

PINT: achieving faster and more realistic tests at Scania

- Daniel Sundmark, Avenir Kobetski, Mats Skoglund and Jakob Axelsson
SICS Swedish ICT
- Thomas Gustafsson and Niclas Clashammar
Scania CV AB



SCANIA

Introduction

- Scania and SICS collaborate in the project Parallel INtegration Testing (PINT) funded by Vinnova
 - Full-vehicle integration testing
 - Can parallel execution of scenario-based test cases make integration testing more efficient and effective?
 - Originates in a Scania funded prestudy

Background

- Scania uses one of the largest and most complex Hardware-In-the-Loop (HIL) systems in the world for:
 - Integration testing
 - Regression testing of user functions
 - Note: the simulator must support Scania's module system, i.e., millions of possible vehicle configurations



Background

- Automation part of testing with HIL:
 - Scenario-based and scripted testing
 - Tool-chain selects set of valid test scripts
 - Test automation framework executes valid tests and reports results
 - Sequential execution of the tests
 - Each test is a sequence of steps

Objectives of PINT

- Better real-world representativeness
 - Currently, the testscripts execute one at a time, but
 - User functions are distributed functions
 - User functions are activated concurrently
 - Left direction indicator, and High beam, and Starting from driver area
- Reduced time for testing
- Increased defect detection

Challenges

- Scenario-based and scripted testing means
 - User function 1
 - Act 1
 - Stimuli 1
 - Stimuli 2
 - ...
 - Stimuli n
 - Check expected response
 - Act 2
 - ...

Challenges

- Parallel execution is a prerequisite for the objectives, but is a challenge

TC1

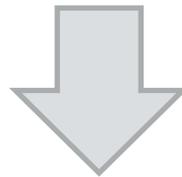
Act1

Drive at speed 85 km/h
Check expected response

TC2

Act1

Press brake pedal 50%
Check expected response



Parallel execution

Drive at speed 85 km/h
Press brake pedal to 50%
Check expected response tc1
Check expected response tc2

Challenges

- How to detect conflicting states and conflicting stimuli leading to conflicting states
- In the prestudy, some mutually excluded states were modeled using automata
- Supremica was used to find possible interleavings of two scripts
- Modeling domain knowledge



