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.

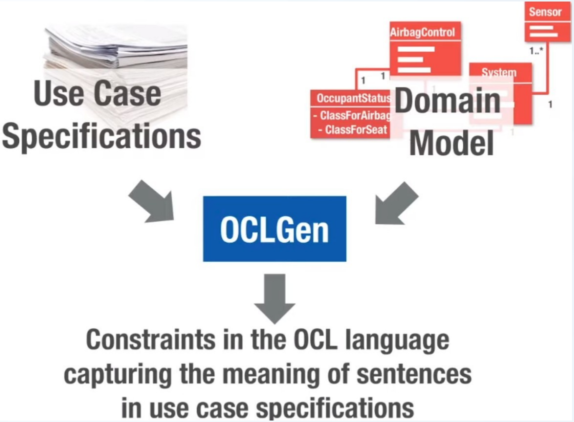
.

- Automatic Generation of System Test Cases from Use Case Specifications [2].
- **UMTG: a toolset to automatically generate system test cases from use case specifications** [1].
- Automated generation of constraints from use case specifications to support system testing [3].
- Automatic Generation of Acceptance Test Cases from Use Case Specifications: an NLP-based Approach [4]

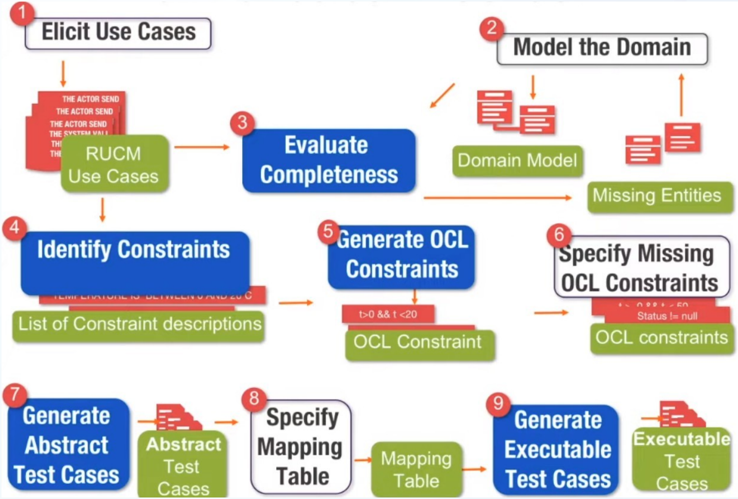
Table: Examples of OCL constraints.

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
Sensor
AirbagControl
1..*
1
1
System
OccupantStatus
- ClassForAirbag
- ClassForSeat
Domain Model
1
OCLGen
Constraints in the OCL language capturing the meaning of sentences in use case specifications

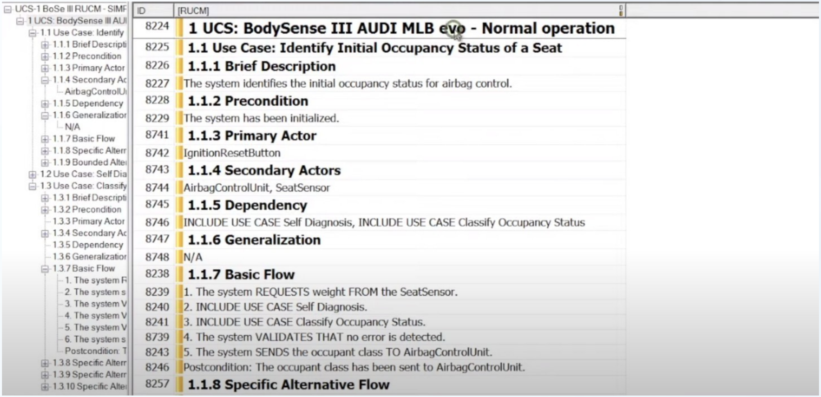
Workflow



1
Elicit Use Cases
THE ACTOR SEND
THE ACTOR SEND
THE ACTOR SEND
RUCM
Use Cases
3
Evaluate
Completeness
2
Model the Domain
Domain Model
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



