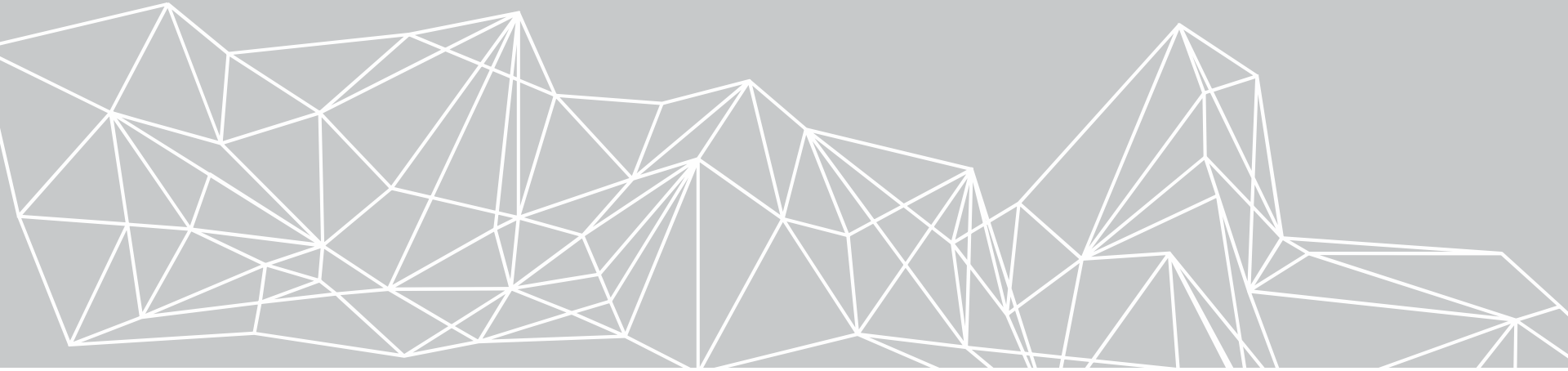


# EXECUTION OF UTP TEST CASES USING FUML



Niels Hoppe  
EXE 2018 in Copenhagen, Denmark  
October 14th, 2018

# INTRODUCTION

Niels Hoppe

Student in working group of Marc-Florian Wendland, SQC

[niels.hoppe@fokus.fraunhofer.de](mailto:niels.hoppe@fokus.fraunhofer.de)

Co-author

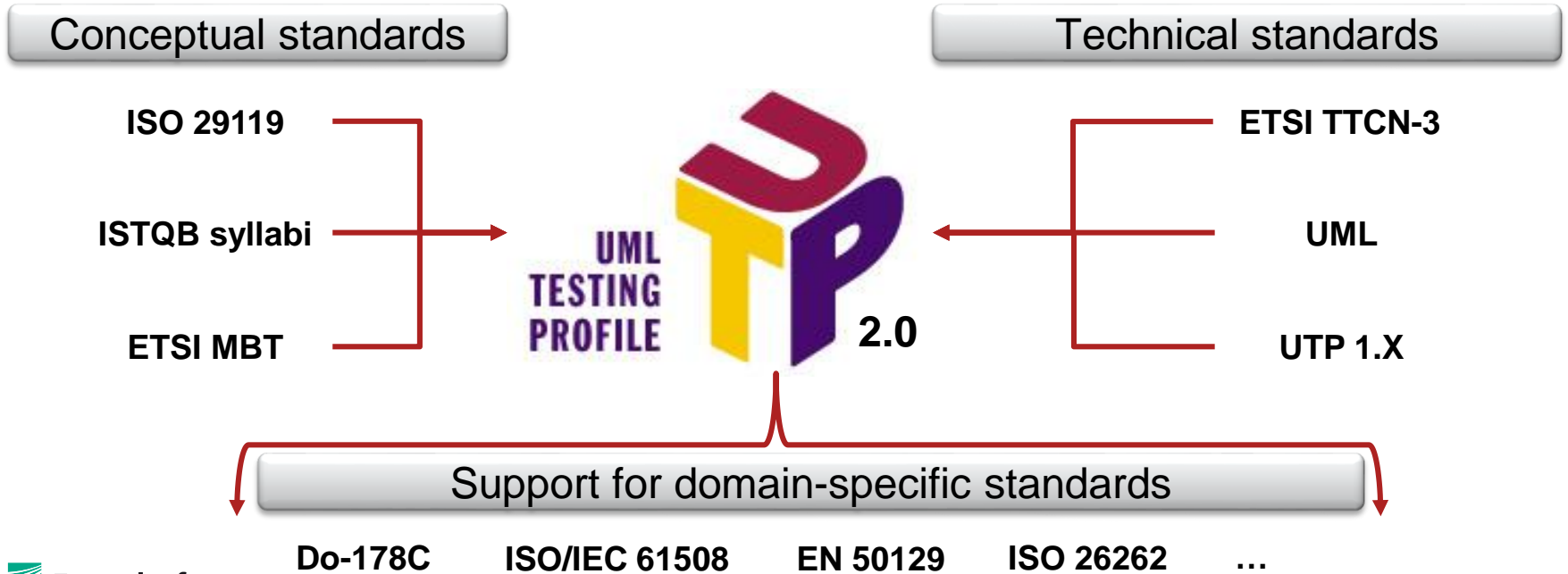
# INTRODUCTION

## Agenda

1. Introduction and motivation
2. Executable UTP test models
3. Mapping and transformation
4. Execution in Moka
5. Conclusion

