AI Suggestions for Continuous MDE

Ludovic Apvrille, Bastien Sultan
{ludovic.apvrille, bastien.sultan}@telecom-paris.fr

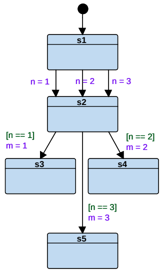
MODELSWARD '26 – Marbella

Outline

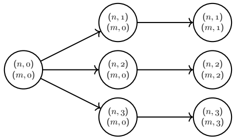
A few preliminary words

Our (first) scope: **formal** MDE

- Mathematically-grounded modeling languages
- Expression of requirements in temporal logic properties
- Formal verification (including model checking)
- Simulation

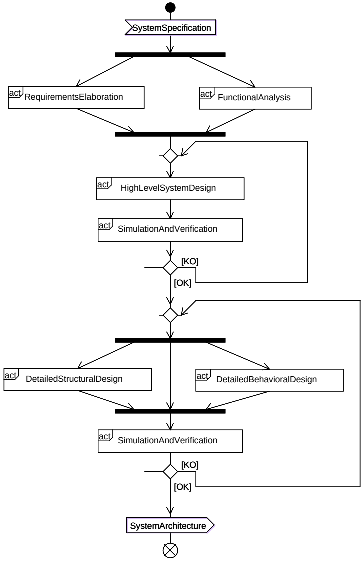


$\langle s_0, S, T\rangle \mapsto \langle V, E, A, val\rangle$
– $E \subseteq V \times V$
– $val : V \to \{(a, x) | a \in A, x \in type(a)\}$
– $type : A \to \{\mathbb{B}ool, \mathbb{Z}\}$



T $E <> n = m$
F $A[]\ n = m$
T $A <> n = m$

A few preliminary words


SystemSpecification
act RequirementsElaboration
act FunctionalAnalysis
act HighLevelSystemDesign
act SimulationAndVerification
[KO]
[OK]
act DetailedStructuralDesign
act DetailedBehavioralDesign
act SimulationAndVerification
[KO]
[OK]
SystemArchitecture

Our (second) scope: incremental MDE

- MDE as a dynamic, iterative process
- Model mutations
- Incremental model checking

AI-assisted MDE

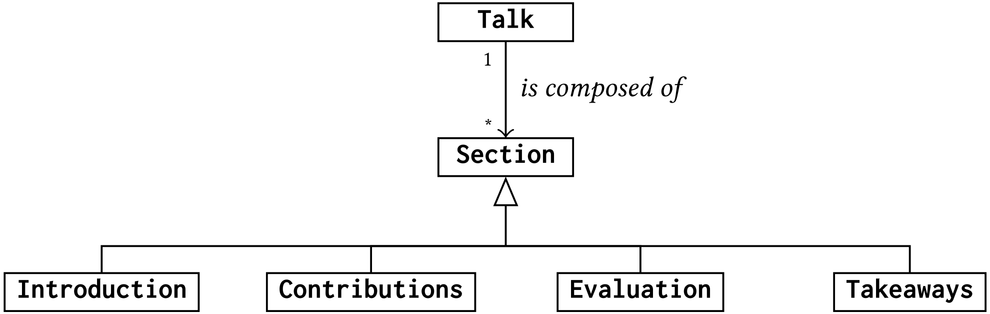
- Drafting models from specification, generating mutation scripts, deriving CTL formulas from requirements, ...
- Yet, **triggered by the user**

A few preliminary words

ContinuousAI

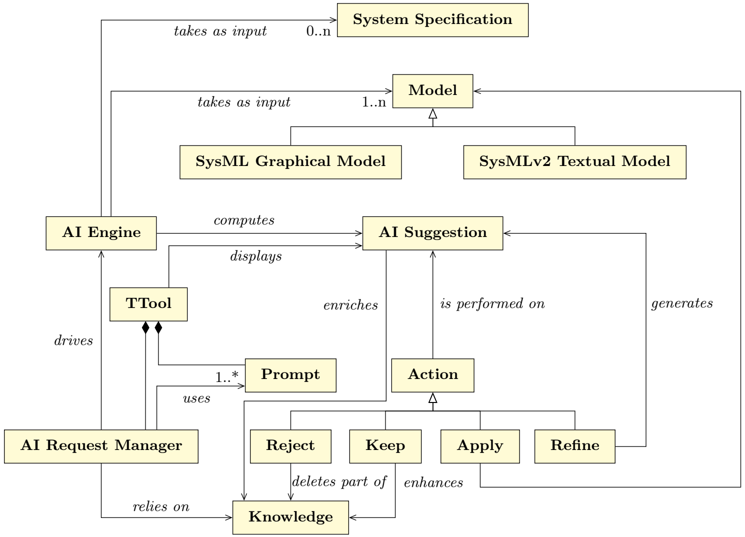
- Proactive user assistance
- Modeling suggestions
- Triggered either by the user, by a model change, or continuously as a background task
- Implementation compatible with local LLMs to reduce both cost and environmental footprint

