



# AI-Driven Consistency of SysML Diagrams

Bastien Sultan, Ludovic Apvrille

{bastien.sultan, ludovic.apvrille}@telecom-paris.fr

MoDELS '24 – Linz



# Outline

Introduction

Contribution

In practice

Takeaways

# Introduction

## Consistency in UML/SysML Models

- Not really a new research issue!

# Introduction

## Consistency in UML/SysML Models

- Not really a new research issue!
- Formal rules-based approaches
  - Unequivocal, algorithmically verifiable
  - Address a wide spectrum of inconsistencies

# Introduction

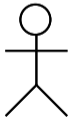
## Consistency in UML/SysML Models

- Not really a new research issue!
- Formal rules-based approaches
  - Unequivocal, algorithmically verifiable
  - Address a wide spectrum of inconsistencies
  - But...

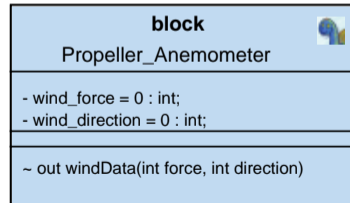
# Introduction

## Consistency in UML/SysML Models

- Not really a new research issue!
- Formal rules-based approaches
  - Unequivocal, algorithmically verifiable
  - Address a wide spectrum of inconsistencies
  - But...



Wind\_Sensor





# Introduction

## AI for MDE

- Maybe you've heard about LLMs?

# Introduction

## AI for MDE

- Maybe you've heard about LLMs?
- More and more integrated into model generation methodologies and tools
  - That raises new internal/cross-view consistency concerns



# Introduction

## AI for MDE

- Maybe you've heard about LLMs?
- More and more integrated into model generation methodologies and tools
  - That raises new internal/cross-view consistency concerns
- LLMs are quite effective at determining whether two texts have the same meaning

# Introduction

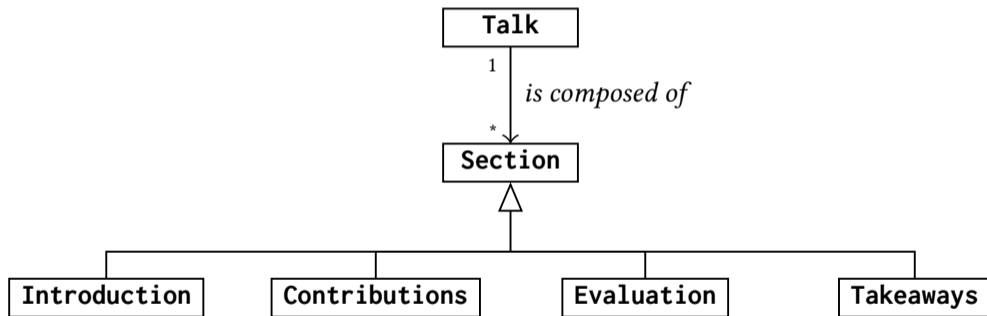
## AI for MDE

- Maybe you've heard about LLMs?
- More and more integrated into model generation methodologies and tools
  - That raises new internal/cross-view consistency concerns
- LLMs are quite effective at determining whether two texts have the same meaning

## Our idea

- Blend formal rules and LLMs in symbiosis to detect and correct internal and cross-view inconsistencies
  - Generate consistent diagrams from a textual specification
  - Enhance the consistency of input diagrams

