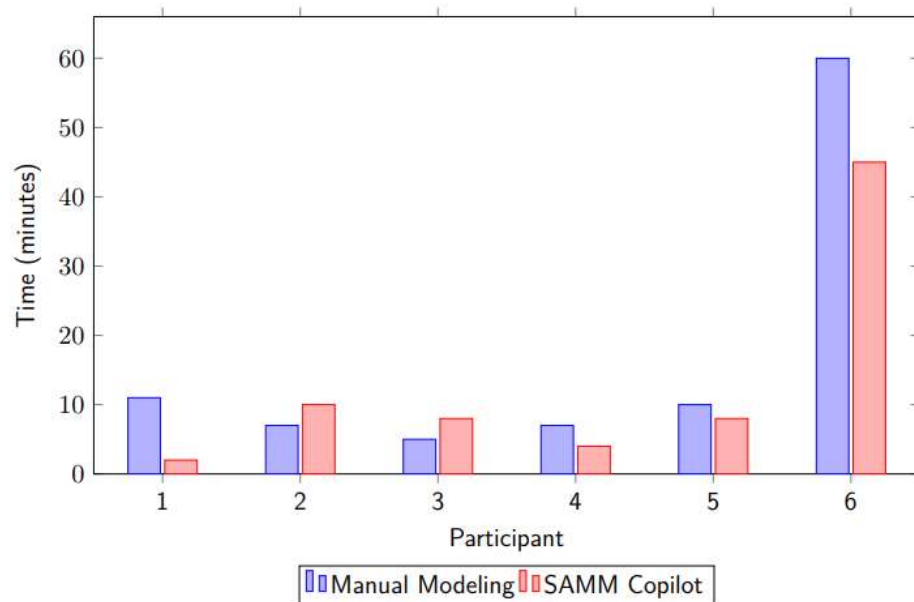
samm:description

Entity encapsulating the contact details.^{en}

GraphDB 10.7.3 • RDF4J 4.3.13 • Connectors 16.2.9 • Workbench 2.7.3 • © 2002–2025 Ontotext AD. All rights reserved.

Human Evaluation

- Manual modeling generally took longer compared to using SAMM Copilot in most cases.
- SAMM Copilot was able to generate valid models in most cases.



