Verification Using Contracts

Bentley James Oakes

University of Antwerp/Flanders Make

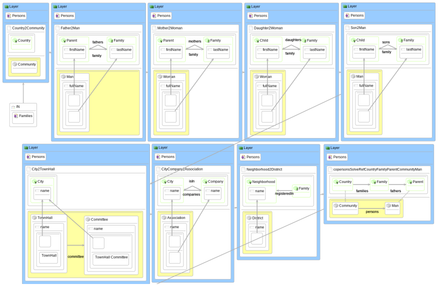
October 22, 2018

Overview

Outline

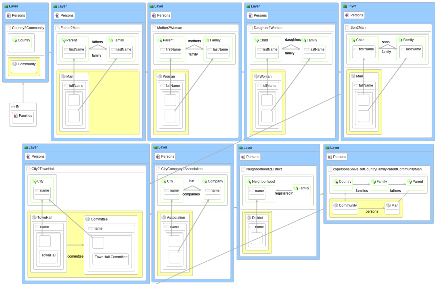
Motivation

Motivation



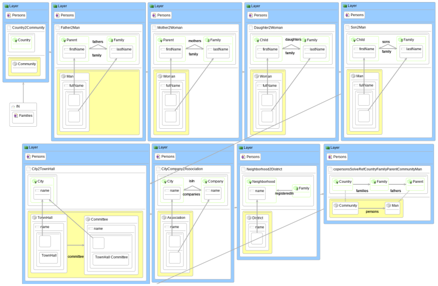
Layer
Persons
Country2Community
Country
Community
IN
Families
Father2Man
Parent
firstName
fathers
family
Family
lastName
Man
fullName
Mother2Woman
mothers
Woman
Daughter2Woman
Child
daughters
Son2Man
sons
City2TownHall
City
name
TownHall
Committee
committee
TownHall Committee
CityCompany2Association
isIn
companies
Company
Association
Neighborhood2District
Neighborhood
registeredIn
District
copersonsSolveRefCountryFamilyParentCommunityMan
families
persons

Motivation



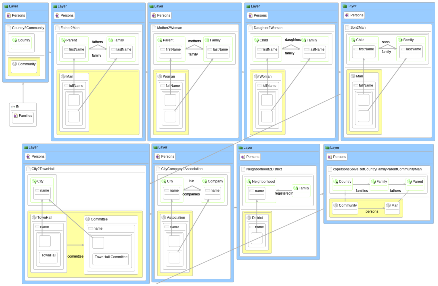
- GIVEN: A transformation divided into layers, containing LHS/RHS rules

Motivation



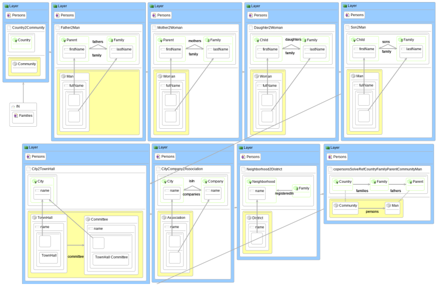
- GIVEN: A transformation divided into layers, containing LHS/RHS rules
- GOAL/WHY: Understand transformation's behaviour

Motivation



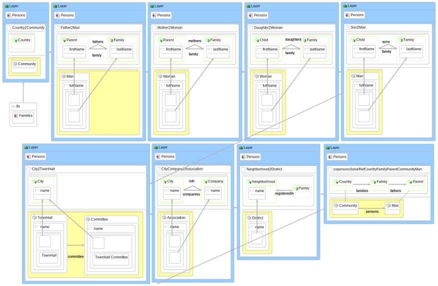
- GIVEN: A transformation divided into layers, containing LHS/RHS rules
- GOAL/WHY: Understand transformation's behaviour
    - Relation between input/output elements

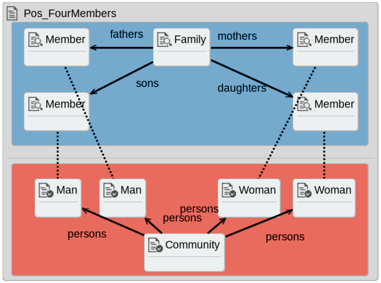
Motivation



- GIVEN: A transformation divided into layers, containing LHS/RHS rules
- GOAL/WHY: Understand transformation's behaviour
  - Relation between input/output elements

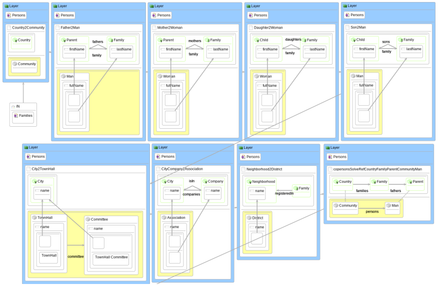
Motivation

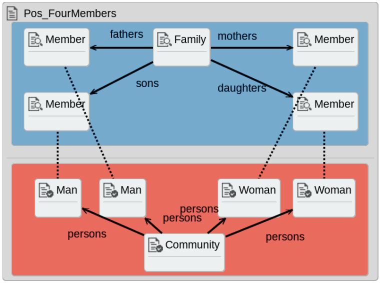




- GIVEN: A transformation divided into layers, containing LHS/RHS rules
- GOAL/WHY: Understand transformation's behaviour
  - Relation between input/output elements

