Universidad Nacional de San Agustín

# Data bases

## UMTG: a toolset to automatically generate system test cases from use case specifications

MSc. Vicente Machaca Arceda

2021

# Content

# UMTG




UMTG, a toolset that generates executable system test cases by exploiting the behavioural information implicitly described in use case specifications [1].

UMTG generates OCL constraints for:

- Use case preconditions.
- Postconditions.
- Conditional steps.

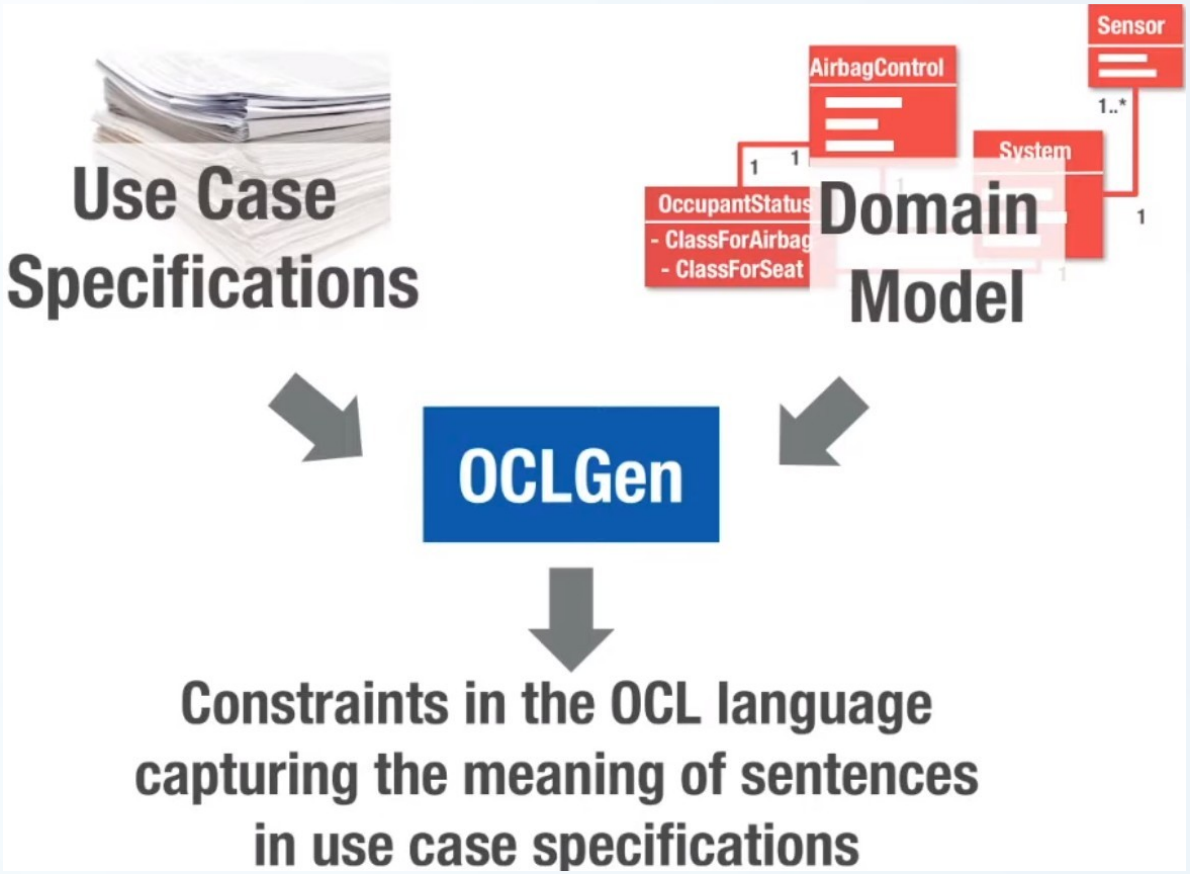
.