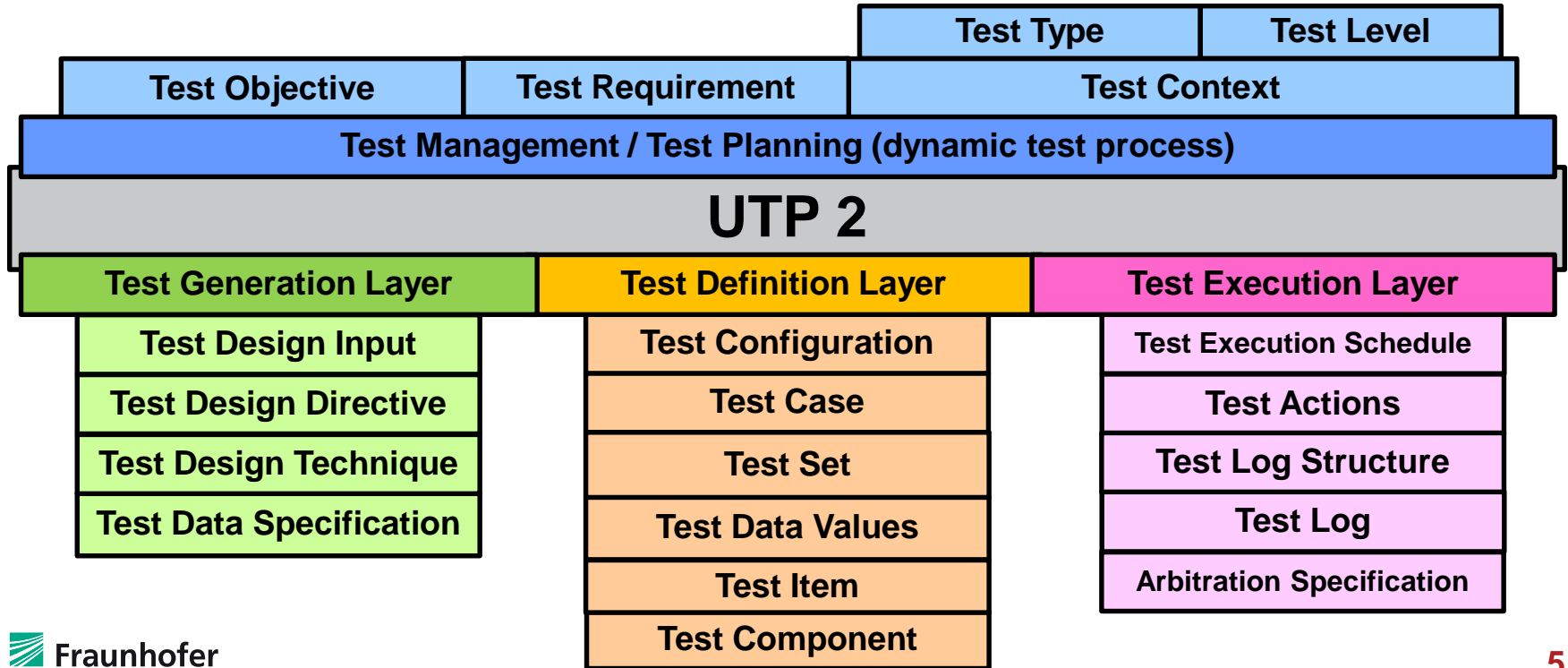
# INTRODUCTION TO UTP

## Influencing standards



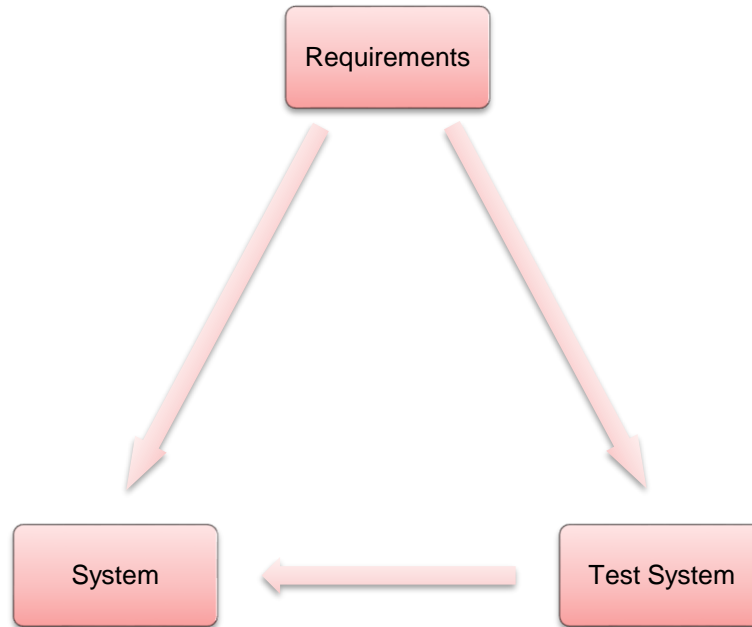
# INTRODUCTION TO UTP

## Conceptual overview of UTP



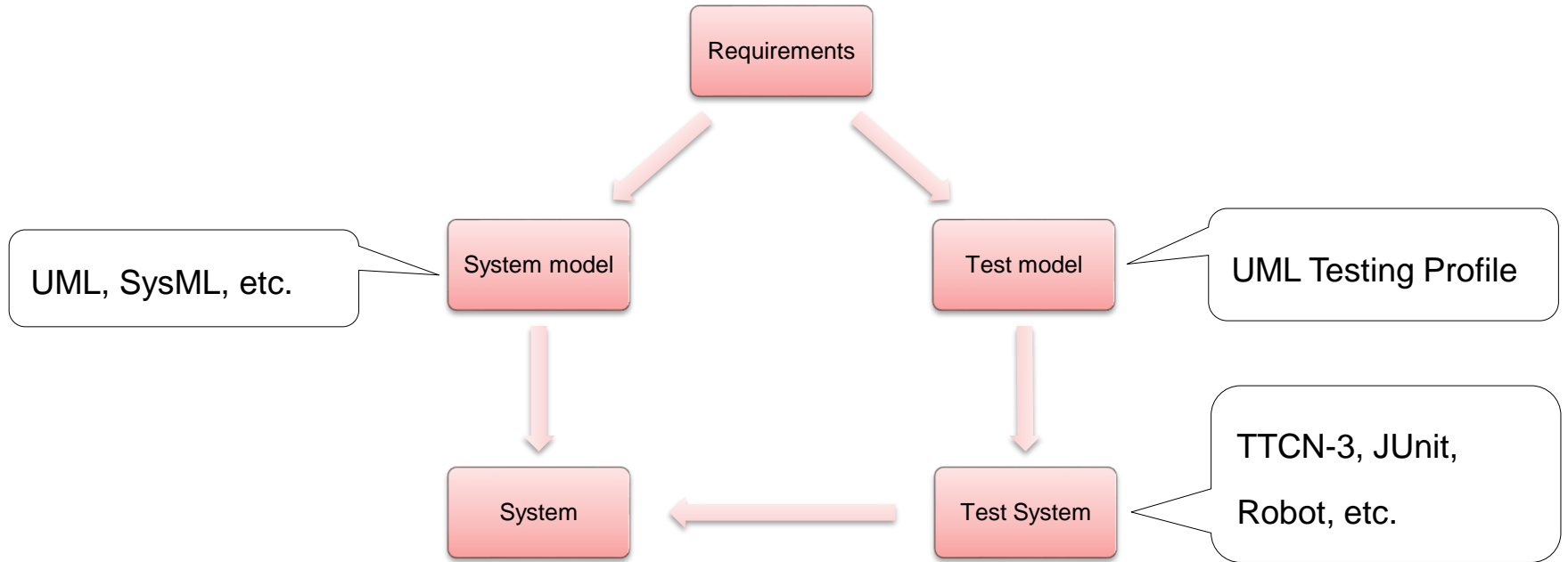
# MOTIVATION

## Classic Software Engineering



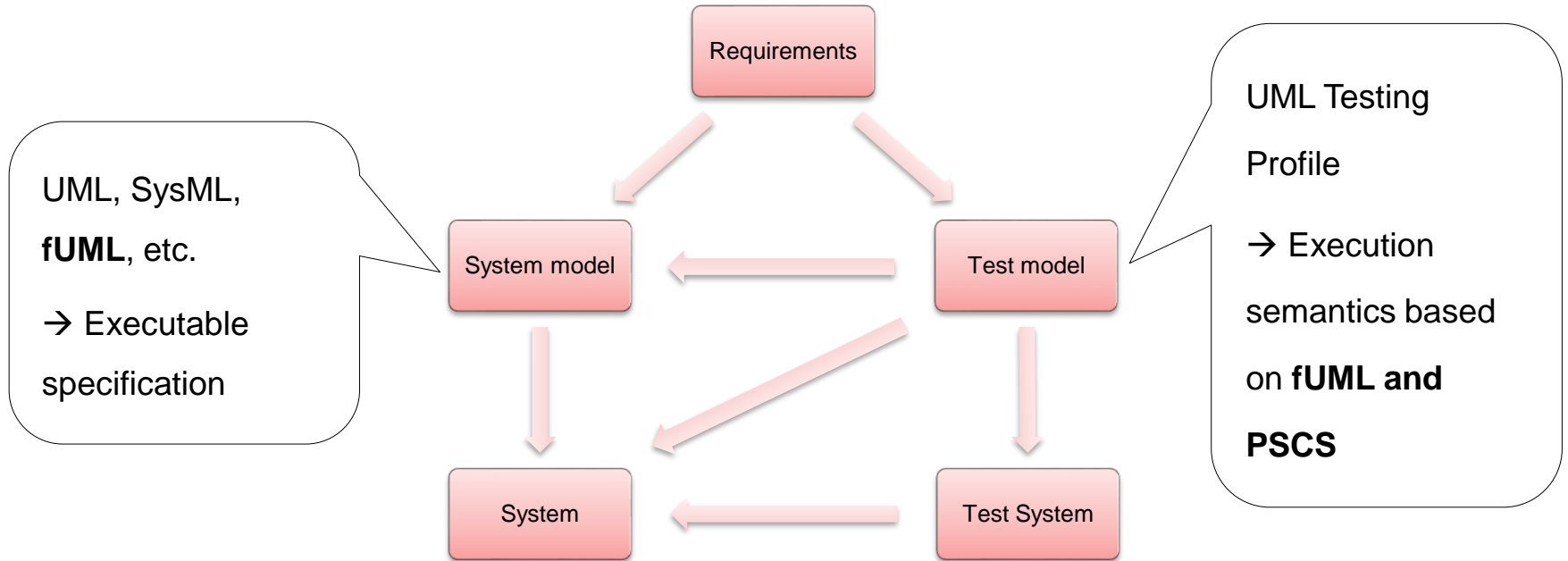
# MOTIVATION

## Model-Driven Software Engineering



# MOTIVATION

## Model-Based Shift Left Testing





# MOTIVATION

## Contributions of this paper

1. Requirements and constraints for executable UTP test models
2. Examples for executable UTP test model and adaptation model
3. Mapping and QVTo transformation
4. Execution environment based on Eclipse and Moka