Motivation

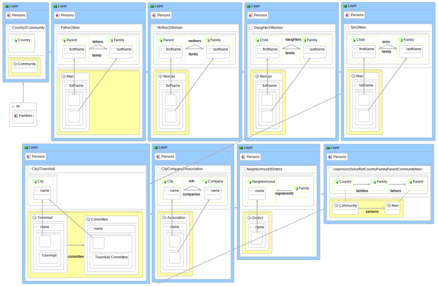


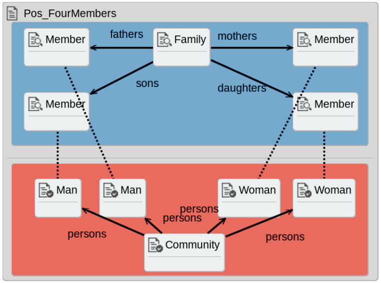


- GIVEN: A transformation divided into layers, containing LHS/RHS rules
- GOAL/WHY: Understand transformation's behaviour
  - Relation between input/output elements

- WHAT: Prove structural contracts to guarantee element existence

Motivation

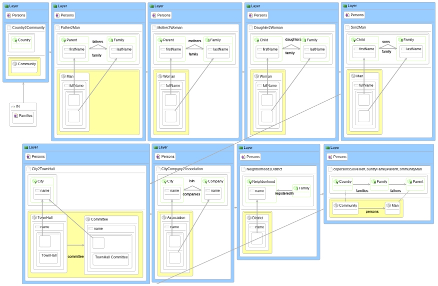


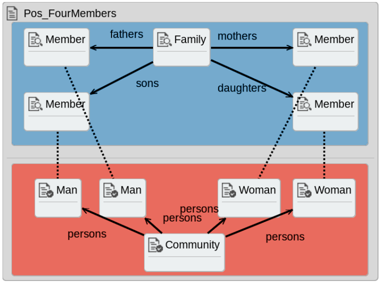


- GIVEN: A transformation divided into layers, containing LHS/RHS rules
- GOAL/WHY: Understand transformation's behaviour
  - Relation between input/output elements

- WHAT: Prove structural contracts to guarantee element existence
- HOW: Create all possible rule combinations through symbolic execution

Motivation

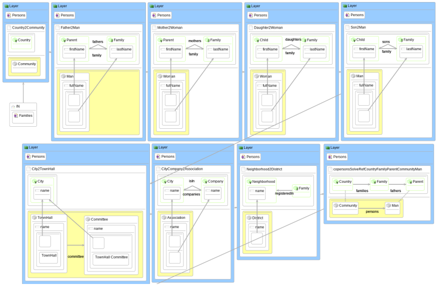


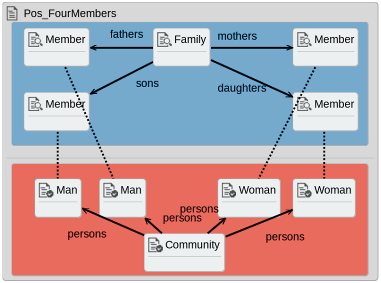


- GIVEN: A transformation divided into layers, containing LHS/RHS rules
- GOAL/WHY: Understand transformation's behaviour
  - Relation between input/output elements

- WHAT: Prove structural contracts to guarantee element existence
- HOW: Create all possible rule combinations through symbolic execution

Motivation





- GIVEN: A transformation divided into layers, containing LHS/RHS rules
- GOAL/WHY: Understand transformation's behaviour
    - Relation between input/output elements

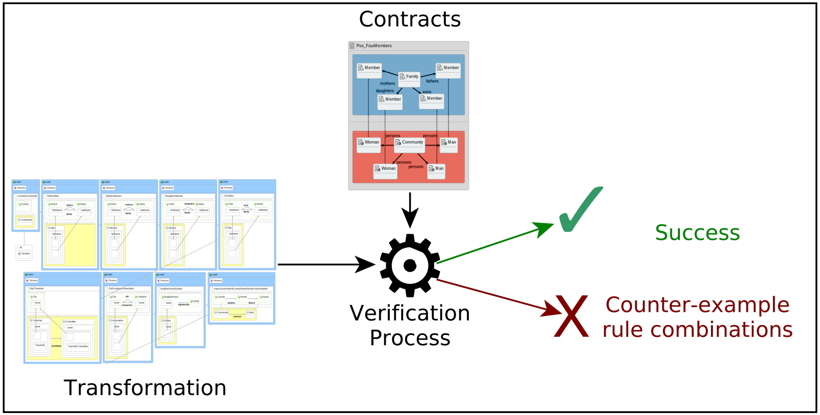
- WHAT: Prove structural contracts to guarantee element existence
- HOW: Create all possible rule combinations through symbolic execution

Bentley Oakes. 2018. *A Symbolic Execution-Based Approach to Model Transformation Verification Using Structural Contracts*. Ph.D. Dissertation. McGill University.