Table: Examples of OCL constraints.

| **Constraint** | **OCL Equivalent** |
|---|---|
| The age of a person is not negative | **context** Person **inv**: self.age >=0 |
| A Person has 2 parents at max | **context** Person **inv**: self.parents->size()<=2 |
| The system sets the occupancy status to empty | **BodySense**.allinstances()->forAll( i\|i.occupancyStatus = **Occupancy::Empty** ) |

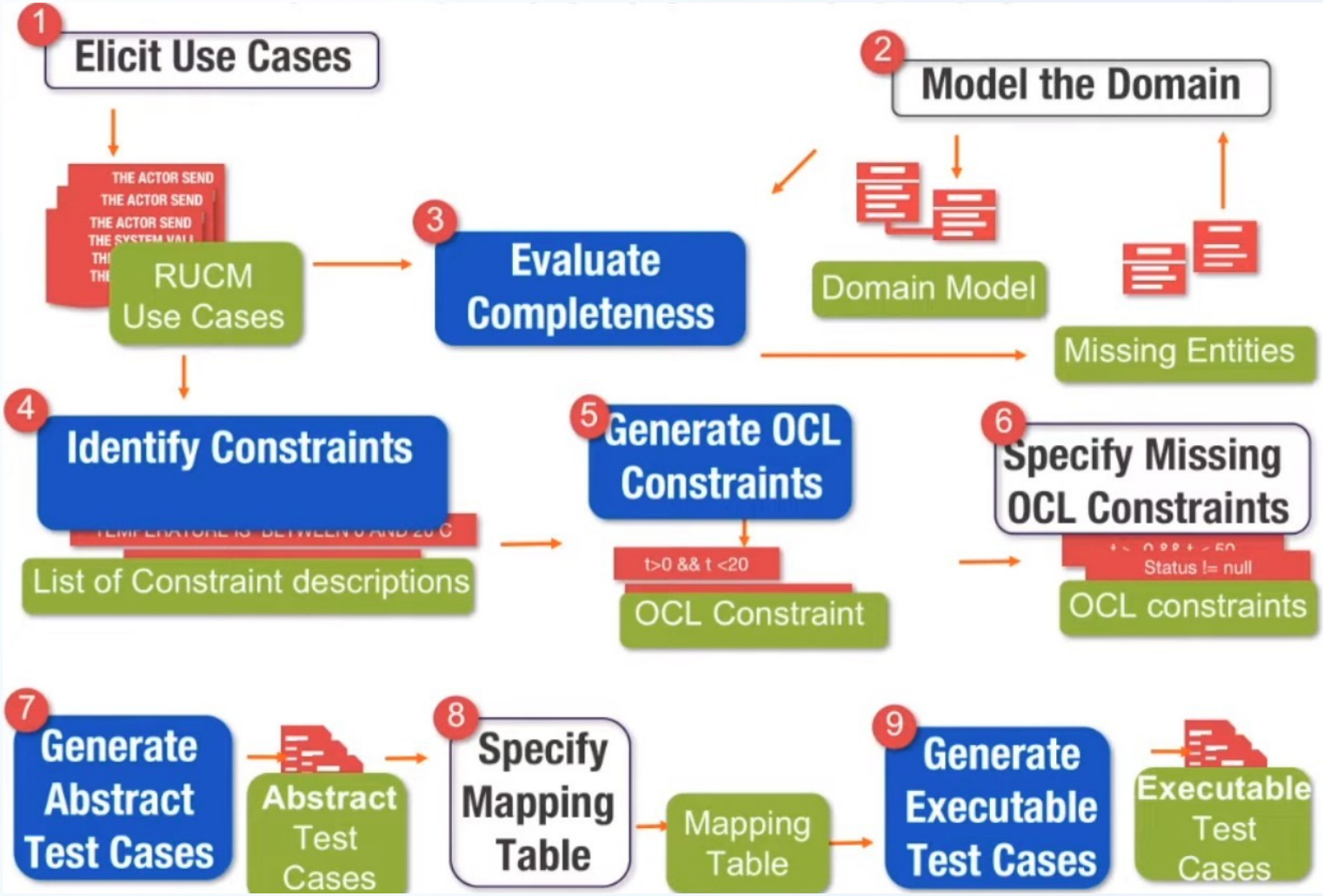
# UMTG




Use Case Specifications
AirbagControl
Sensor
System
OccupantStatus
- ClassForAirbag
- ClassForSeat
Domain Model
OCLGen
Constraints in the OCL language capturing the meaning of sentences in use case specifications

# Workflow




1
Elicit Use Cases
THE ACTOR SEND
THE ACTOR SEND
THE ACTOR SEND
RUCM
Use Cases
2
Model the Domain
Domain Model
3
Evaluate
Completeness
Missing Entities
4
Identify Constraints
List of Constraint descriptions
5
Generate OCL
Constraints
t>0 && t <20
OCL Constraint
6
Specify Missing
OCL Constraints
Status != null
OCL constraints
7
Generate
Abstract
Test Cases
Abstract
Test
Cases
8
Specify
Mapping
Table
Mapping
Table
9
Generate
Executable
Test Cases
Executable
Test
Cases