Outline

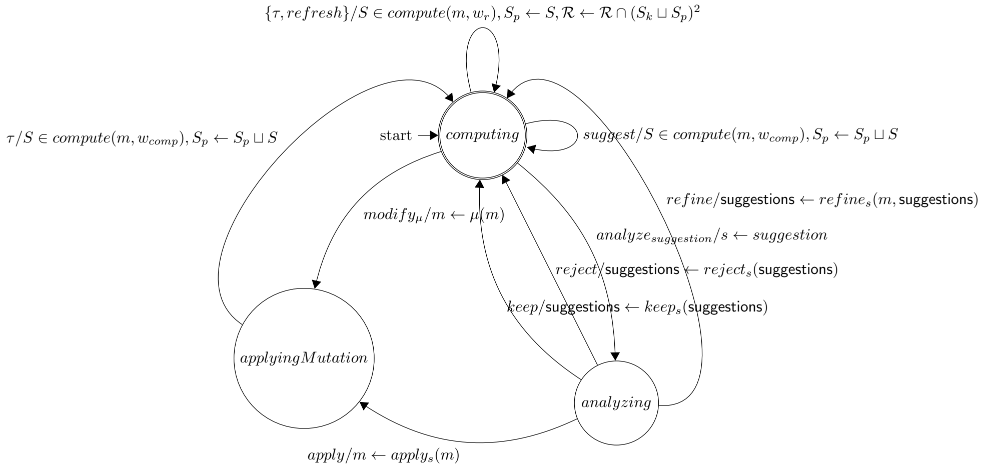

Talk
1
is composed of
*
Section
Introduction
Contributions
Evaluation
Takeaways

Outline

ContinuousAI: core concepts



ContinuousAI: dynamic behavior


$\{\tau, refresh\}/S \in compute(m, w_r), S_p \leftarrow S, \mathcal{R} \leftarrow \mathcal{R} \cap (S_k \sqcup S_p)^2$
$\tau/S \in compute(m, w_{comp}), S_p \leftarrow S_p \sqcup S$
start
$computing$
$suggest/S \in compute(m, w_{comp}), S_p \leftarrow S_p \sqcup S$
$refine$/suggestions $\leftarrow refine_s(m,$ suggestions$)$
$modify_\mu/m \leftarrow \mu(m)$
$analyze_{suggestion}/s \leftarrow suggestion$
$reject$/suggestions $\leftarrow reject_s($suggestions$)$
$keep$/suggestions $\leftarrow keep_s($suggestions$)$
$applyingMutation$
$analyzing$
$apply/m \leftarrow apply_s(m)$