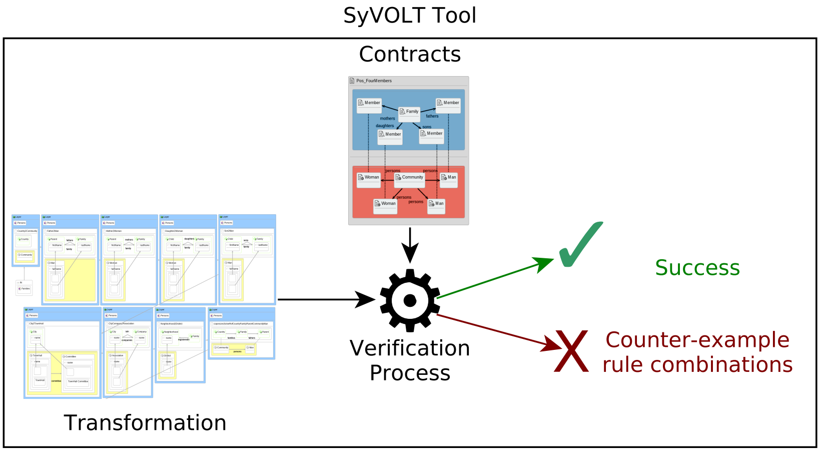
SyVOLT

SyVOLT

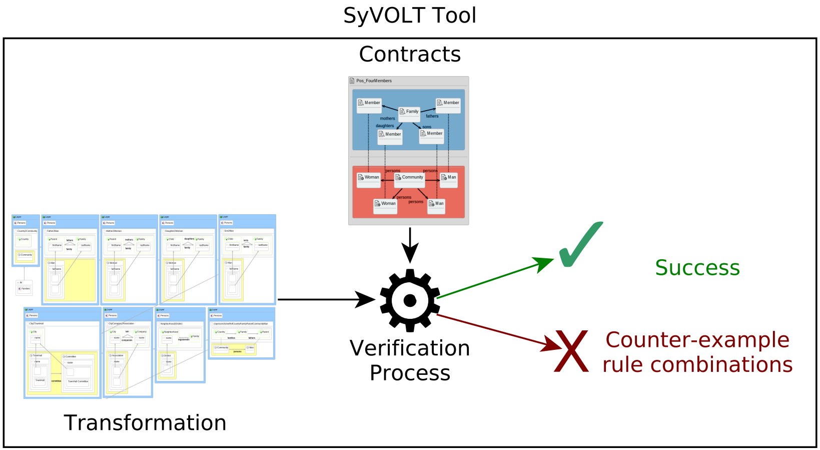
SyVOLT Tool



SyVOLT



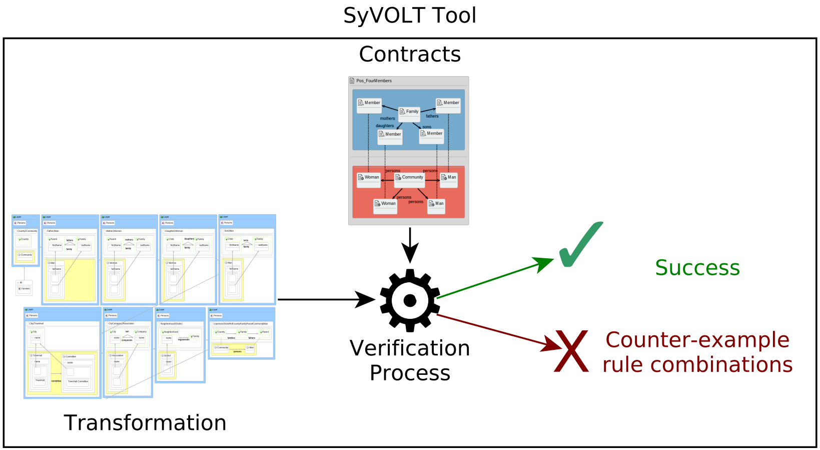
Features:



Features:

- Eclipse/MPS visual editors

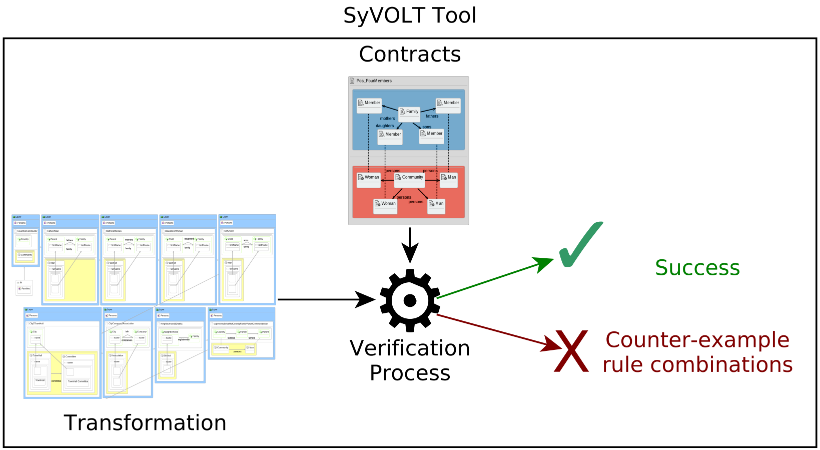
SyVOLT



Features:

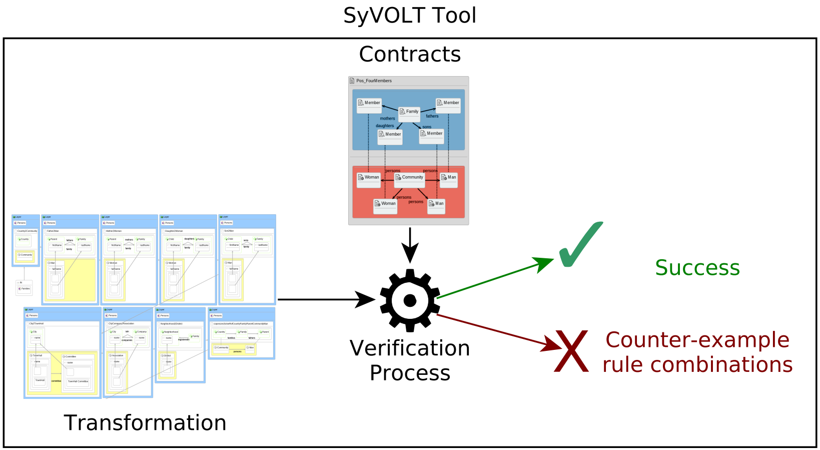
- Eclipse/MPS visual editors
- HOT from ATL

SyVOLT



Features:

