UmpleRun: a Dynamic Analysis Tool for Textually Modeled State Machines using Umple

Hamoud Aljamaan, Miguel Garzon, Timothy Lethbridge

CRuiSE (Complexity Reduction in Software Engineering) Research Group
University of Ottawa

Outline

- Introduction
- Umple
  - Car transmission model
- UmpleRun
- Car Transmission Dynamic Analysis
  - Successful case
  - Failed case
- Future work

Introduction

- Umple
  - A model-oriented programming language that allows modelers to model UML constructs textually or graphically
    - focus of this paper: state machines
  - Generate high quality code in a number of targeted programming languages (Java/Ruby/Php … etc)
- MOTL
  - trace specification at the model level for various modeling constructs using model level textual trace directives

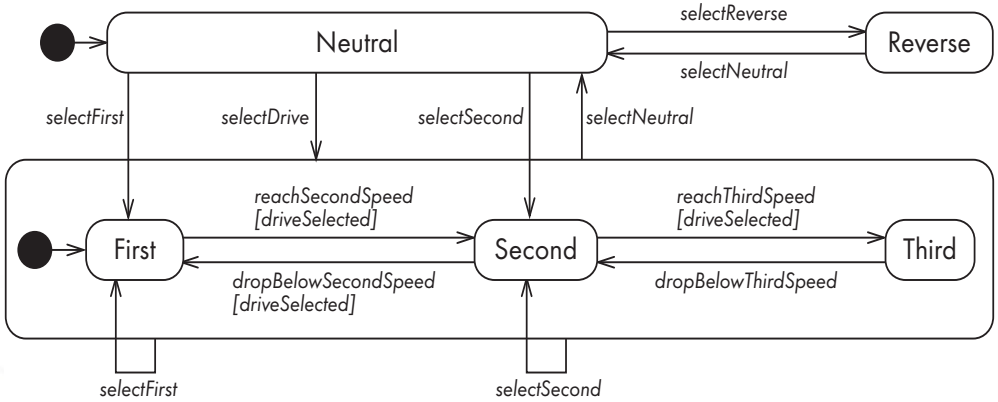
Introduction (cont'd)

- What is missing?
  - Validate the model's dynamic behavior.
  - Generate execution traces.
- Why?
  - High-level validation of model dynamic behavior.
  - White box testing of models.

Introduction (cont'd)

- UmpleRun
  - prototype tool – under continuous improvement.

Umple

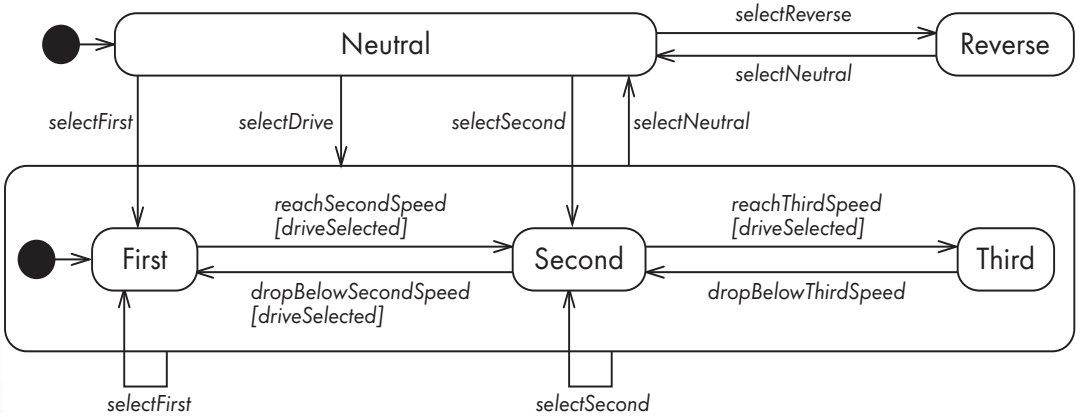


Umple

```
1  class CarTransmission {
2    Boolean driveSelected = false;
3    status {
4      neutral {
5        selectReverse -> reverse;
6        selectDrive -> drive;
7        selectFirst -> first;
8        selectSecond -> second;
9      }
10     reverse {
11       selectNeutral -> neutral;
12     }
13     drive {
14       exit / { driveSelected = false;}
15       selectNeutral -> neutral;
16       selectFirst -> first;
17       selectSecond -> second;
18       first {
19     reachSecondSpeed [driveSelected] -> second;
20       }
21       second {
22     reachThirdSpeed [driveSelected] -> third;
23  dropBelowSecondSpeed [driveSelected] -> first;
24       }
25       third {
26         dropBelowThirdSpeed -> second;
27       }
28     }
29   }
30   before selectDrive {
31     driveSelected = true;
32   }
33 }
34
```