ID	[RUCM]
8224	**1 UCS: BodySense III AUDI MLB evo - Normal operation**
8225	**1.1 Use Case: Identify Initial Occupancy Status of a Seat**
8226	**1.1.1 Brief Description**
8227	The system identifies the initial occupancy status for airbag control.
8228	**1.1.2 Precondition**
8229	The system has been initialized.
8741	**1.1.3 Primary Actor**
8742	IgnitionResetButton
8743	**1.1.4 Secondary Actors**
8744	AirbagControlUnit, SeatSensor
8745	**1.1.5 Dependency**
8746	INCLUDE USE CASE Self Diagnosis, INCLUDE USE CASE Classify Occupancy Status
8747	**1.1.6 Generalization**
8748	N/A
8238	**1.1.7 Basic Flow**
8239	1. The system REQUESTS weight FROM the SeatSensor.
8240	2. INCLUDE USE CASE Self Diagnosis.
8241	3. INCLUDE USE CASE Classify Occupancy Status.
8739	4. The system VALIDATES THAT no error is detected.
8243	5. The system SENDS the occupant class TO AirbagControlUnit.
8246	Postcondition: The occupant class has been sent to AirbagControlUnit.
8257	**1.1.8 Specific Alternative Flow**

Figure: RUCM Use Case Specification in IBM DOORS.

Workflow

OCL constraints

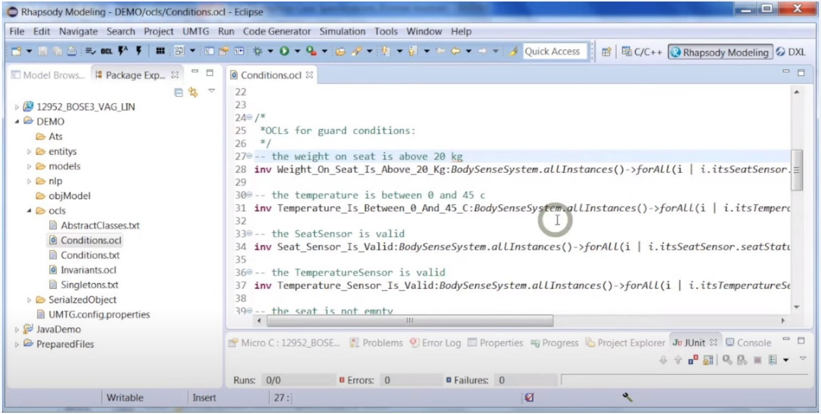


Figure: OCL constraints generated by UMTG.