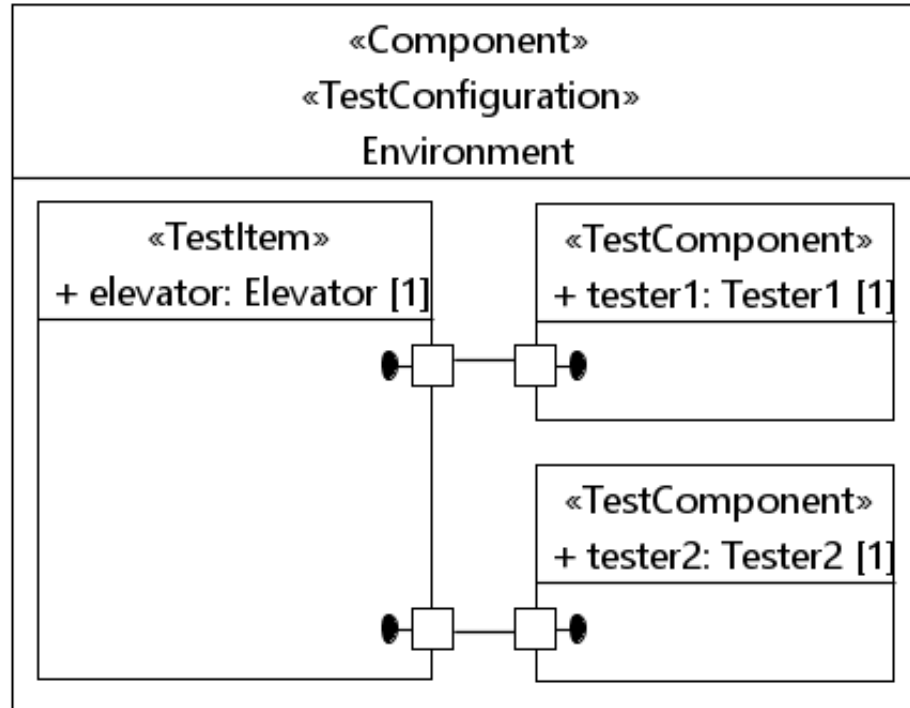
# EXECUTABLE UTP TEST MODELS

## Structure overview

- Test context
  - Test set 1
    - Test configuration
    - Test case 1.1
    - Test case 1.2
  - Test set 2