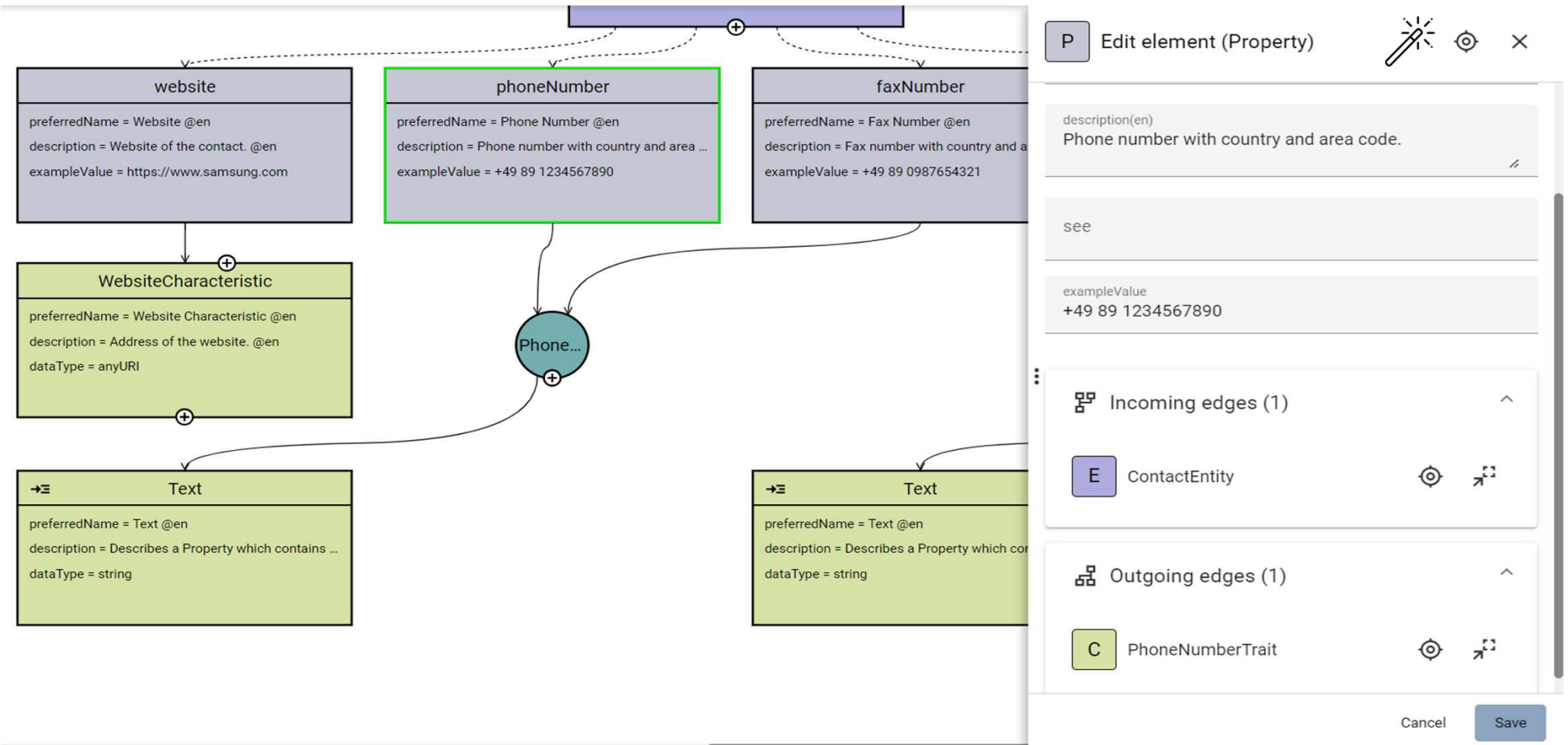
Participant	Manual Modeling Time	External Sources Used	SAMM Copilot Time	Satisfaction with SAMM Copilot
1	11 minutes	No	2 minutes	Yes, valid and complete model
2	7 minutes	No	>10 minutes	No, invalid model
3	5 minutes	None	8 minutes	Yes, valid but incomplete model
4	7 minutes	SAMM specification, Smart Plug Datasheet	4 minutes	Yes, valid but incomplete model
5	60 minutes	ESMF documentation	45 minutes	No, valid but incorrect model
6	10 minutes	No	Less than 3 minutes	No, invalid model

Limitations & Future Work

- Evaluation Scope Limitations:
 - Including further commercial models (Gemini, Claude) and open-source models (DeepSeek, ...)
 - Exploration of various LLM sizes and hyperparameters
- Human Evaluation Constraints:
 - Outputs from different LLMs were not compared through human evaluation.
 - Including more human evaluations.
- Single JSON Example Limitation:

A single JSON example may not fully represent the data due to structural variability.
- Integration of Generative AI approach in Aspect Model Editor.

GenAI Integration in Aspect Model Editor



GenAI Integration in Aspect Model Editor

The screenshot displays the Aspect Model Editor interface. A modal dialog is open, asking "How do you want to enhance this element?". The dialog contains a text area with the suggestion: "The phone number should have an appropriate regular expression". The background shows a diagram with three elements: "website", "phoneNumber" (highlighted with a green border), and "faxNumber". The "phoneNumber" element has a preferred name of "Phone Number @en". The "website" element has a preferred name of "Website @en". The "faxNumber" element has a preferred name of "Fax Number @en". The dialog has a "Generate" button and a "Cancel" button. The "Save" button is visible in the bottom right corner of the editor.