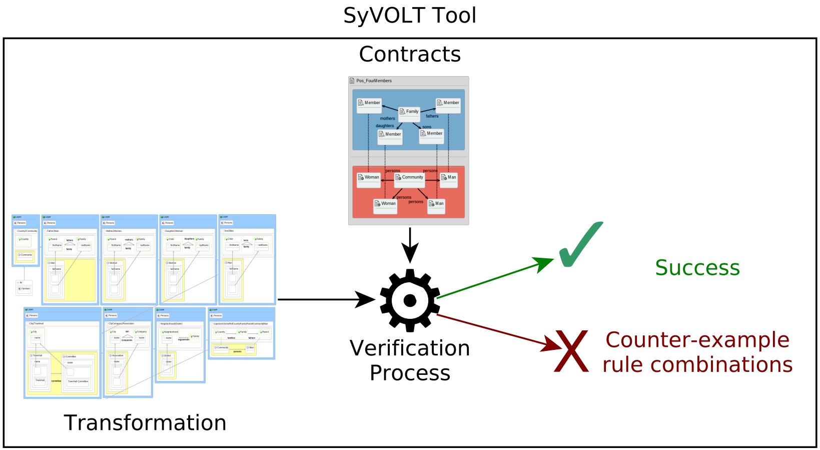
- Eclipse/MPS visual editors
- HOT from ATL
- Verif. possible in seconds



Features:

- Eclipse/MPS visual editors
- HOT from ATL
- Verif. possible in seconds

SyVOLT

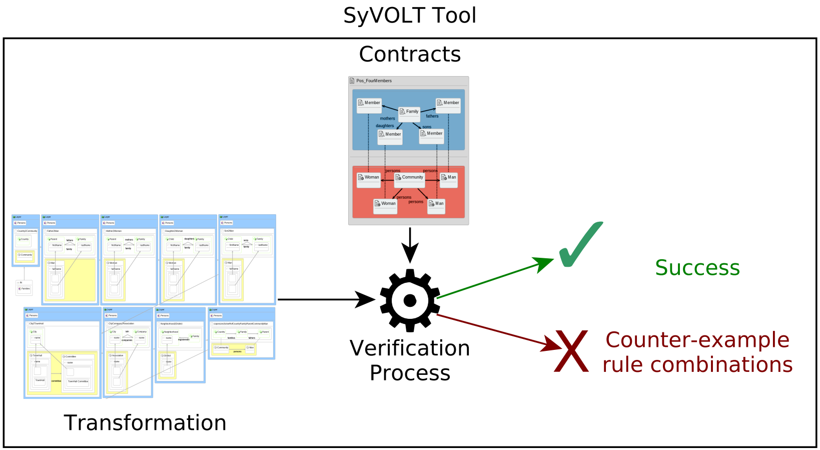


Features:

- Eclipse/MPS visual editors
- HOT from ATL
- Verif. possible in seconds

Limitations:

SyVOLT



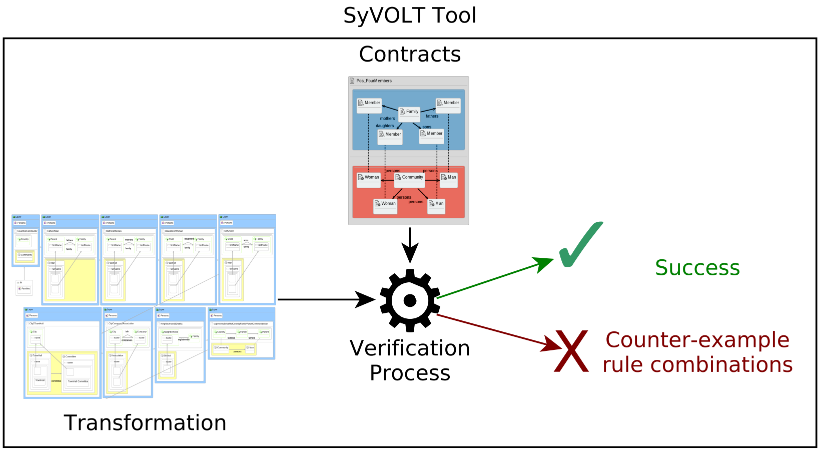
Features:

- Eclipse/MPS visual editors
- HOT from ATL
- Verif. possible in seconds

Limitations:

- Reduced expressiveness

SyVOLT



Features:

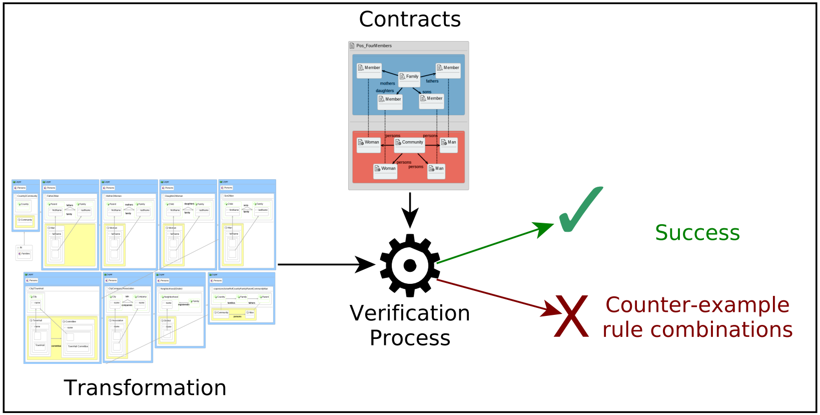
- Eclipse/MPS visual editors
- HOT from ATL
- Verif. possible in seconds

Limitations:

- Reduced expressiveness
- Structural contracts only

SyVOLT

SyVOLT Tool



Features:

- Eclipse/MPS visual editors
- HOT from ATL
- Verif. possible in seconds

Limitations:

- Reduced expressiveness
- Structural contracts only
- Limited contract language

Future Work

Future Work

- Extend to other model transformation languages

Future Work

- Extend to other model transformation languages
- Promote "contract-based design" of model transformations, with continuous verification

Future Work

- Extend to other model transformation languages
- Promote "contract-based design" of model transformations, with continuous verification
- Tooling: Integrate transformation verification into the ModelVerse

Outline

Motivation

Motivation

- CONTEXT: Development of a safety-critical system - car, airplane, smart home, etc.