ContinuousAI: implementation

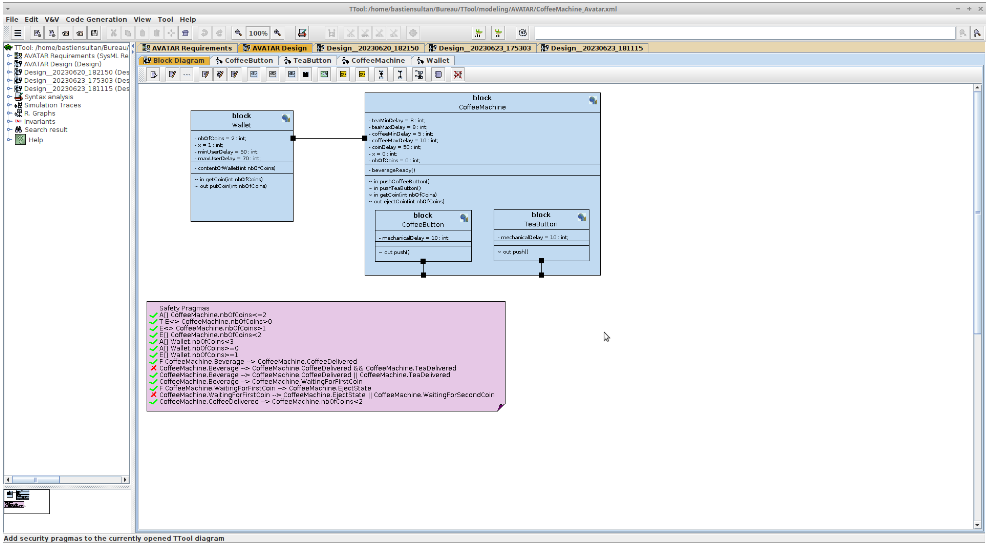
Platform

- Open-source MDE toolkit TTool
- Interfaced with local LLMs (gpt-oss-120b, qwen3-coder-30b, devstral-small-2-2512, etc., through LMStudio) or remote LLMs
- GUI or CLI ($\leftarrow$ adaptability with other MDE environments)

Supported languages and diagrams

- UML/SysML v1/SysML v2
- Use cases, state machines, requirement diagrams, sequence diagrams, block diagrams, attack trees

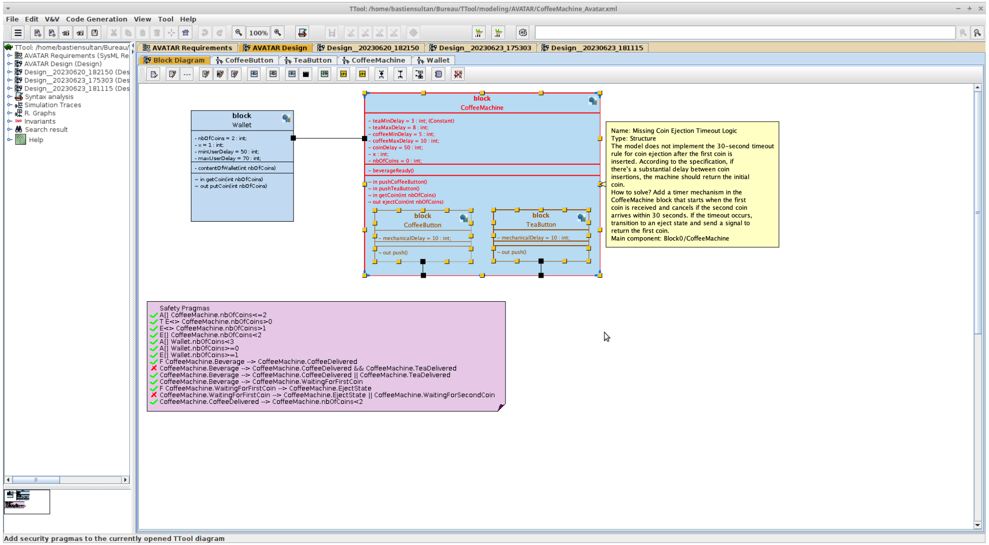
ContinuousAI: implementation



```
Safety Pragmas
✓ A[] CoffeeMachine.nbOfCoins<=2
✓ T E<> CoffeeMachine.nbOfCoins>0
✓ E<> CoffeeMachine.nbOfCoins>1
✓ E[] CoffeeMachine.nbOfCoins<2
✓ A[] Wallet.nbOfCoins<3
✓ A[] Wallet.nbOfCoins>=0
✓ E[] Wallet.nbOfCoins>=1
✓ F CoffeeMachine.Beverage --> CoffeeMachine.CoffeeDelivered
✗ CoffeeMachine.Beverage --> CoffeeMachine.CoffeeDelivered && CoffeeMachine.TeaDelivered
✓ CoffeeMachine.Beverage --> CoffeeMachine.CoffeeDelivered || CoffeeMachine.TeaDelivered
✓ CoffeeMachine.Beverage --> CoffeeMachine.WaitingForFirstCoin
✓ F CoffeeMachine.WaitingForFirstCoin --> CoffeeMachine.EjectState
✗ CoffeeMachine.WaitingForFirstCoin --> CoffeeMachine.EjectState || CoffeeMachine.WaitingForSecondCoin
✓ CoffeeMachine.CoffeeDelivered --> CoffeeMachine.nbOfCoins<2
```