# Outline





# Outline

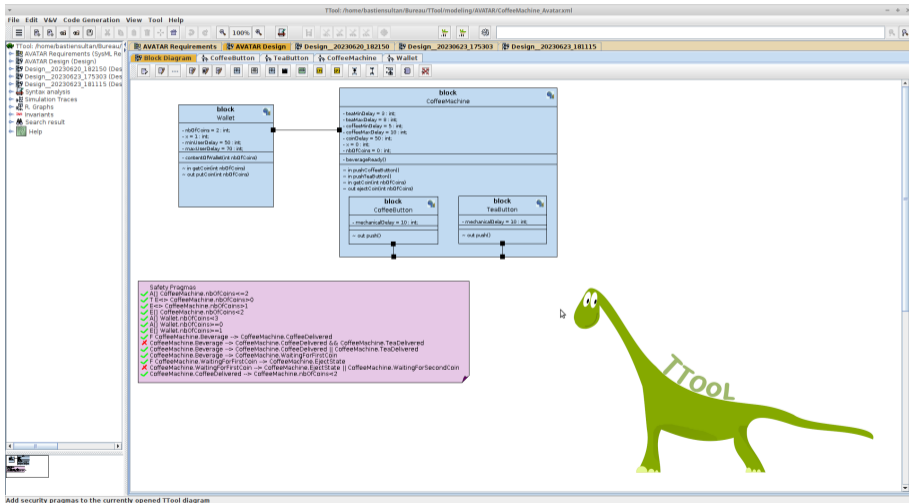
Introduction

**Contribution**

In practice

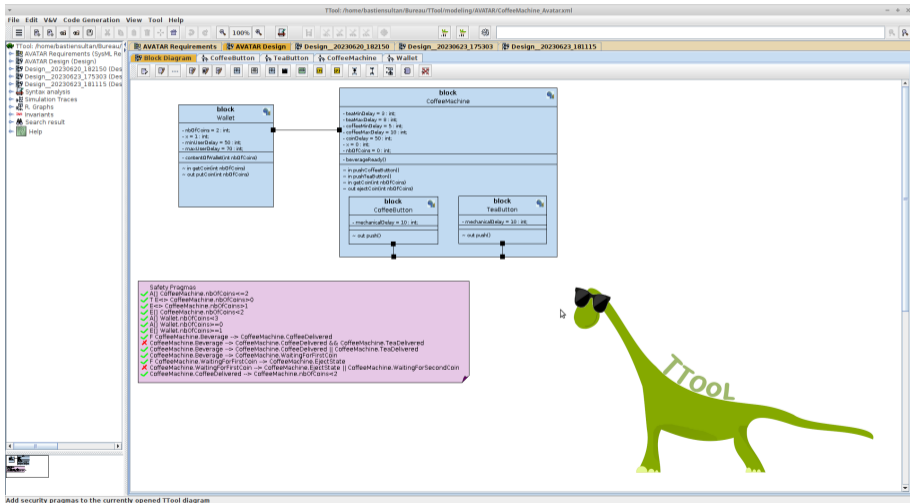
Takeaways

# Some context – TTool



Add security pragmas to the currently opened TTool diagram

## Some context – TTool



# Contribution overview

## Our contribution in a nutshell

- Applicable to any SysML diagram but the paper focuses on UCD/BD consistency

# Contribution overview

## Our contribution in a nutshell

- Applicable to any SysML diagram but the paper focuses on UCD/BD consistency
- Sets of formal consistency rules, both internal (UCD and BD) and external (crossed consistency UCD/BD)
  - You can find them in the paper if you like first-order logic :-)

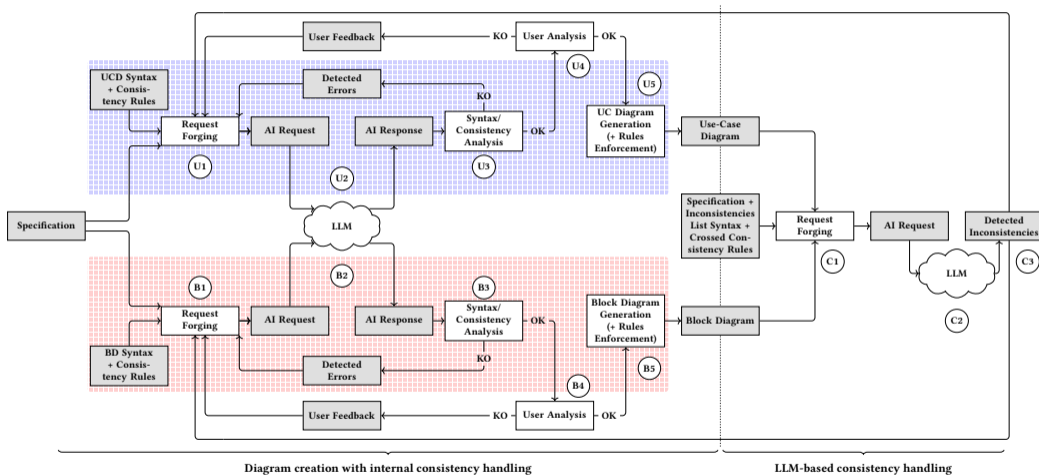


# Contribution overview

## Our contribution in a nutshell

- Applicable to any SysML diagram but the paper focuses on UCD/BD consistency
- Sets of formal consistency rules, both internal (UCD and BD) and external (crossed consistency UCD/BD)
  - You can find them in the paper if you like first-order logic :-)
- Consistency handling methodology blending formal rules and LLMs, and GPT-based implementation in TTool

# Contribution overview



# Focus on LLM-based inconsistency detection – generated LLM prompts

## Textual diagram description

```
actors: User Propeller_Anerometer ...
Use cases: Define_PositionAndCourse ...
Connections: include(Activate_BowThrusters, Maintain_SetPosition ...
```

## Inconsistencies list syntax

When you are asked to identify all the relevant incoherencies between two diagrams, return them as a JSON specification formatted as follows:

```
{incoherencies: [{ \"diagram\" : \"diagram1 or diagram2\", \"description\" : \"description of the incoherency\"}..."]};
```

# Focus on LLM-based inconsistency detection – generated LLM prompts

## Rules

# Respect: In a block diagram, the blocks representing actors as defined in the use case diagram must bear identical names to their corresponding use cases.