Motivation

- CONTEXT: Development of a safety-critical system - car, airplane, smart home, etc.
- GOAL/WHY: Ensure traceability of requirements/safety goals/test cases

Motivation

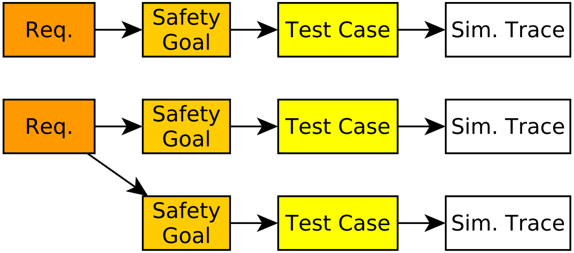
- CONTEXT: Development of a safety-critical system - car, airplane, smart home, etc.
- GOAL/WHY: Ensure traceability of requirements/safety goals/test cases
  - Part of certification - ISO 26262

Motivation

- CONTEXT: Development of a safety-critical system - car, airplane, smart home, etc.
- GOAL/WHY: Ensure traceability of requirements/safety goals/test cases
    - Part of certification - ISO 26262

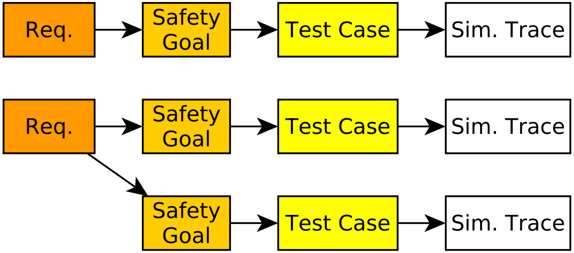
Motivation

- CONTEXT: Development of a safety-critical system - car, airplane, smart home, etc.
- GOAL/WHY: Ensure traceability of requirements/safety goals/test cases
    - Part of certification - ISO 26262



Motivation

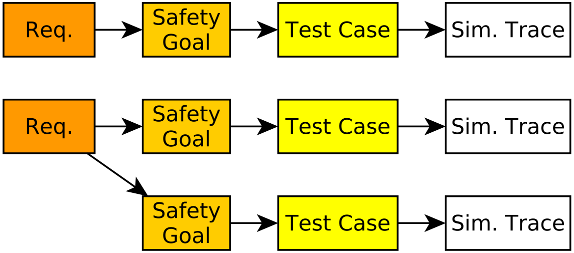
- CONTEXT: Development of a safety-critical system - car, airplane, smart home, etc.
- GOAL/WHY: Ensure traceability of requirements/safety goals/test cases
  - Part of certification - ISO 26262



- EXAMPLE: “A requirement changed, are the related safety goals still valid?”

Motivation

- CONTEXT: Development of a safety-critical system - car, airplane, smart home, etc.
- GOAL/WHY: Ensure traceability of requirements/safety goals/test cases
  - Part of certification - ISO 26262



- EXAMPLE: “A requirement changed, are the related safety goals still valid?”
- EXAMPLE: “Is every safety goal connected to a (consistent) requirement?”

Req. Mgmt.

Req. Mgmt.

- EXAMPLE: "A requirement changed, are the related safety goals still valid?"
- EXAMPLE: "Is every safety goal connected to a (consistent) requirement?"

Req. Mgmt.

- EXAMPLE: "A requirement changed, are the related safety goals still valid?"
- EXAMPLE: "Is every safety goal connected to a (consistent) requirement?"

WHAT: Techniques/DSL/tool for expressing/enforcing traceability

Req. Mgmt.

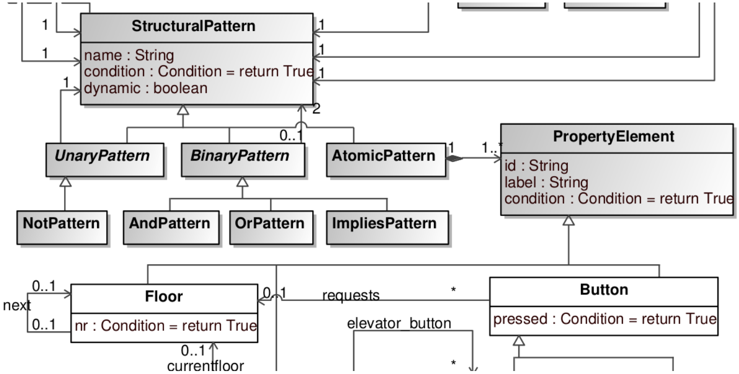
- EXAMPLE: “A requirement changed, are the related safety goals still valid?”
- EXAMPLE: “Is every safety goal connected to a (consistent) requirement?”

WHAT: Techniques/DSL/tool for expressing/enforcing traceability
HOW: ProMoBox-like approach for defining contracts?

Req. Mgmt.

- EXAMPLE: "A requirement changed, are the related safety goals still valid?"
- EXAMPLE: "Is every safety goal connected to a (consistent) requirement?"