Umple (cont'd)

- MOTL trace directives



Umple

```
1  class CarTransmission {
2    trace driveSelected;
3    trace neutral;
4    trace selectReverse;
5  }
```

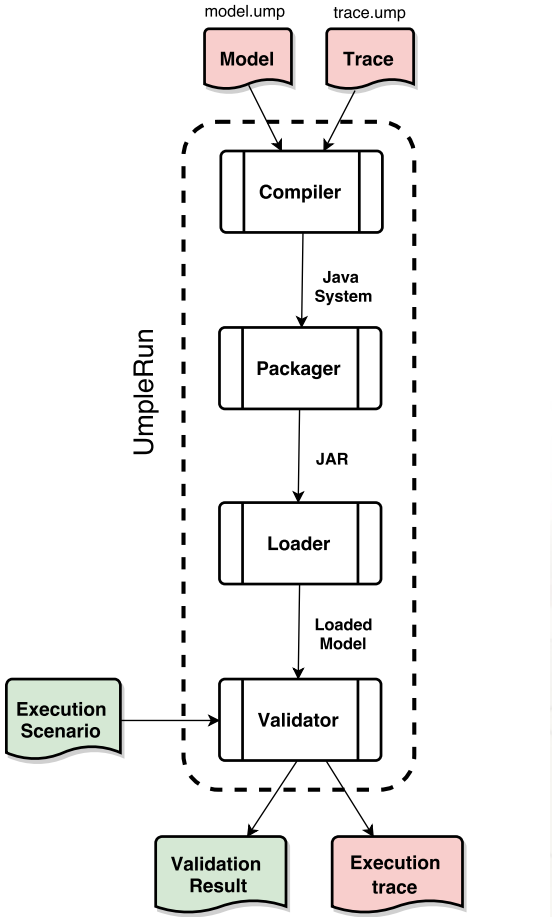
UmpleRun

- Our tool for running a set of execution scenarios against a targeted model.

- UmpleRun interprets and executes the commands in an execution scenario to produce a model validation verdict, including the failed assertions.

```
command, method_calls_after_commands ...
command_1, values_from_method_calls ...
command_2, values_from_method_calls ...
...
command_n, values_from_method_calls ...
```

UmpleRun (cont'd)



1. **Compilation**: an Umple model is parsed, analyzed and a Java system is created.
2. **Packaging**: The Java classes are then packaged into a container (JAR).
3. **Loading** the model into memory: allow creating new instances of the classes.
4. **Running**: The commands in the execution scenario are run against the class instances and the assertions are validated. The validation verdict is produced at this final stage.

uOttawa

Car Transmission Dynamic Analysis

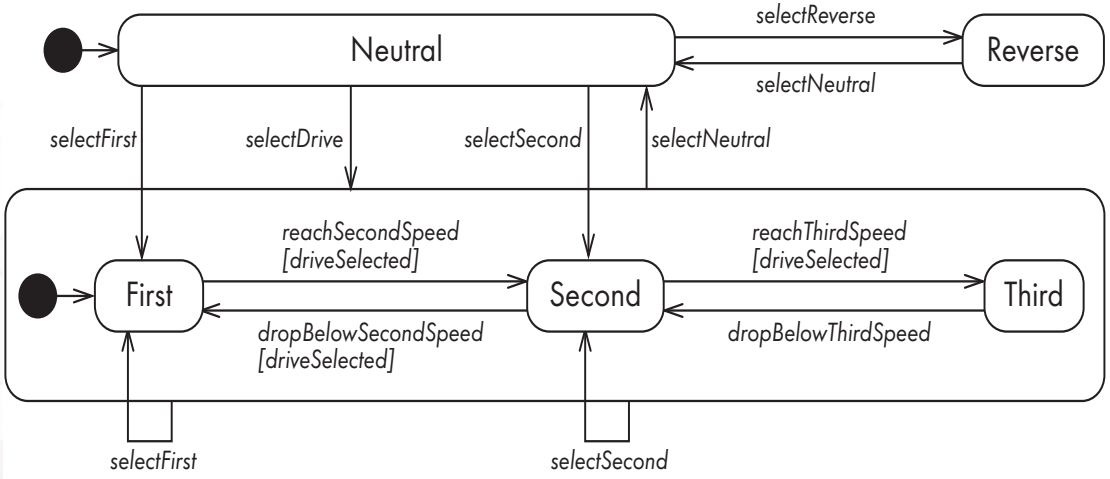
- Two execution scenarios to verify the behavior of the Car transmission state machine and explore successful validation cases of model dynamic behavior.