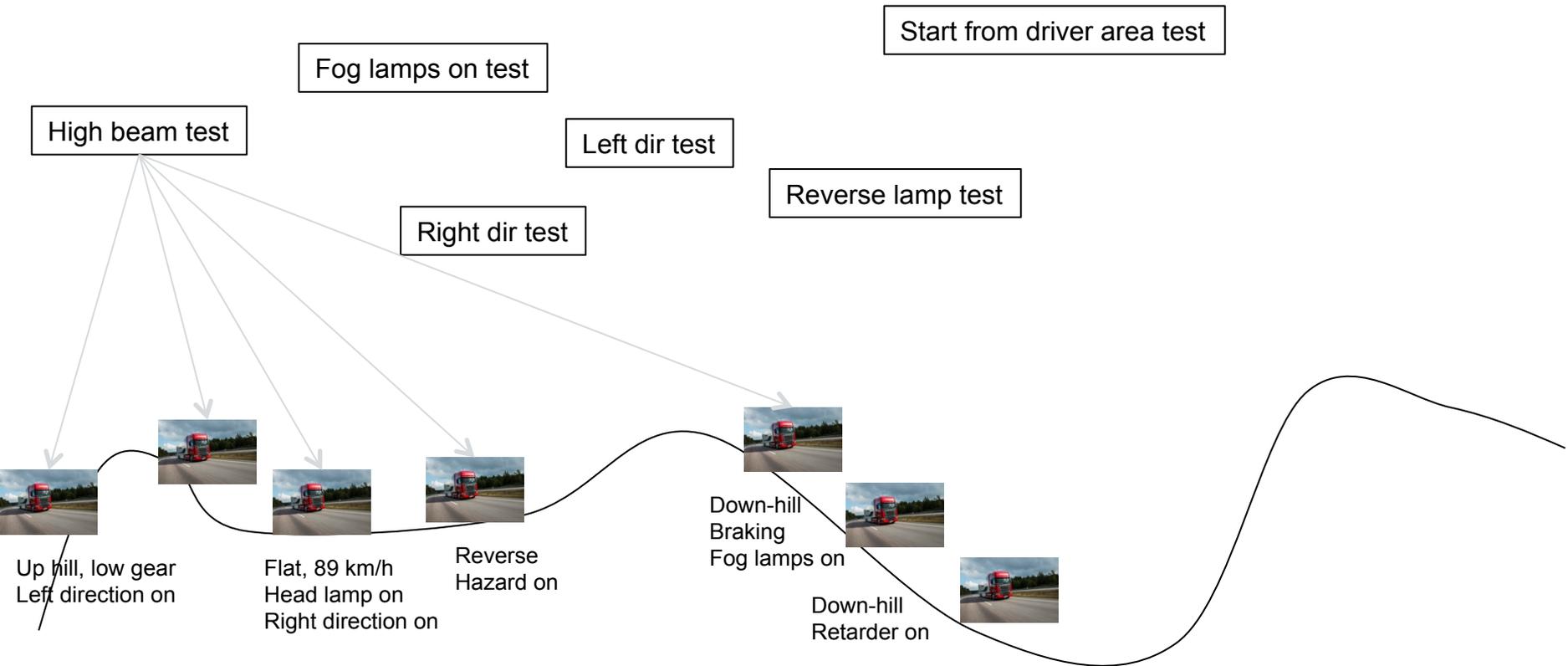
Declarative testing

Approach

- Declarative testing
 - Make test scripts read-only
 - Act 1
 - Loop
 - Wait for state, e.g., driving at 85 km/h
 - check expected response
- Read-only test scripts can execute in parallel without conflicting
- Construct a course setting the vehicle in different states