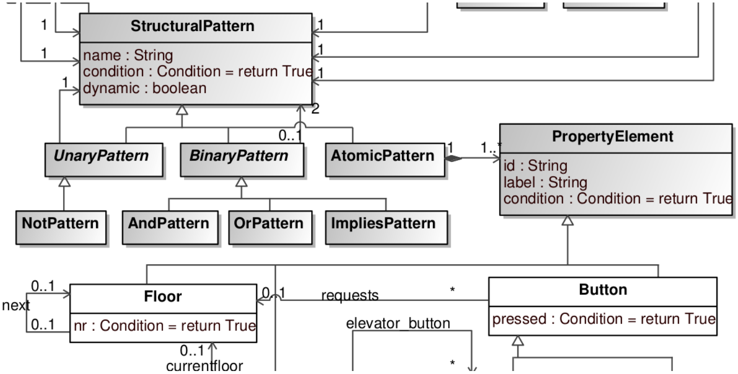
WHAT: Techniques/DSL/tool for expressing/enforcing traceability
HOW: ProMoBox-like approach for defining contracts?



Req. Mgmt.

- EXAMPLE: “A requirement changed, are the related safety goals still valid?”
- EXAMPLE: “Is every safety goal connected to a (consistent) requirement?”

WHAT: Techniques/DSL/tool for expressing/enforcing traceability
HOW: ProMoBox-like approach for defining contracts?

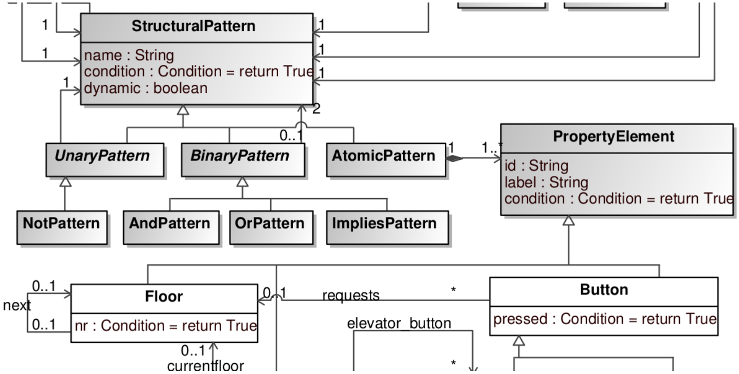


- OCL for structural constraints?

Req. Mgmt.

- EXAMPLE: "A requirement changed, are the related safety goals still valid?"
- EXAMPLE: "Is every safety goal connected to a (consistent) requirement?"

WHAT: Techniques/DSL/tool for expressing/enforcing traceability
HOW: ProMoBox-like approach for defining contracts?

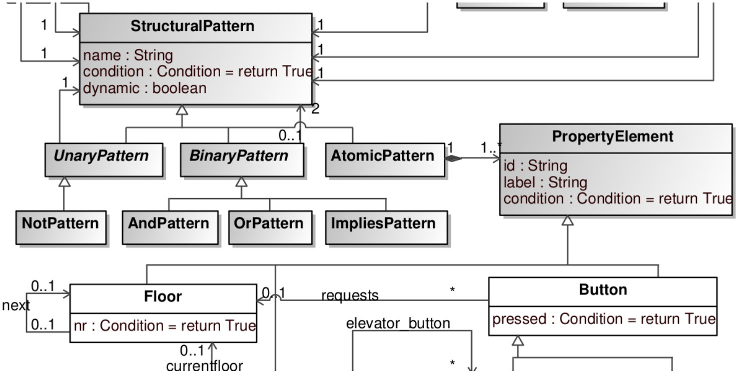


- OCL for structural constraints?
- DSL with semantics for editing model state?

Req. Mgmt.

- EXAMPLE: "A requirement changed, are the related safety goals still valid?"
- EXAMPLE: "Is every safety goal connected to a (consistent) requirement?"

WHAT: Techniques/DSL/tool for expressing/enforcing traceability
HOW: ProMoBox-like approach for defining contracts?



- OCL for structural constraints?
- DSL with semantics for editing model state?
  - "When req. is edited, mark connected safety goals as needing manual check"

Outline

Motivation

Motivation

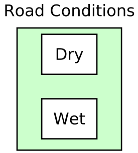
CONTEXT: Development of safety-critical system - car, airplane, smart home, etc.

Motivation

CONTEXT: Development of safety-critical system - car, airplane, smart home, etc.
GOAL/WHY: Automate assigning of severity levels to scenarios

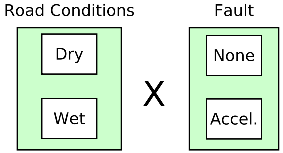
Motivation

CONTEXT: Development of safety-critical system - car, airplane, smart home, etc.
GOAL/WHY: Automate assigning of severity levels to scenarios



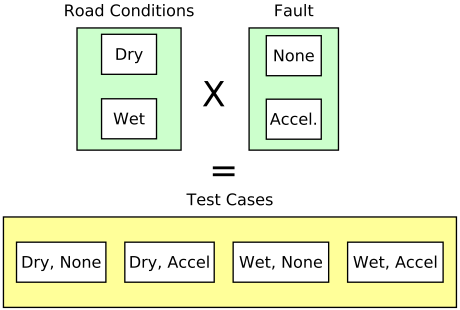
Motivation

CONTEXT: Development of safety-critical system - car, airplane, smart home, etc.
GOAL/WHY: Automate assigning of severity levels to scenarios