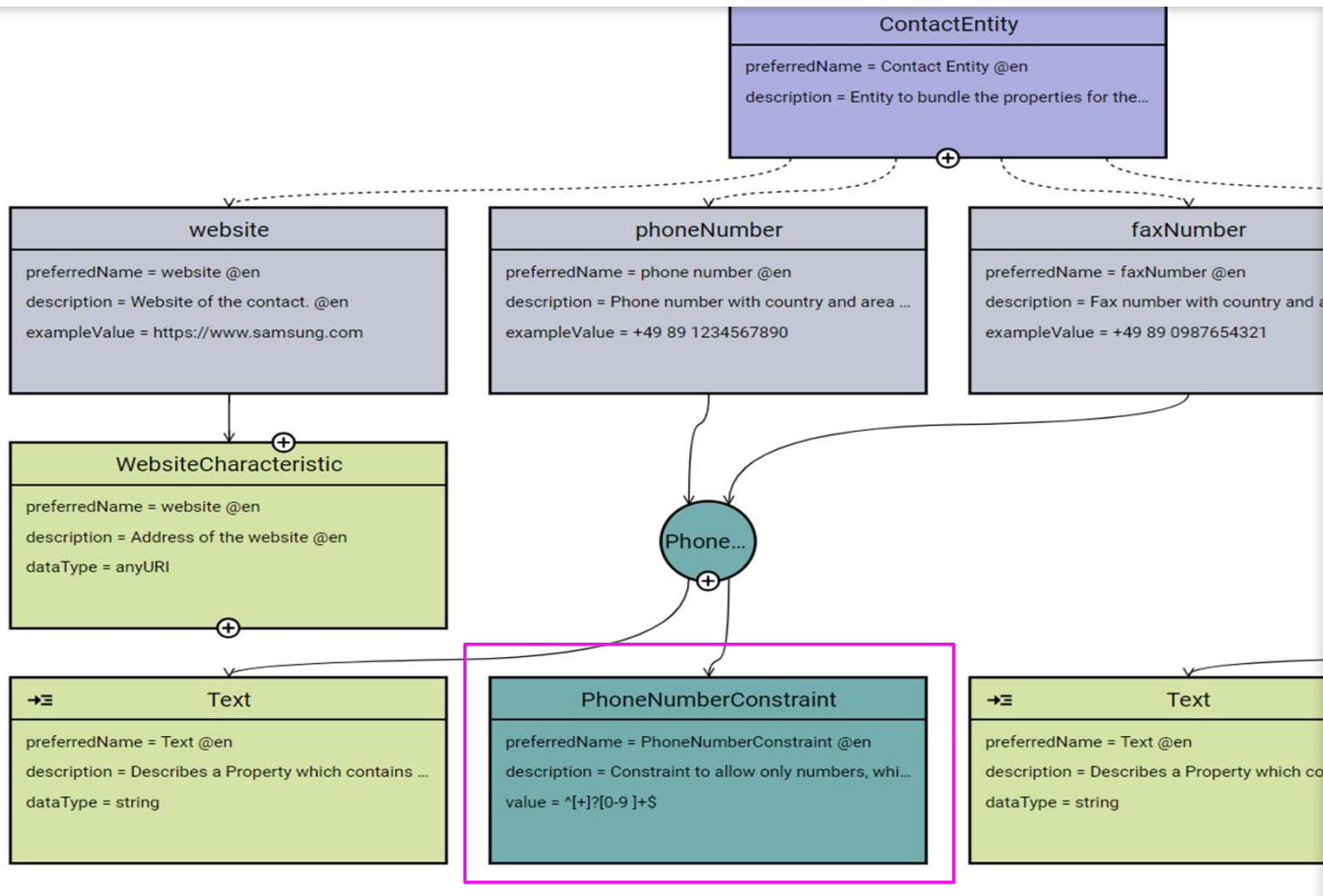
How do you want to enhance this element?

The phone number should have an appropriate regular expression

Generate

Cancel Save

GenAI Integration in Aspect Model Editor



Co Edit element (RegularExpressionConstraint)

Constraint class
RegularExpressionConstraint

Name*
PhoneNumberConstraint

preferredName(en)
PhoneNumberConstraint

description(en)
Constraint to allow only numbers, whitespaces and an optional leading + sign.

see
Type to search or write an URN or URL

value*
^[+]?[0-9]+\$

Incoming edges (1)

PhoneNumberTrait

Cancel Save

Conclusion

- LLMs can automatically generate Aspect models from a sample JSON (RQ1):
 - Reduces time and effort for experts.
 - Eliminates the need to start from scratch.
 - Modeling is up to 4 times faster based on human evaluation.
- Commercial models outperform open-source models (RQ2).
- Methods to evaluate the output of LLMs have been introduced (RQ3).
- Our data augmentation method did not improve accuracy but added value during evaluation (RQ4).
- A chat user interface has been deployed, with further prototype ideas demonstrated (RQ5).
- SAMM Aspect Model and AAS Submodel have been integrated into a Knowledge Graph (RQ6).

[linkedin.com/in/nmtafreshi/](https://www.linkedin.com/in/nmtafreshi/)



From Data To Semantic

or How to use LLMs to bootstrap your semantic model



Thank you for your attention!



Any Questions?