- Introduce a bug in the Car transmission state machine and study the validation verdict and inject trace directives to produce execution traces from UmpleRun.

Successful validation case

- Execution scenario

```
1 | command,getStatus,getStatusDrive,getDriveSelected
2 | new CarTransmission, neutral, Null, false
3 | selectReverse, reverse, Null, false
4 | selectNeutral, neutral, Null, false
5 | selectDrive, drive, first, true
6 | reachSecondSpeed, drive, second, true
7 | reachThirdSpeed, drive, third, true
8 | selectNeutral, neutral, Null, false
```

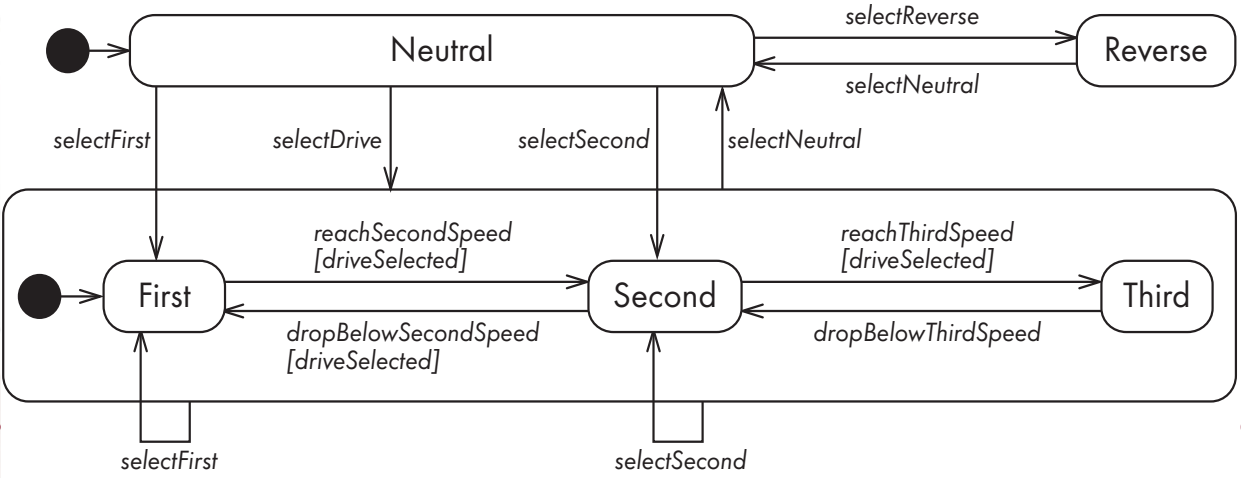



```
Compiling CarTrans.ump... success.
Building model... success.
Loading model into memory... success.
Running commands:
  Created CarTransmission
  getStatus = neutral
  getStatusDrive = Null
  getDriveSelected = false
  Executed #selectReverse
  getStatus = reverse
  getStatusDrive = Null
  getDriveSelected = false
  Executed #selectNeutral
  getStatus = neutral
  getStatusDrive = Null
  getDriveSelected = false
  Executed #selectDrive
  getStatus = drive
  getStatusDrive = first
  getDriveSelected = true
  Executed #reachSecondSpeed
  getStatus = drive
  getStatusDrive = second
  getDriveSelected = true
  Executed #reachThirdSpeed
  getStatus = drive
  getStatusDrive = third
  getDriveSelected = true
  Executed #selectNeutral
  getStatus = neutral
  getStatusDrive = Null
  getDriveSelected = false
Done.
```

uOttawa

Failed validation case

- Model defect
  - Removing code injection for the setting of Boolean attribute 'driveSelected'.
  - Thus, making guarded events non triggerable.