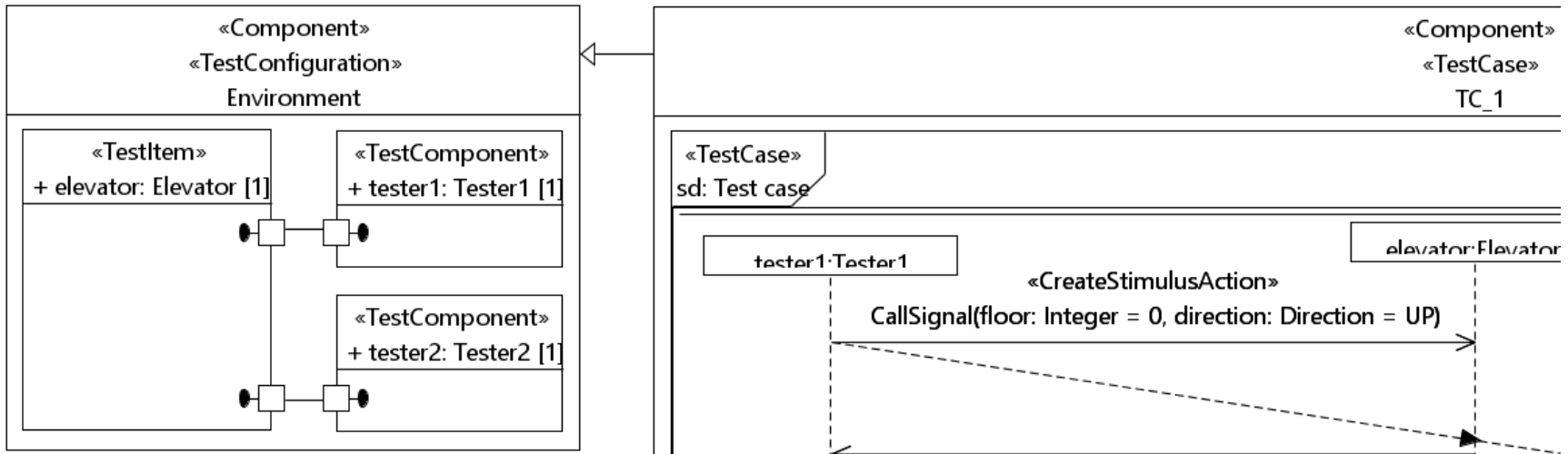
# EXECUTABLE UTP TEST MODELS

## Test configuration



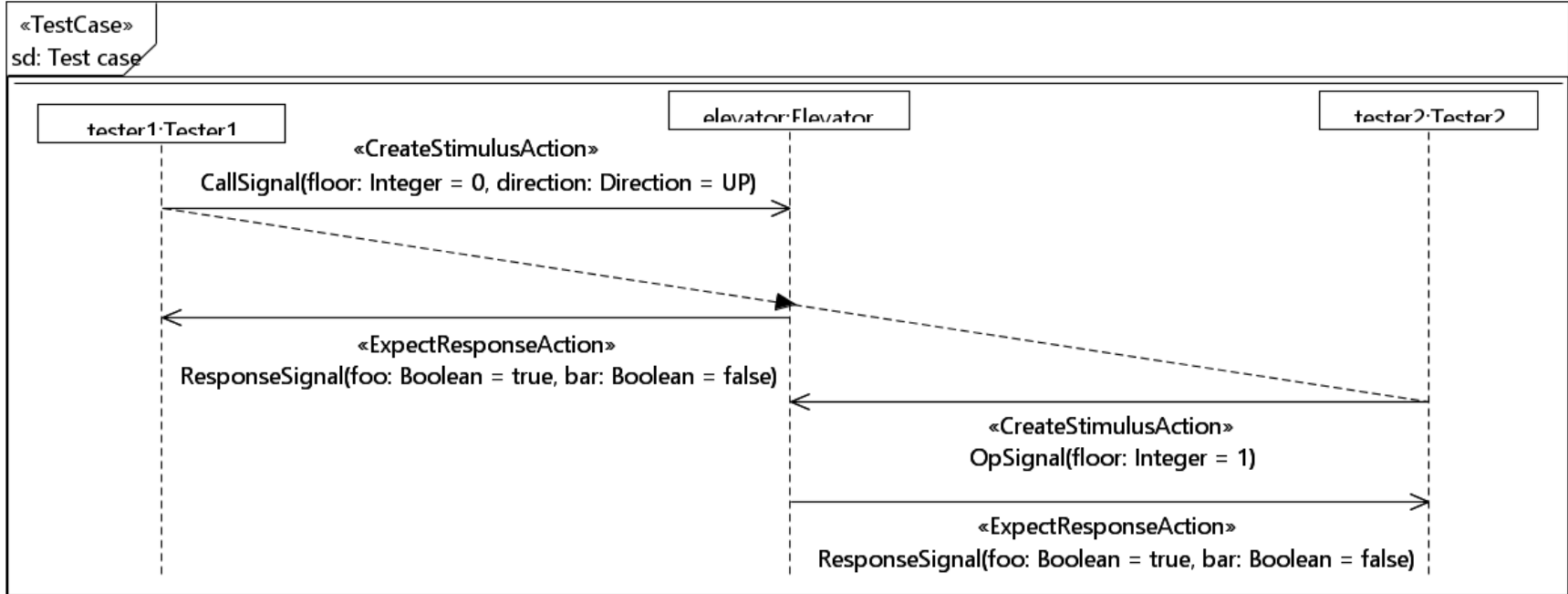
# EXECUTABLE UTP TEST MODELS

## Test configuration and test cases



# EXECUTABLE UTP TEST MODELS

## Test case behavior



# MAPPING AND TRANSFORMATION

## Transformation

- Transformation from **platform-independent** test model to **platform-specific** test model
  - Platform is fUML and PSCS
- Mapping rules for:
  - Test sets
  - Test cases
  - Test components and actions

# MAPPING AND TRANSFORMATION

## Test sets

- (Structure)
- Behaviors:
  1. Setup
  2. Teardown
  3. Main
  4. Factory

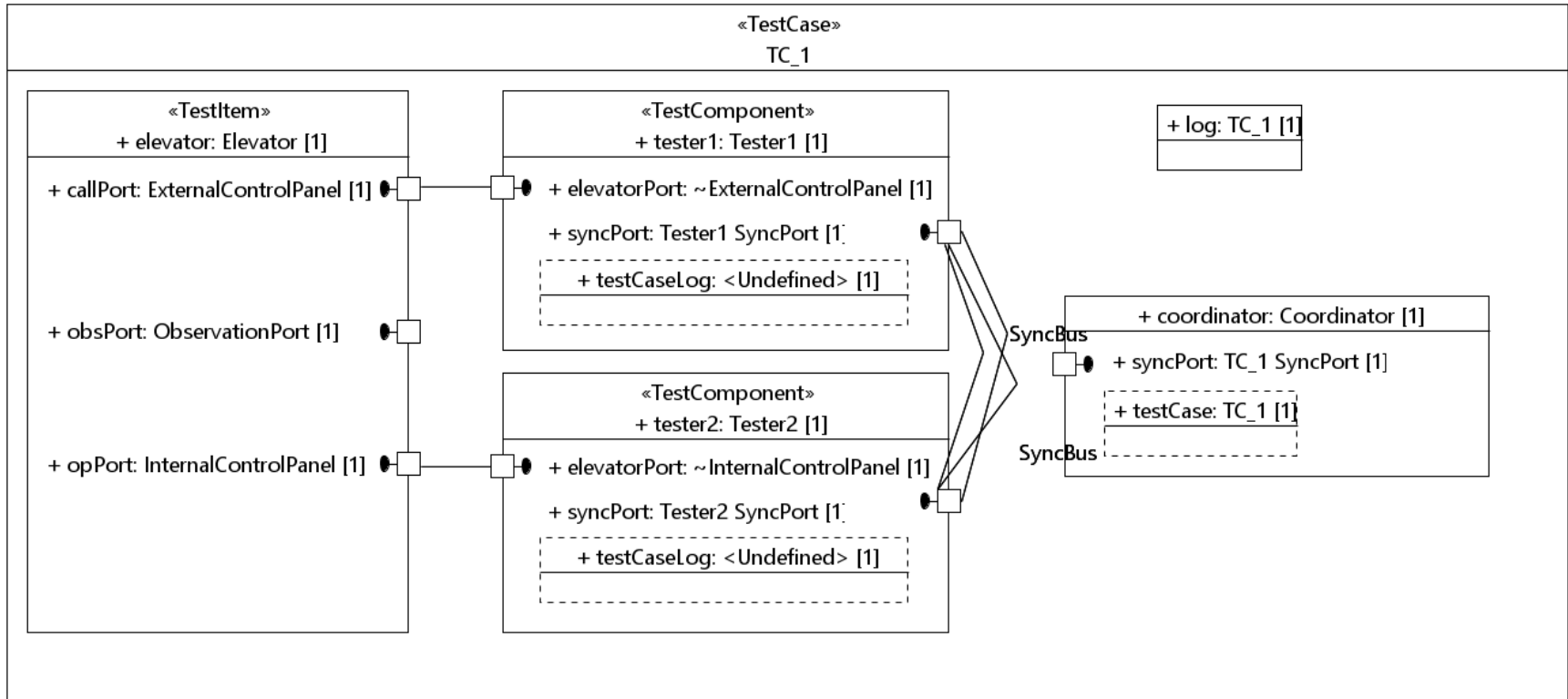
# MAPPING AND TRANSFORMATION

## Test cases

- Structure
  1. Resolution of Generalization
  2. Property coordinator: Component to coordinate test components
  3. Owned Connector: Synchronization Bus
- Behaviors:
  1. Setup
  2. Teardown
  3. Main
  4. Factory
  5. <<Create>> Constructor