Add security pragmas to the currently opened TTool diagram

ContinuousAI: implementation


TTool: /home/bastiensultan/Bureau/TTool/modeling/AVATAR/CoffeeMachine_Avatar.xml
File Edit V&V Code Generation View Tool Help
100%
TTool: /home/bastiensultan/Bureau/
AVATAR Requirements (SysML Re
AVATAR Design (Design)
Design__20230620_182150 (Des
Design__20230623_175303 (Des
Design__20230623_181115 (Des
Syntax analysis
Simulation Traces
R. Graphs
Invariants
Search result
Help
AVATAR Requirements
AVATAR Design
Design__20230620_182150
Design__20230623_175303
Design__20230623_181115
Block Diagram
CoffeeButton
TeaButton
CoffeeMachine
Wallet

block
Wallet

- nbOfCoins = 2 : int;
- x = 1 : int;
- minUserDelay = 50 : int;
- maxUserDelay = 70 : int;

- contentOfWallet(int nbOfCoins)

~ in getCoin(int nbOfCoins)
~ out putCoin(int nbOfCoins)

block
CoffeeMachine

- teaMinDelay = 3 : int; (Constant)
- teaMaxDelay = 8 : int;
- coffeeMinDelay = 5 : int;
- coffeeMaxDelay = 10 : int;
- coinDelay = 50 : int;
- x : int;
- nbOfCoins = 0 : int;

- beverageReady()

~ in pushCoffeeButton()
~ in pushTeaButton()
~ in getCoin(int nbOfCoins)
~ out ejectCoin(int nbOfCoins)

block
CoffeeButton

- mechanicalDelay = 10 : int;

~ out push()

block
TeaButton

- mechanicalDelay = 10 : int;

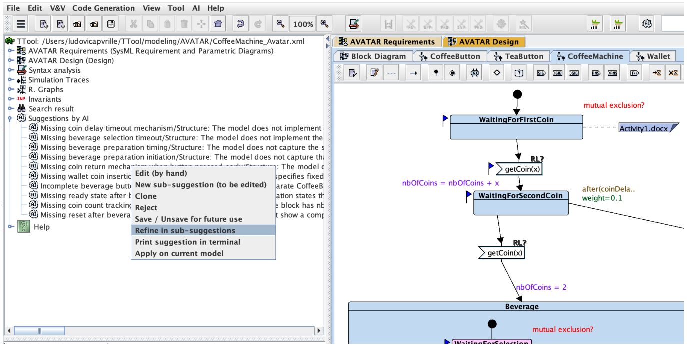
~ out push()

Name: Missing Coin Ejection Timeout Logic
Type: Structure
The model does not implement the 30-second timeout rule for coin ejection after the first coin is inserted. According to the specification, if there's a substantial delay between coin insertions, the machine should return the initial coin.
How to solve? Add a timer mechanism in the CoffeeMachine block that starts when the first coin is received and cancels if the second coin arrives within 30 seconds. If the timeout occurs, transition to an eject state and send a signal to return the first coin.
Main component: Block0/CoffeeMachine

Safety Pragmas
✓ A[] CoffeeMachine.nbOfCoins<=2
✓ T E<> CoffeeMachine.nbOfCoins>0
✓ E<> CoffeeMachine.nbOfCoins>1
✓ E[] CoffeeMachine.nbOfCoins<2
✓ A[] Wallet.nbOfCoins<3
✓ A[] Wallet.nbOfCoins>=0
✓ E[] Wallet.nbOfCoins>=1
✓ F CoffeeMachine.Beverage --> CoffeeMachine.CoffeeDelivered
✗ CoffeeMachine.Beverage --> CoffeeMachine.CoffeeDelivered && CoffeeMachine.TeaDelivered
✓ CoffeeMachine.Beverage --> CoffeeMachine.CoffeeDelivered || CoffeeMachine.TeaDelivered
✓ CoffeeMachine.Beverage --> CoffeeMachine.WaitingForFirstCoin
✓ F CoffeeMachine.WaitingForFirstCoin --> CoffeeMachine.EjectState
✗ CoffeeMachine.WaitingForFirstCoin --> CoffeeMachine.EjectState || CoffeeMachine.WaitingForSecondCoin
✓ CoffeeMachine.CoffeeDelivered --> CoffeeMachine.nbOfCoins<2