# Workflow
## RUCM use cases



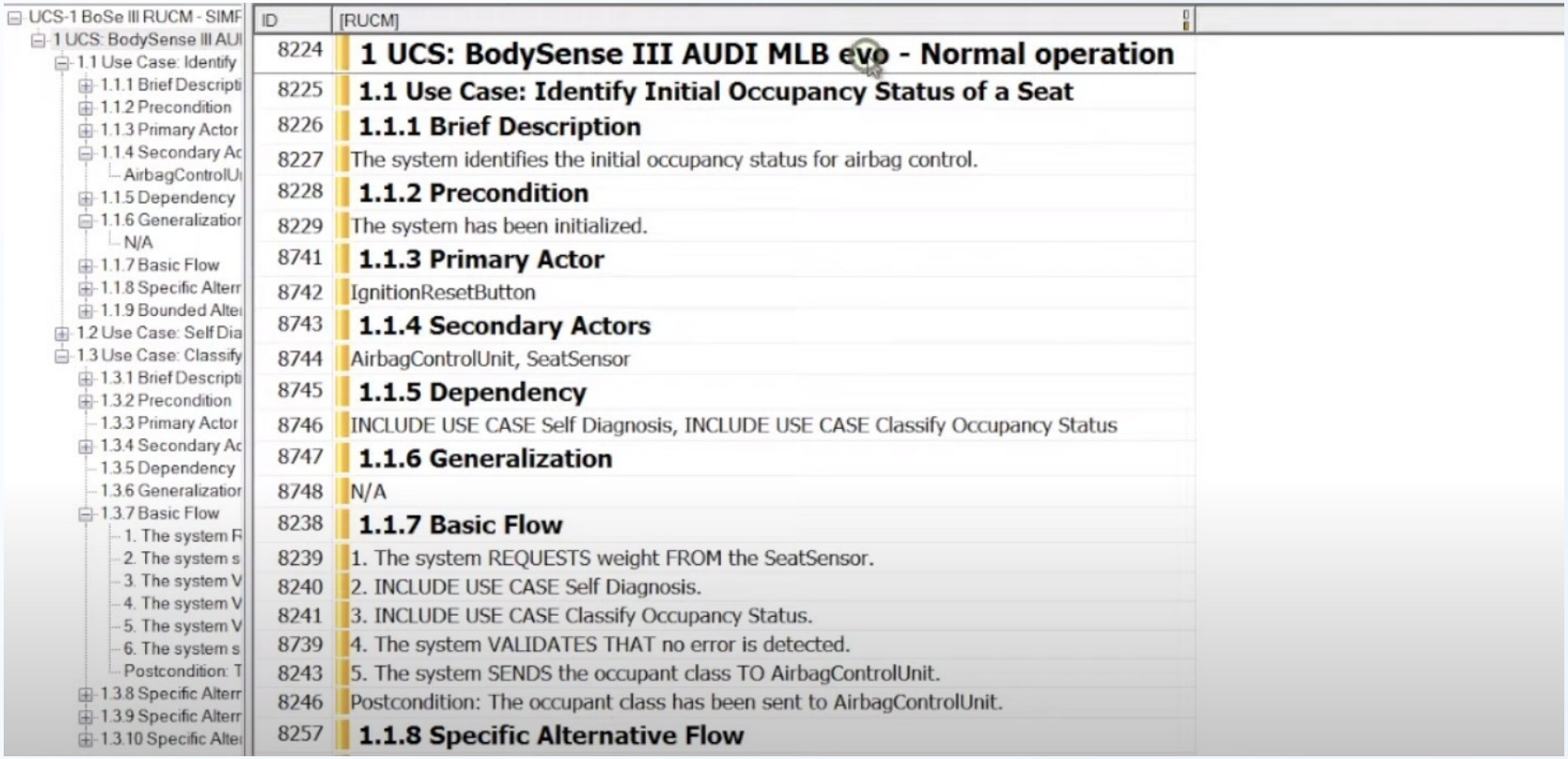



Figure: RUCM Use Case Specification in IBM DOORS.