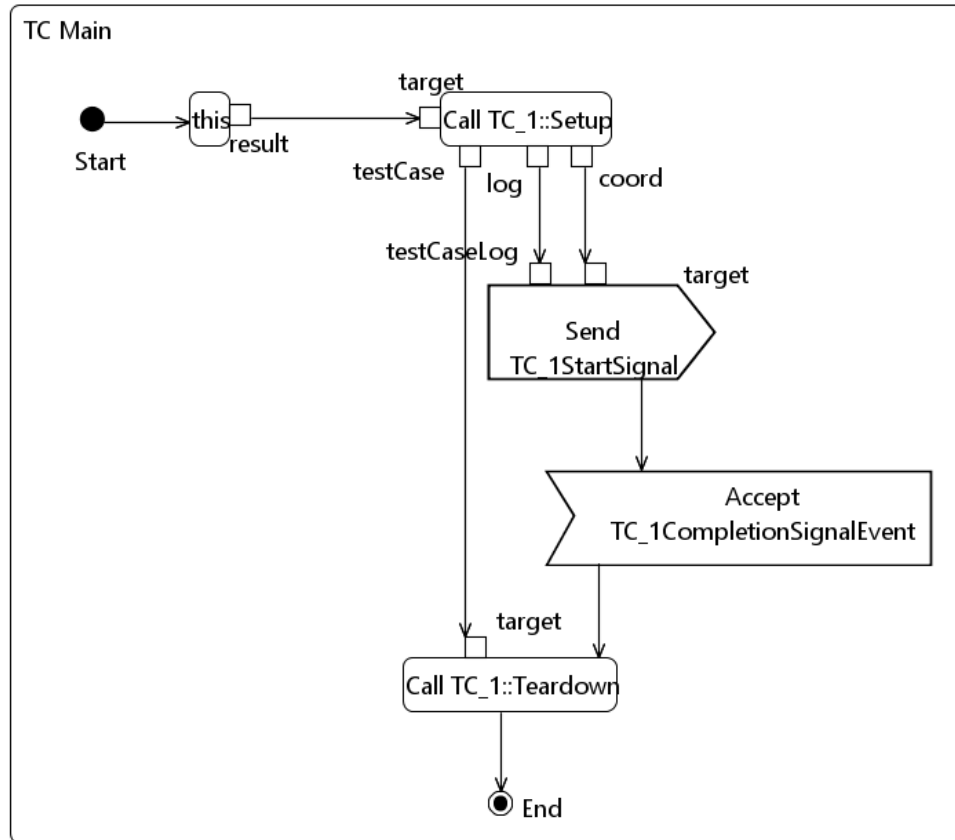


# MAPPING AND TRANSFORMATION



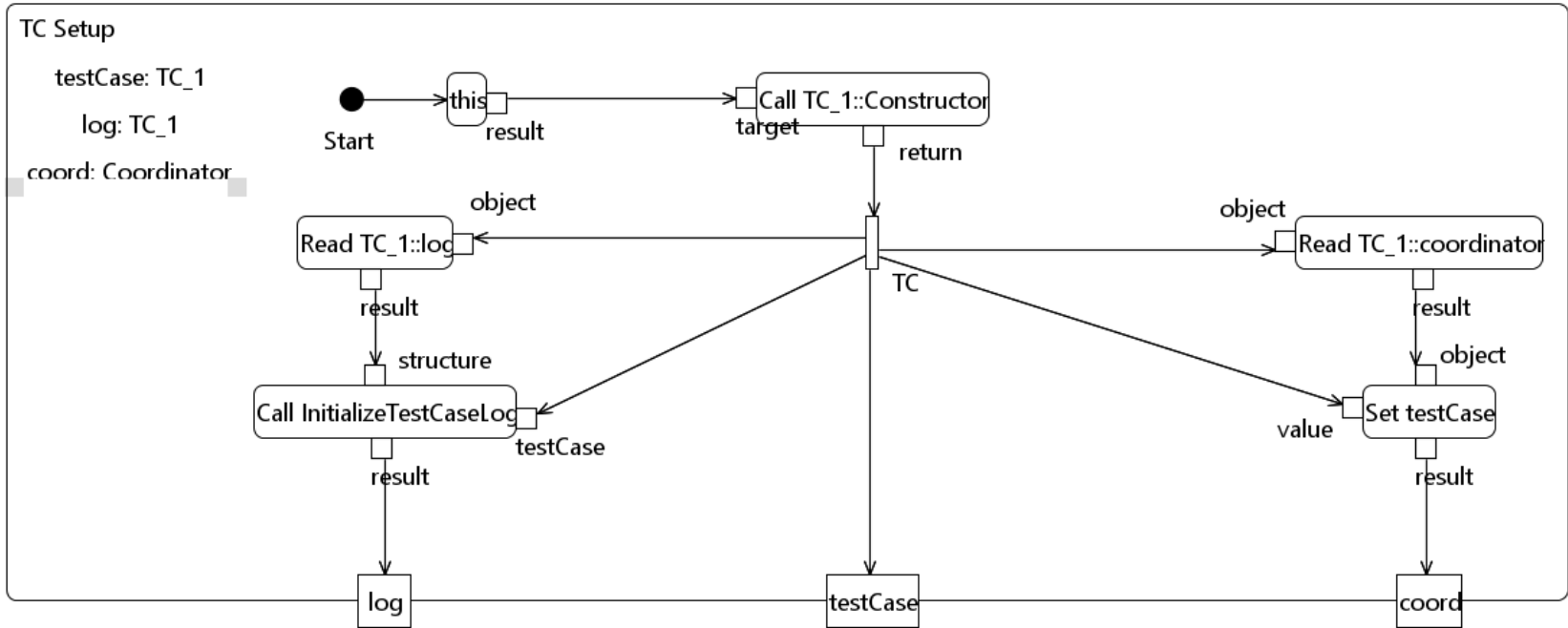
# MAPPING AND TRANSFORMATION

## Test cases



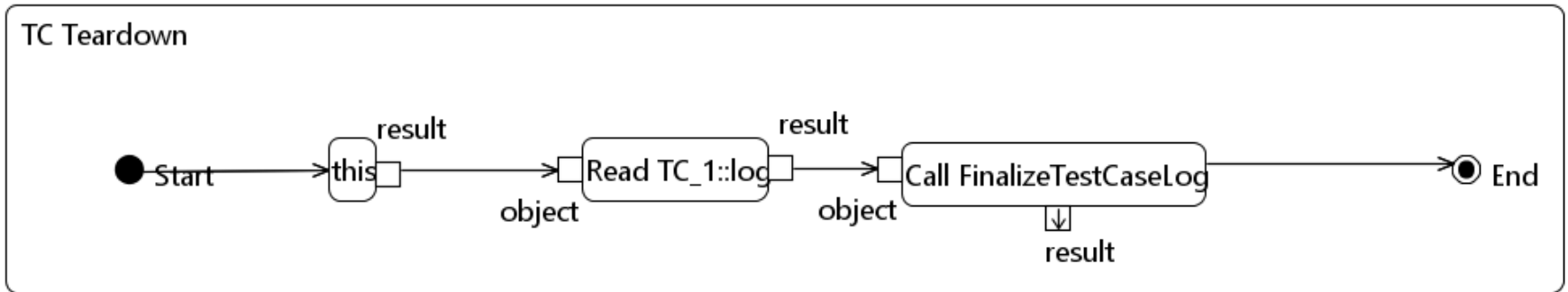
# MAPPING AND TRANSFORMATION

## Test cases



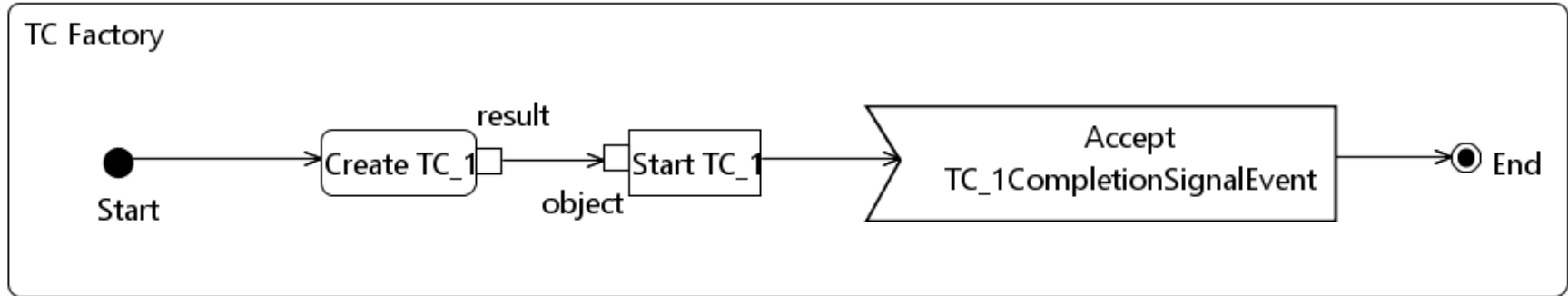
# MAPPING AND TRANSFORMATION

## Test cases



# MAPPING AND