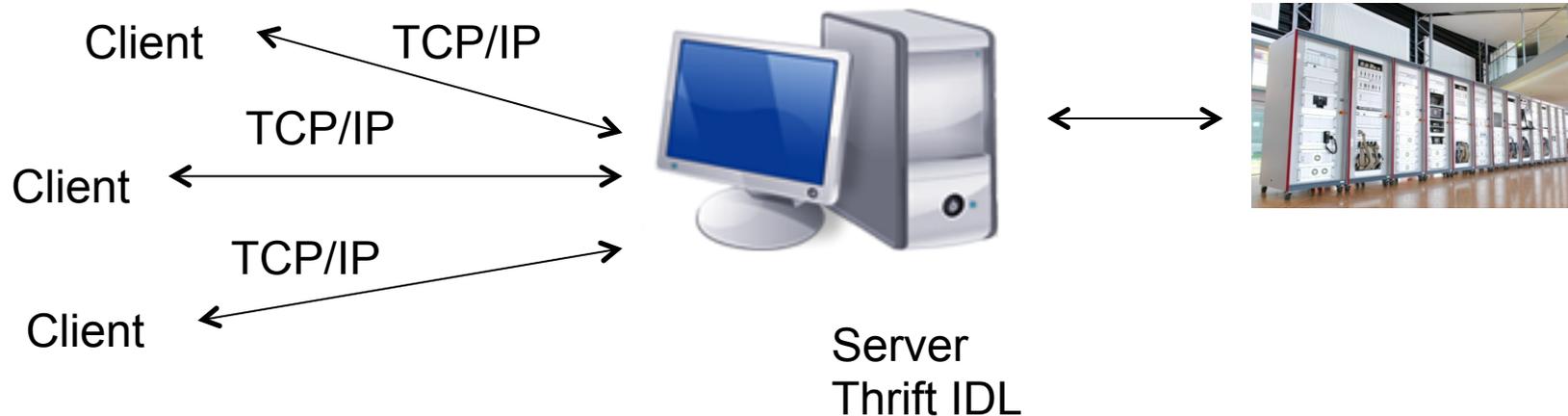
Approach

- Constructed course, example



Approach

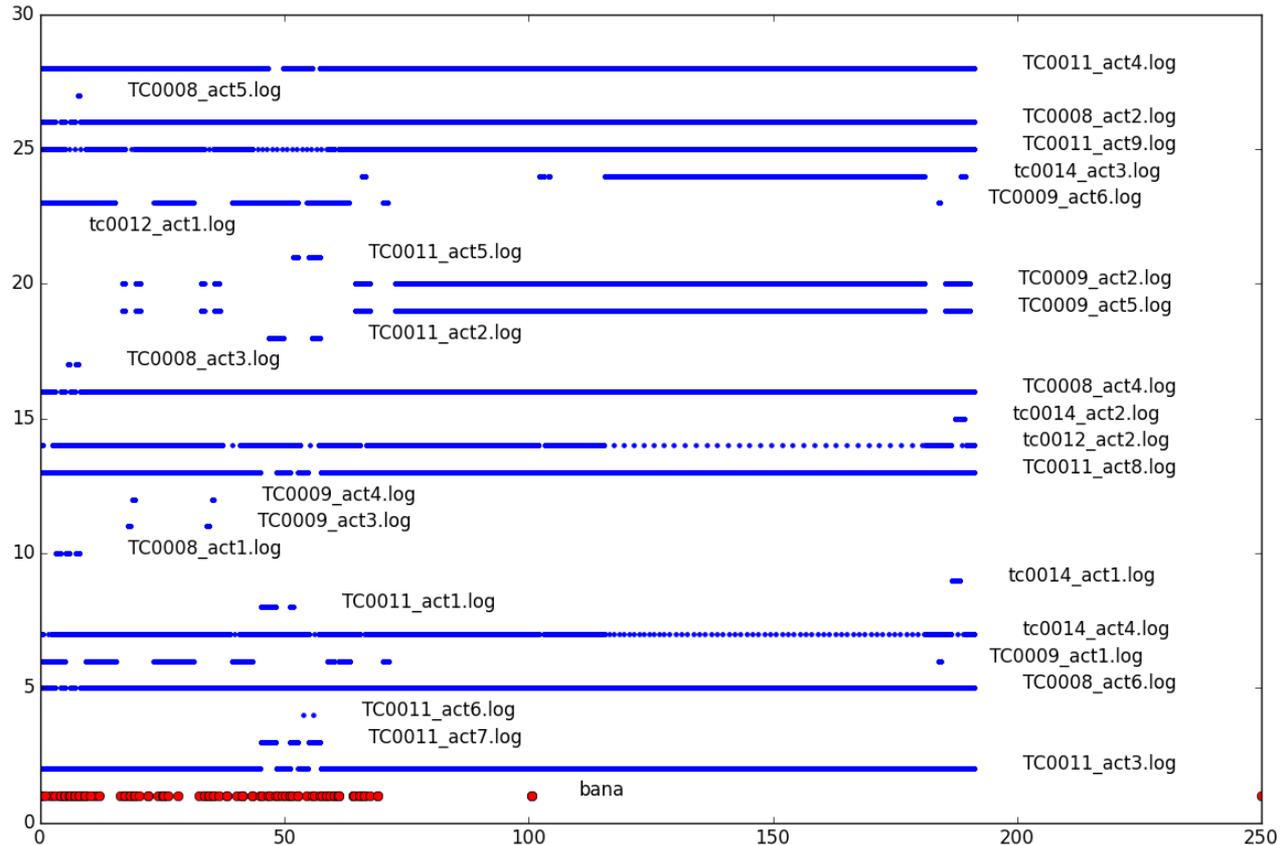
- Scania's test automation framework uses client/server architecture



Preliminary results

- Several test scripts have been converted to read-only format
- The scripts' stimuli have been moved to a track script
- One act in each new script; 27 concurrent scripts
- When an evaluation has been done, sleep for 10 ms
- Plot
 - Each time a script performs a check against expected response
 - Each time the track performs a stimuli
- Note: the converted scripts have not been reviewed, nor evaluated against the old scripts

Preliminary results



Conclusions

- Declarative testing could be the answer to the objectives
 - Read-only scripts performing evaluations
 - One script changes state of SUT; denoted the track
 - Focusing on testing, and not how to reach a state
- Declarative testing needs a new test automation framework supporting concurrency

Conclusions

- Early observations
 - Potential for reducing testing time
 - Scripts are evaluated concurrently
 - One part of a track can run multiple scripts in parallel
 - Potential for better real-world representativeness
 - Several distributed functions are active and tested concurrently
 - Potential for improved defect detection
 - Many more evaluations are performed
 - Evaluations are done in settings where the old script would not have executed

Future work

- Investigate how the technical solution scales
- Automatic construction of tracks
- Optimization of tracks
- Expressiveness in testscript implementation language

Preliminary results

Act	Evaluated to	# True	# False
TC008_act1	True	253	0
TC008_act2	False	43	18709
TC008_act3	True	74	0
TC008_act4	False	43	18709
TC008_act5	True	33	0
TC008_act6	False	43	18709
TC009_act1	True	2896	0
TC009_act2	True	11931	0
TC009_act3	True	78	0
TC009_act4	True	78	0
TC009_act5	True	11923	11
TC009_act6	True	2891	1646
TC0012_act1	True	0	0
TC0012_act2	True	10882	49
TC0014_act1	True	171	0
TC0014_act2	True	93	82
TC0014_act3	True	6810	0
TC0014_act4	True	10876	415

Preliminary results

- Promising results
- Several scripts are evaluating concurrently
- The scripts evaluate many more times than one time