# Workflow
## OCL constraints



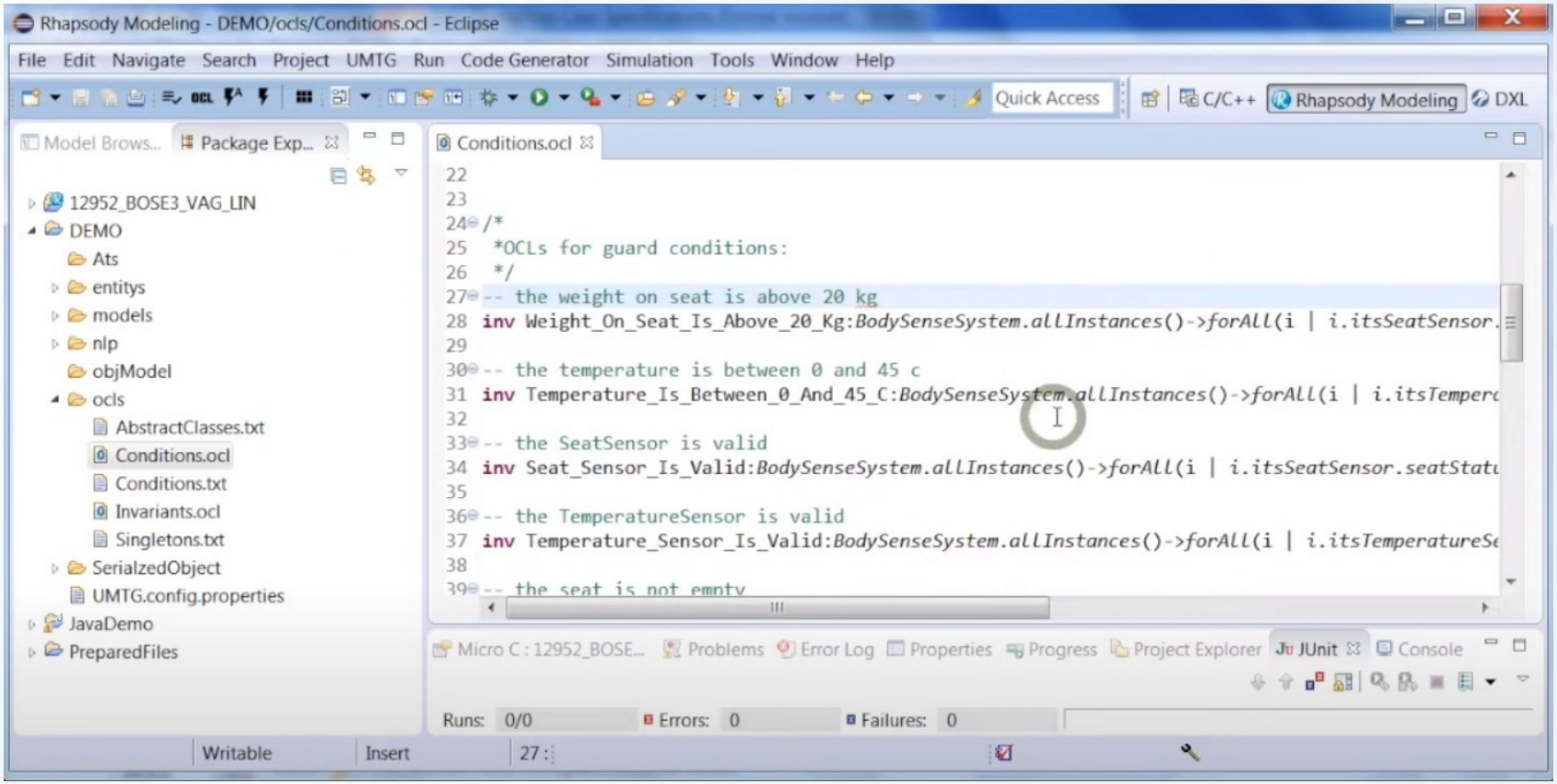



Figure: OCL constraints generated by UMTG.

# Workflow
## Abstract test cases



```
!# CoveredModulePath = /FSE-Demo/Use Case Specifications/UCS-1 BoSe III RUCM - SIMPLIFIED
!# CoveredFlowID = 8238, 8579, 8516
# The system has been initialized.
# The system REQUESTS weight FROM the SeatSensor.
# The system sets MeasurementDeviceError to not detected.
# The system REQUESTS TemperatureSensorStatus FROM the TemperatureSensor.
# [TRUE] The system VALIDATES THAT the TemperatureSensor is valid.
# The system REQUESTS SeatStatus FROM the SeatSensor.
# [TRUE] The system VALIDATES THAT the SeatSensor is valid.
# Postcondition: There is no MeasurementDeviceError detected.
# The system REQUESTS temperature FROM the TemperatureSensor.
# The system sets TemperatureError to not detected.
# [TRUE] The system VALIDATES THAT the temperature is between 0 and 45 c.
# [TRUE] The system VALIDATES THAT the seat is not empty.
# [TRUE] The system VALIDATES THAT the weight on seat is above 20 kg.
# The system sets occupant class to adult.
# Postcondition: The adult occupant class has been derived.
# [TRUE] The system VALIDATES THAT no error is detected.
# The system SENDS the occupant class TO AirbagControlUnit.
# Postcondition: The occupant class has been sent to AirbagControlUnit.



<SET>     BodySenseSystem.initialized = True

<INPUT>  SeatSensor.weight = 64
<INPUT>  TemperatureSensor.TemperatureSensorStatus = HWStatus_Valid
<INPUT>  SeatSensor.SeatStatus = HWStatus_Valid
<CHECK> There_is_no_MeasurementDeviceError_detected
```

Figure: Abstract test cases.