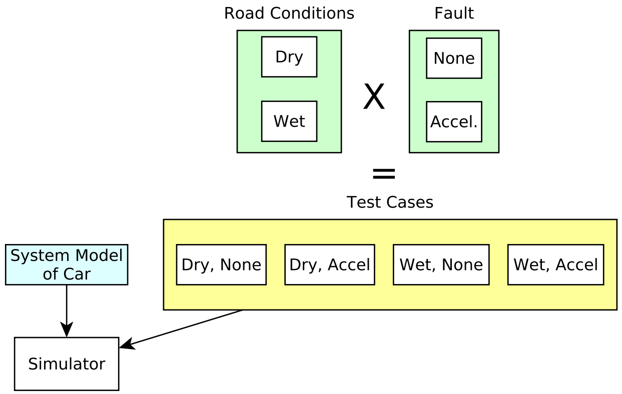
Motivation

CONTEXT: Development of safety-critical system - car, airplane, smart home, etc.
GOAL/WHY: Automate assigning of severity levels to scenarios



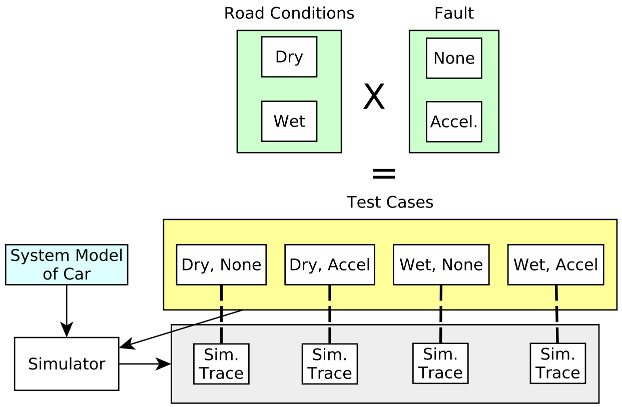
Motivation

CONTEXT: Development of safety-critical system - car, airplane, smart home, etc.
GOAL/WHY: Automate assigning of severity levels to scenarios



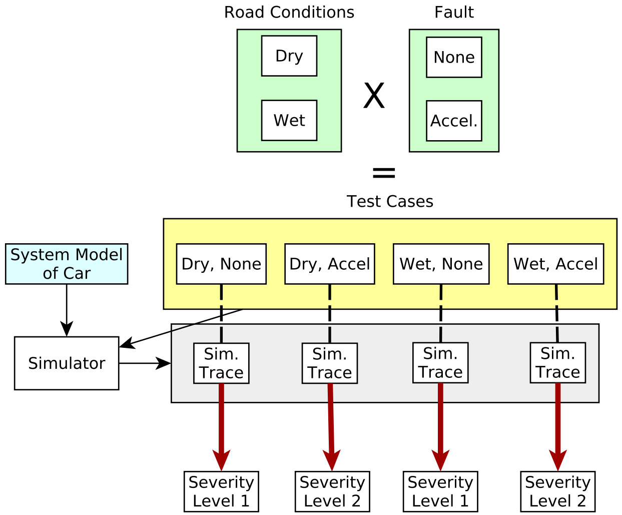
Motivation

CONTEXT: Development of safety-critical system - car, airplane, smart home, etc.
GOAL/WHY: Automate assigning of severity levels to scenarios



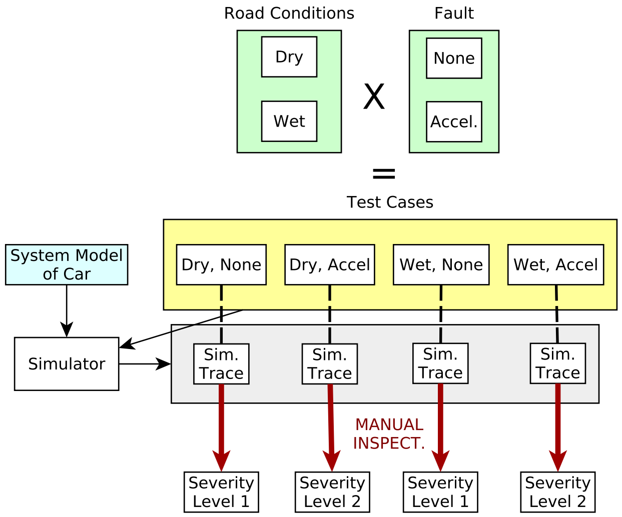
Motivation

CONTEXT: Development of safety-critical system - car, airplane, smart home, etc.
GOAL/WHY: Automate assigning of severity levels to scenarios



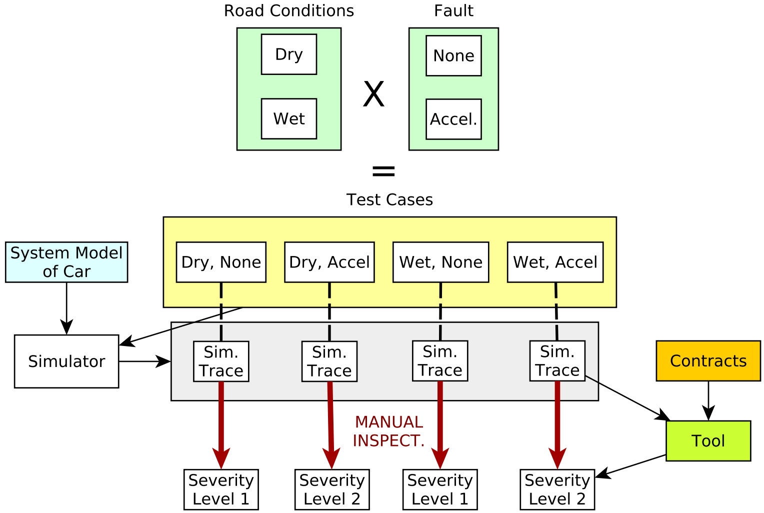
Motivation

CONTEXT: Development of safety-critical system - car, airplane, smart home, etc.
GOAL/WHY: Automate assigning of severity levels to scenarios