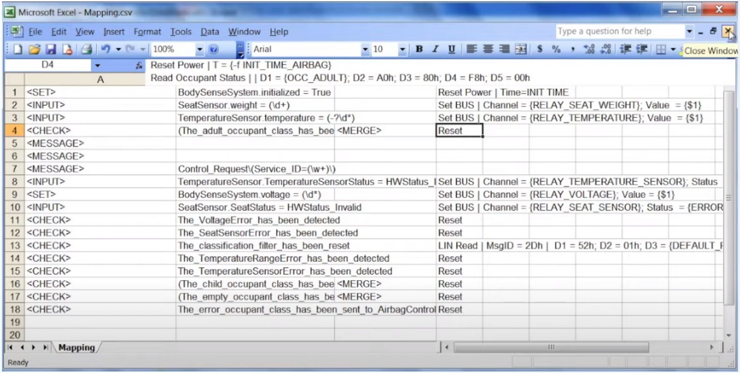
Workflow
Abstract test cases


```
!# 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



Microsoft Excel - Mapping.csv

D4: Reset Power | T = {-f INIT_TIME_AIRBAG}
Read Occupant Status | | D1 = {OCC_ADULT}; D2 = A0h; D3 = 80h; D4 = F8h; D5 = 00h

<table>
<tr><th></th><th>A</th><th></th><th></th><th></th></tr>
<tr><td>1</td><td><SET></td><td>BodySenseSystem.initialized = True</td><td></td><td>Reset Power | Time=INIT TIME</td></tr>
<tr><td>2</td><td><INPUT></td><td>SeatSensor.weight = (\d+)</td><td></td><td>Set BUS | Channel = {RELAY_SEAT_WEIGHT}; Value = {$1}</td></tr>
<tr><td>3</td><td><INPUT></td><td>TemperatureSensor.temperature = (-?\d*)</td><td></td><td>Set BUS | Channel = {RELAY_TEMPERATURE}; Value = {$1}</td></tr>
<tr><td>4</td><td><CHECK></td><td>(The_adult_occupant_class_has_bee</td><td><MERGE></td><td>Reset</td></tr>
<tr><td>5</td><td><MESSAGE></td><td></td><td></td><td></td></tr>
<tr><td>6</td><td><MESSAGE></td><td></td><td></td><td></td></tr>
<tr><td>7</td><td><MESSAGE></td><td>Control_Request\(Service_ID=(\w+)\)</td><td></td><td></td></tr>
<tr><td>8</td><td><INPUT></td><td>TemperatureSensor.TemperatureSensorStatus = HWStatus_I</td><td></td><td>Set BUS | Channel = {RELAY_TEMPERATURE_SENSOR}; Status</td></tr>
<tr><td>9</td><td><SET></td><td>BodySenseSystem.voltage = (\d*)</td><td></td><td>Set BUS | Channel = {RELAY_VOLTAGE}; Value = {$1}</td></tr>
<tr><td>10</td><td><INPUT></td><td>SeatSensor.SeatStatus = HWStatus_Invalid</td><td></td><td>Set BUS | Channel = {RELAY_SEAT_SENSOR}; Status = {ERROR</td></tr>
<tr><td>11</td><td><CHECK></td><td>The_VoltageError_has_been_detected</td><td></td><td>Reset</td></tr>
<tr><td>12</td><td><CHECK></td><td>The_SeatSensorError_has_been_detected</td><td></td><td>Reset</td></tr>
<tr><td>13</td><td><CHECK></td><td>The_classification_filter_has_been_reset</td><td></td><td>LIN Read | MsgID = 2Dh | D1 = 52h; D2 = 01h; D3 = {DEFAULT_F</td></tr>
<tr><td>14</td><td><CHECK></td><td>The_TemperatureRangeError_has_been_detected</td><td></td><td>Reset</td></tr>
<tr><td>15</td><td><CHECK></td><td>The_TemperatureSensorError_has_been_detected</td><td></td><td>Reset</td></tr>
<tr><td>16</td><td><CHECK></td><td>(The_child_occupant_class_has_bee</td><td><MERGE></td><td>Reset</td></tr>
<tr><td>17</td><td><CHECK></td><td>(The_empty_occupant_class_has_be</td><td><MERGE></td><td>Reset</td></tr>
<tr><td>18</td><td><CHECK></td><td>The_error_occupant_class_has_been_sent_to_AirbagControl</td><td></td><td>Reset</td></tr>
<tr><td>19</td><td></td><td></td><td></td><td></td></tr>
<tr><td>20</td><td></td><td></td><td></td><td></td></tr>
</table>

Mapping

Ready

Figure: Mapping table.

Workflow

Test case

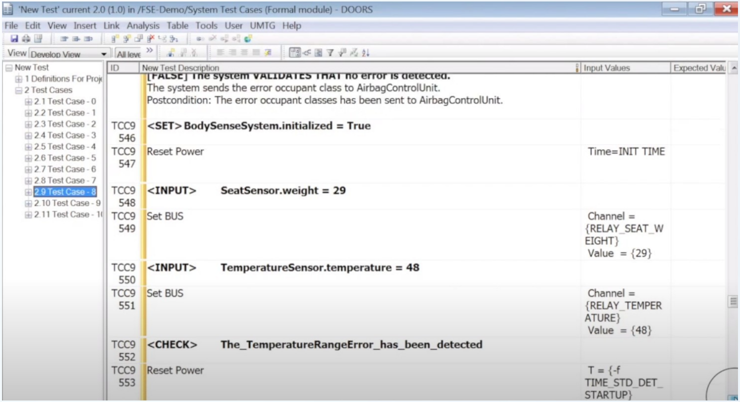


Figure: Test case.

Results

Table: Case study details.

Attribute	Detail
Use case specifications	06
Steps by each use case	25 - 50
Alternative flows by each use case	6 - 13

UMTG covers a total of **100 scenarios**, while manually written test cases cover **80 scenarios**.

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.

[2] C. Wang, F. Pastore, A. Goknil, L. Briand, and Z. Iqbal, "Automatic generation of system test cases from use case specifications," in *Proceedings of the 2015 international symposium on software testing and analysis*, 2015, pp. 385–396.

[3] C. Wang, F. Pastore, and L. Briand, "Automated generation of constraints from use case specifications to support system testing," in *2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST)*. IEEE, 2018, pp. 23–33.

References II

[4] C. Wang, F. Pastore, A. Goknil, and L. Briand, "Automatic generation of acceptance test cases from use case specifications: an nlp-based approach," *IEEE Transactions on Software Engineering*, 2020.