Add security pragmas to the currently opened TTool diagram

ContinuousAI: implementation



TTool: /Users/ludovicapvrille/TTool/modeling/AVATAR/CoffeeMachine_Avatar.xml
- AVATAR Requirements (SysML Requirement and Parametric Diagrams)
- AVATAR Design (Design)
- Syntax analysis
- Simulation Traces
- R. Graphs
- Invariants
- Search result
- Suggestions by AI
  - Missing coin delay timeout mechanism/Structure: The model does not implement
  - Missing beverage selection timeout/Structure: The model does not implement the
  - Missing beverage preparation timing/Structure: The model does not capture the s
  - Missing beverage preparation initiation/Structure: The model does not capture th
  - Missing coin return mecha … : The model c
  - Missing wallet coin insertic … specifies fixed
  - Incomplete beverage butt … arate CoffeeB
  - Missing ready state after b … ation states th
  - Missing coin count tracking … e block has nb
  - Missing reset after bevera … t show a comp
- Help

Outline

Evaluation

Evaluation setup

- Four kinds of diagrams: RD, UCD, BD, SMD
- Three systems: fault-tolerant satellite communication system, ICS of a packaging chain, ICS of a smart greenhouse
  - Diagrams produced by final-year MSc in engineering students
- Both component-level and overall suggestions
- Two local LLMs: gpt-oss-120b and qwen3-coder-30b

Metrics

- Relevance of the suggestion
- Ability to improve the model
- Computation time
- Carbon footprint

Evaluation

		Before applying suggestions				After applying suggestions	
		Quality of suggestions: score /10				Model quality evolution (-3 to +3)	
		GPT-OSS-20b		Qwen3			
	Model grade /10	Score	Time (in ms)	Score	Time (in ms)	Score OSS	Score Qwen
RD	9.4	7.7	9,337	8.7	17,241	+1.7	+2.5
UCD	8.9	9	6,544	9	26,088	+2.7	+2.7
BD	9.2	8.8	8,070	8.7	15,644	+2.7	+2.7
	Average	8.3	7,940	8.8	21,664	Average +2.2	+2.6

Suggestions bearing on the whole diagram

		Before applying suggestions				After applying suggestions	
		Quality of suggestions: score /10				Model quality evolution (-3 to +3)	
		GPT-OSS-20b		Qwen3			
	Model grade /10	Score	Time (in ms)	Score	Time (in ms)	Score OSS	Score Qwen
RD	9.4	8.7	1,935	5.7	9,564	+2.7	+1.7
UCD	8.9	6.3	1,995	10	8,878	+1.7	+3
BD	9.2	9.2	2,571	10	11,885	+2.7	+3
	Average	7.5	1,965	7.8	9,221	Average +2.2	+2.3

Suggestions bearing on a given component

Evaluation

Key points

- In almost all cases, at least the half of the proposed suggestions were judged relevant
- Ability to improve model:
  - Applying ContinuousAI suggestions would have kept or improved model quality in 98.6% of the cases
  - Strictly improved it in 88.9%
- Computation time:
  - Overall model: ~ 9 s with gpt-oss to ~ 26.4 s with qwen3-coder
  - Single component: ~ 3.8 s with gpt-oss to ~ 7.7 s with qwen3-coder

Evaluation

	GPT-OSS-20b	Qwen3
Average power consumption per request (J)	966	1529
Grams of CO2 per request	0.0135	0.0214
Average power consumption per suggestion (J)	293	463
Grams of CO2 per suggestion	0.0041	0.0065

Some thoughts

- Average Google query: 0.2g CO^2
- Yet: those figures depend on the energy mix of the country
  - In France we rely widely on nuclear plants, so on average kWh-to-CO^2 is low

Outline

Takeaways

ContinuousAI

- Continuous AI assistance for MDE
- Several modes: model change, user's request, *continuous*
- Several analysis scopes: overall model vs selected component
- Compatible with self-hosted LLMs
- Free and open source!

Takeaways

Pros

- Encouraging results in terms of relevance/ability to improve model quality
- Reasonable computation times
- User in the loop

Limits and improvement directions

- Suggestions are textual: the user needs to apply them manually
    - Future works: return a list of (formal) mutation operators ready to apply
- Some suggestions are not relevant
    - Future works: algorithmic analysis of the mutations
- Opens the way to integration of ContinuousAI with TTool's incremental model checker

Questions?

Latest version of TTool includes what has been presented in the paper. . . and much more! `https://ttool.telecom-paris.fr/ttoolai`

Detailed evaluation results and detailed guidance on how to use ContinuousAI are available on the paper's GitHub repository: `https://github.com/ZebreDeSoixanteQuatorzeCanons/ContinuousAI`