# Workflow
## Mapping table



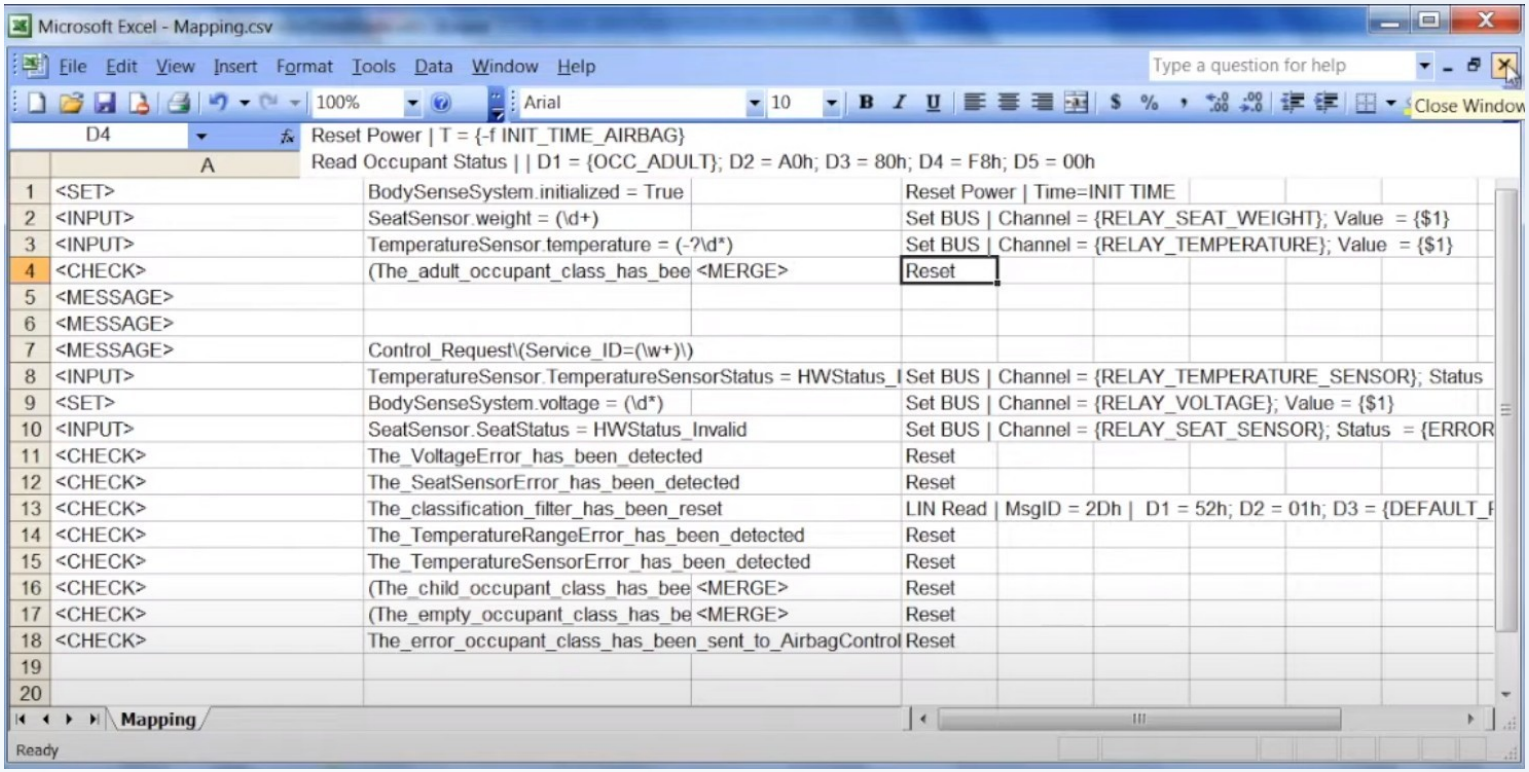



Figure: Mapping table.