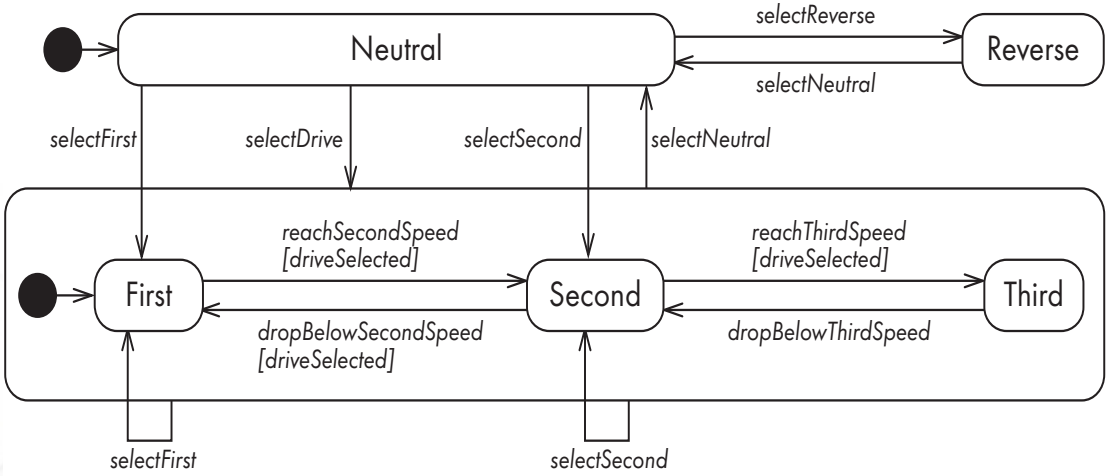



```
Compiling CarTrans.ump... success.
Building model... success.
Loading model into memory... success.
Running commands:
  Created CarTransmission
  getStatus = neutral
  getStatusDrive = Null
  getDriveSelected = false
  Executed #selectReverse
  getStatus = reverse
  getStatusDrive = Null
  getDriveSelected = false
  Executed #selectNeutral
  getStatus = neutral
  getStatusDrive = Null
  getDriveSelected = false
  Executed #selectDrive
  getStatus = drive
  getStatusDrive = first
  !!! ASSERTION FAILED on getDriveSelected,
EXPECTED true, ACTUAL false
  Executed #reachSecondSpeed
  getStatus = drive
  !!! ASSERTION FAILED on getStatusDrive,
EXPECTED second, ACTUAL first
  !!! ASSERTION FAILED on getDriveSelected,
EXPECTED true, ACTUAL false
  Executed #reachThirdSpeed
  getStatus = drive
  !!! ASSERTION FAILED on getStatusDrive,
EXPECTED third, ACTUAL first
  !!! ASSERTION FAILED on getDriveSelected,
EXPECTED true, ACTUAL false
  Executed #selectNeutral
  getStatus = neutral
  getStatusDrive = Null
  getDriveSelected = false
Done.
```

uOttawa

Failed validation case (cont'd)

- Trace Directive
  - obtain execution trace.



Umple

```
1  class CarTransmission {
2    trace drive record driveSelected;
3  }
```

```
1  Time,Thread,UmpleFile,LineNumber,Class,Object,Operation,Name,Value
2  *,1,CarTrans.ump,6,CarTransmission,874088044,sm_t,neutral,selectDrive,drive,false
3  *,1,CarTrans.ump,6,CarTransmission,874088044,sm_t,drive,selectNeutral,neutral,false
```

Future

- Automatically generating a comprehensive set of execution scenarios

- Full model execution including associations.

Try Umple

try.umple.org

Questions ?

uOttawa

Command line execution

```
java -jar umplerun.jar model.ump exeScenrio.cmd
```

Information sources

Umple open source website

- http://code.umple.org

Continuous integration and quality assurance pages

- http://cc.umple.org http://qa.umple.org

Umple user manual – MOTL pages

- http://motl.umple.org

Downloading

- http://dl.umple.org