Motivation

CONTEXT: Development of safety-critical system - car, airplane, smart home, etc.
GOAL/WHY: Automate assigning of severity levels to scenarios



Process

Process

Natural Language Contract:

Process

Natural Language Contract:

IF accel $>=$ 3 G && accel $<=$ 4 G for duration $<$ 5 sec THEN severity $=$ S2

Process

Natural Language Contract:
IF accel $>=$ 3 G && accel $<=$ 4 G for duration $<$ 5 sec THEN severity $=$ S2

$\rightarrow$

Process

Natural Language Contract:
IF accel >= 3 G && accel <= 4 G for duration < 5 sec THEN severity = S2

$$\rightarrow$$

Signal Temporal Logic (STL):

Process

Natural Language Contract:
IF accel $>=$ 3 G && accel $<=$ 4 G for duration $<$ 5 sec THEN severity $=$ S2

$\rightarrow$

Signal Temporal Logic (STL):
$\diamond_{[0,5)}(accel >= 3 \&\& accel <= 4)$

Process

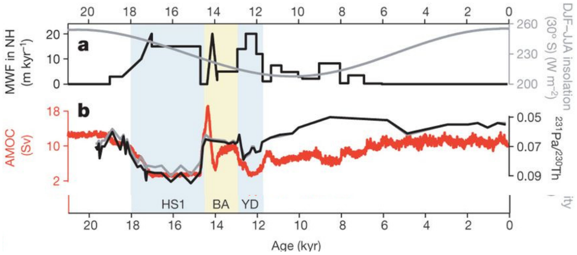
Natural Language Contract:

IF accel >= 3 G && accel <= 4 G for duration < 5 sec THEN severity = S2

$$\rightarrow$$

Signal Temporal Logic (STL):

$$\Diamond_{[0,5)}(accel >= 3 \ \&\& \ accel <= 4)$$

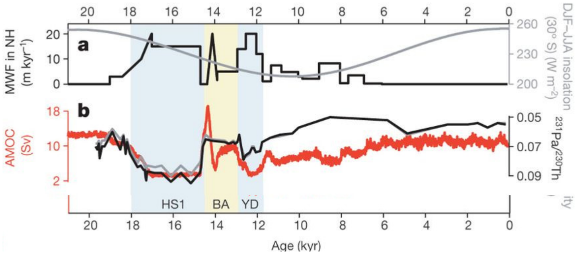


Process

Natural Language Contract:
IF accel >= 3 G && accel <= 4 G for duration < 5 sec THEN severity = S2

$$\rightarrow$$

Signal Temporal Logic (STL):
$\Diamond_{[0,5)}(\mathit{accel} >= 3 \ \&\& \ \mathit{accel} <= 4)$



Tool for checking:

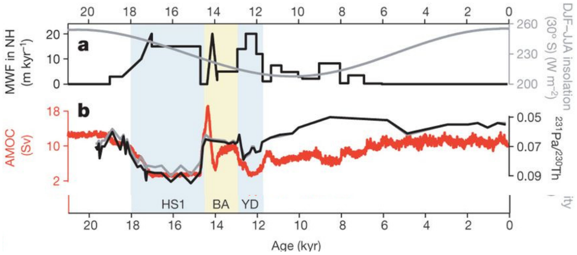
Process

Natural Language Contract:
IF accel >= 3 G && accel <= 4 G for duration < 5 sec THEN severity = S2

$$\rightarrow$$

Signal Temporal Logic (STL):

$$\diamond_{[0,5)}(\mathit{accel} >= 3 \ \&\& \ \mathit{accel} <= 4)$$



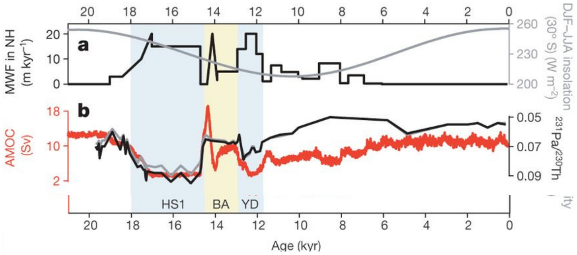
Tool for checking:
BREACH? Custom-made?

Process

Natural Language Contract:
IF accel $>=$ 3 G && accel $<=$ 4 G for duration $<$ 5 sec THEN severity $=$ S2

$$\rightarrow$$

Signal Temporal Logic (STL):
$\diamond_{[0,5)}(accel >= 3 \ \&\& \ accel <= 4)$



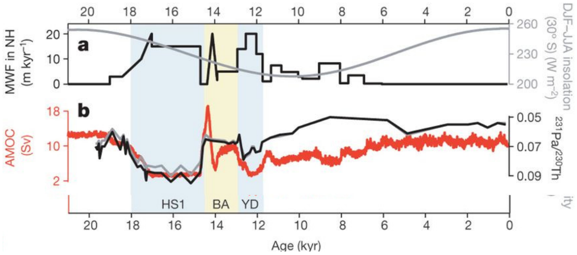
Tool for checking:
BREACH? Custom-made?
Reporting:

Process

Natural Language Contract:
IF accel >= 3 G && accel <= 4 G for duration < 5 sec THEN severity = S2

$\rightarrow$

Signal Temporal Logic (STL):
$\diamond_{[0,5)}(\mathit{accel} >= 3 \ \&\& \ \mathit{accel} <= 4)$



Tool for checking:
BREACH? Custom-made?
Reporting:
Robustness? Visualization?