TRANSFORMATION

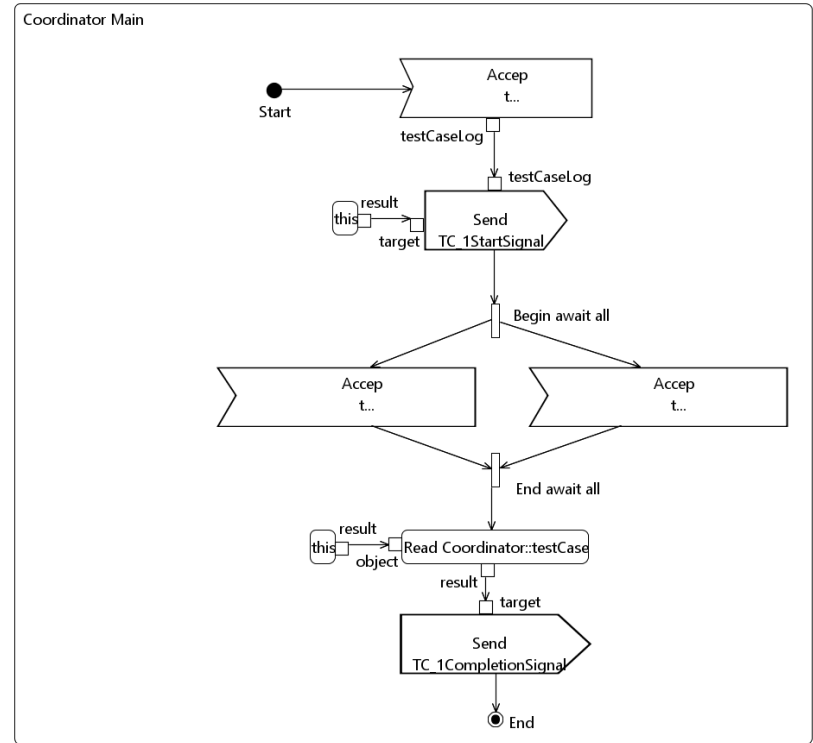
## Test cases



# MAPPING AND TRANSFORMATION

## Test case coordinator

- Structure
  1. Property testCase: Reference to containing test case
- Behavior
  1. Main
    - Synchronization of test components through StartSignal and CompletionSignal
    - Distribute test case log to test components