# Workflow
## Test case



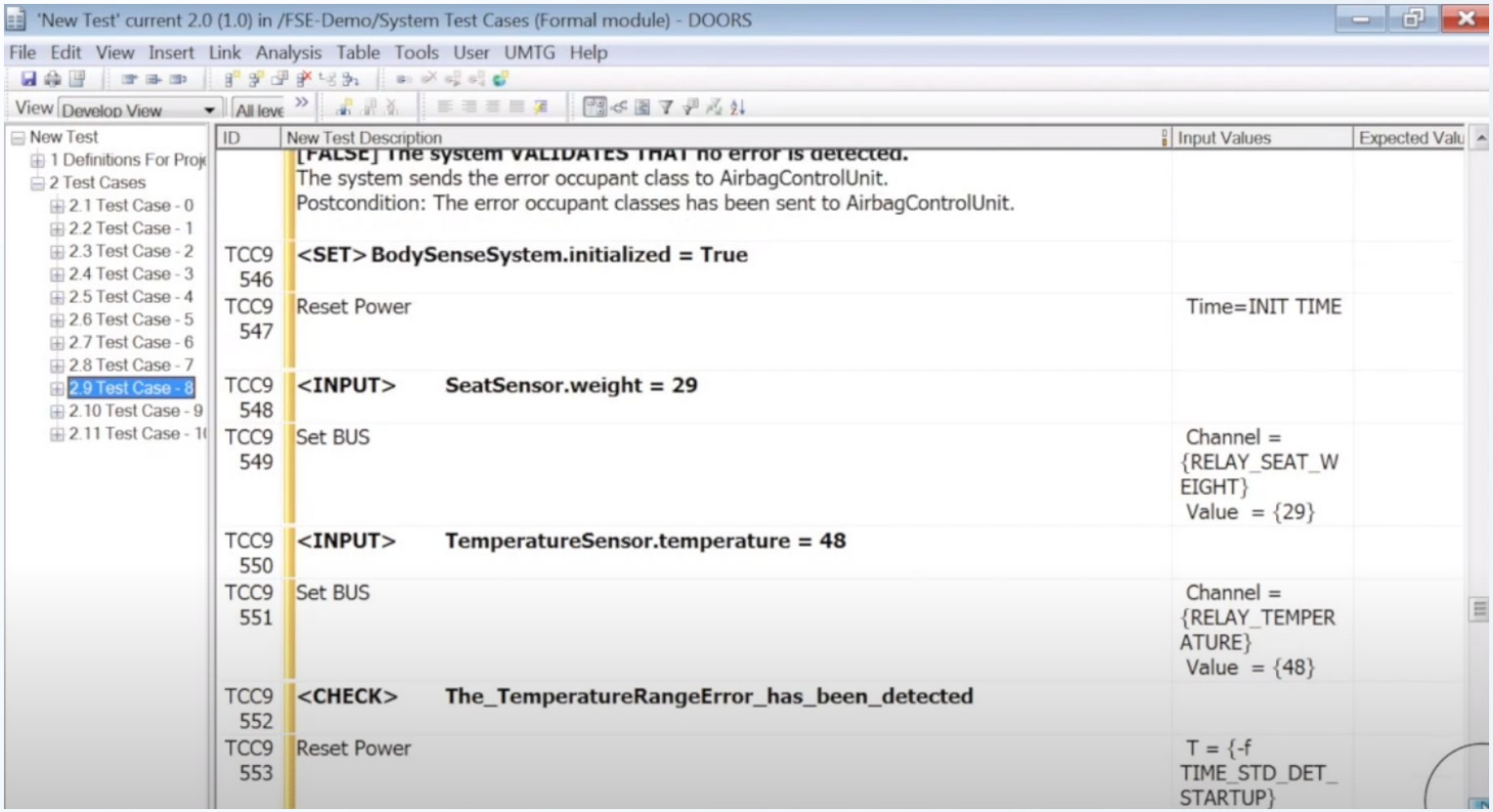



Figure: Test case.

# References I




[1] C. Wang, F. Pastore, A. Goknil, L. C. Briand, and Z. Iqbal, “Umtg: a toolset to automatically generate system test cases from use case specifications,” in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, 2015, pp. 942–945.