# Respect: In a block diagram, blocks representing actors from the use case diagram must not be connected together.

...

# Focus on LLM-based inconsistency detection

System engineering with AI

Options

Selected chat: Diagram coherency

Selected AI model: gpt-4o

Question

```
original attribute: int set_position_latitude = 0
original attribute: int set_position_longitude = 0
original attribute: bool system_status = false
signal: out setPoint(int position,int course)

*** block:DPS
attribute: bool is_active = false
attribute: int desired_course = 0
attribute: int desired_position_latitude = 0
attribute: int desired_position_longitude = 0
original attribute: bool is_active = false
original attribute: int desired_course = 0
original attribute: int desired_position_latitude = 0
original attribute: int desired_position_longitude = 0

Relations:
Relation:Controller.bowControl=Bow_Thrusters.
bowControl
Relation:Controller.azimuthControl=Azimuth_Thrusters.
azimuthControl
Relation:GNSS_Sensor.positionData=Controller.
positionData
Relation:Inertial_Measurement_Unit.driftData=Controller.
driftData
Relation:Propeller_Anemometer.windData=Controller.
windData
Relation:Control_Console.setPoint=Controller.setPoint
```

Answer

Chat

```
signal."
},
},
{
  "diagram": "diagram2",
  "description": "No association between GNSS_Sensor,
Inertial_Measurement_Unit, Propeller_Anemometer and User as actors, which
are present in diagram1."
},
{
  "diagram": "diagram2",
  "description": "Bow_Thrusters and Azimuth_Thrusters have signals but
these signals are not represented in the use case interactions in diagram1."
},
{
  "diagram": "diagram2",
  "description": "Controller and Control_Console blocks are missing in
diagram1, yet these blocks appear to handle major functionalities."
},
{
  "diagram": "diagram2",
  "description": "DPS block is not mentioned in diagram1, but is key to the
system as per the provided specifications."
}
]
...

```

Console

Done. Total time: 5894 ms

Close Start Apply response



# Outline

Introduction

Contribution

**In practice**

Takeaways

## In practice

### Test setup

- Three different systems: an automotive braking system, a space-based system, and ship dynamic positioning system
- All specifications, diagrams and results: Zenodo archive<sup>a</sup>
- Evaluation on model generation (UCD, BD), inconsistency detection, inconsistency correction
- Use GPT 4 as underlying LLM for inconsistency detection and correction

---

<sup>a</sup><https://doi.org/10.5281/zenodo.11962707>

## Results

System	Test	Diagram	Inconsistencies detected				Inconsistencies corrected		
			Internal	External	Errors	Total	Internal	External	Total
Automated braking	BD1 vs UCD1	<i>BD1</i>	1	2	0	3	1	2	3/3
		<i>UCD1</i>	0	0	0	0	0	0	—
	BD1 vs UCD2	<i>BD1</i>	0	1	0	1	0	1	1/1
		<i>UCD2</i>	0	3	0	3	0	2	2/3
	BD2 vs UCD1	<i>BD2</i>	5	1	1	6	4	1	5/6
		<i>UCD1</i>	0	1	1	1	0	1	1/1
	BD2 vs UCD2	<i>BD2</i>	4	2	0	6	3	1	4/6
		<i>UCD2</i>	2	2	0	4	2	2	4/4
Dynamic positioning system	BD1 vs UCD1	<i>BD1</i>	1	1	0	2	1	0	1/2
		<i>UCD1</i>	0	0	1	0	0	0	—
	BD1 vs UCD2	<i>BD1</i>	2	2	0	4	2	1.5	3.5/4
		<i>UCD2</i>	2	0	0	2	0	2	2/2
	BD2 vs UCD1	<i>BD2</i>	1	0	0	1	1	0	1/1
		<i>UCD1</i>	1	1	0	2	1	1	2/2
	BD2 vs UCD2	<i>BD2</i>	3	0	1	3	3	0	3/3
		<i>UCD2</i>	0	0	1	0	0	0	—
//									
Total			36	33	6	69	30	30.5	60.5/69





# Outline

Introduction

Contribution

In practice

Takeaways

# Takeaways

## Pros

- Blending is cool! Significant model improvements derived from the integration of the rule-based and LLM-based approaches
- High automated correction rate for LLM-detected inconsistencies: between 50% and 100% per diagram, averaging at 87%

## Limits and improvement directions

- With cross-diagram consistency rules, the LLM tends to exclusively focus on these rules → the user can opt to use them or not
- Correction rate is not at 100% for LLM-detected inconsistencies
- Extension to other diagrams, to other tools (cross-tool consistency ?)
- Possibility to support user-defined rules

## Questions?



Latest version of TTool includes what has been presented in the paper... and much more! [ttool.telecom-paris.fr](http://ttool.telecom-paris.fr)

*All evaluation results (including models) are available on our public Zenodo repository*