# MAPPING AND TRANSFORMATION

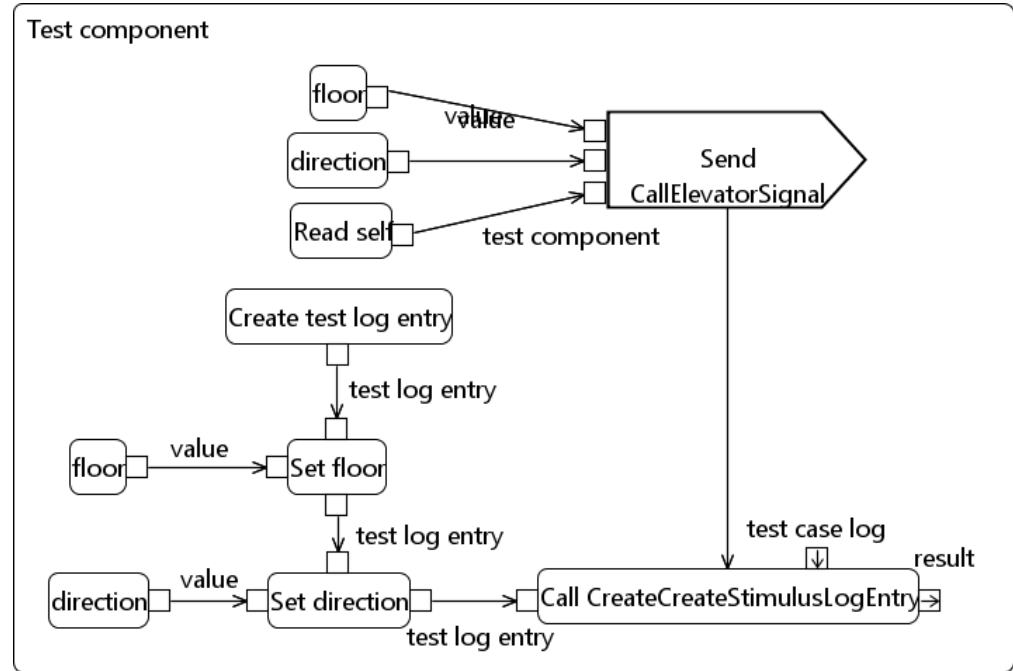
## Test components

- (Structure)
- Behaviors:
  1. Main, implements test actions
    - CreateStimulus
    - ExpectResponse
    - Logging
    - Synchronization via GeneralOrderings

# MAPPING AND TRANSFORMATION

## Test components

- (Structure)
- Behaviors:
  1. Main, implements test actions
    - **CreateStimulus**
    - ExpectResponse
    - **Logging**
    - Synchronization via GeneralOrderings

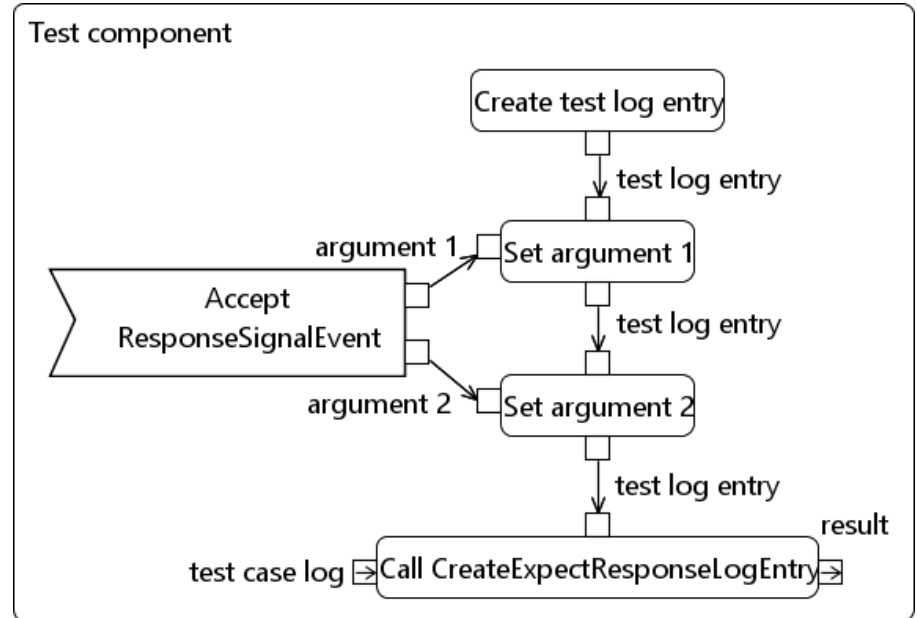




# MAPPING AND TRANSFORMATION

## Test components

- (Structure)
- Behaviors
  1. Main, implements test actions
    - CreateStimulus
    - **ExpectResponse**
    - **Logging**
    - Synchronization via GeneralOrderings



# EXECUTION IN MOKA

## Extension and customization of Moka

- Implementation of OpaqueBehaviors in Java:
  - Initialization and finalization of test logs
  - Creation of test log entries
- Construction strategy for N-ary Connectors

# EXECUTION IN MOKA

## Logs generated during execution

The screenshot displays the MOKA software interface. On the left, the 'Model Explorer' shows a tree structure under 'Log Model' with 'TestSetLogs for Two testers with adapter' expanded to 'TestCaseLogs for TC\_1'. A specific log entry, '«TestCaseLog» TCL @ 1522156580953', is selected and highlighted in blue. The right pane shows the 'Properties' view for this selected entry. The title bar reads '«TestCaseLog» TCL @ 1522156580953'. Below the title bar, there are tabs for 'UML', 'Comments', 'Profile', and 'Advanced'. The 'Applied stereotypes:' section is active, displaying a list of stereotypes and their values:

- executionStart: ValueSpecification [1] = null
- executingEntity: ValueSpecification [\*] = []
- executionDuration: Duration [0..1] = null
- testLogEntry: TestLogEntry [\*] = [MELE @ 1522156608336,
- verdict: ValueSpecification [0..1] = null

# CONCLUSION AND FUTURE WORK

## Achievements

- ✓ First definition of executable subset of UTP
- ✓ Mapping and transformation
- ✓ Execution and logging
- ✓ First steps towards precise execution semantics for UML Testing Profile

## Next steps

- Integration with analysis and arbitration of logs
- Support for operations, complex actions, etc.
- Proof of concept for real-world adapter
- Improve efficiency of implementation

THE END

