Time Partition Testing

Testing from Model-in-the-Loop to Hardware-in-the-Loop using one tool

PikeTec GmbH

andreas.kraemer@piketec.com

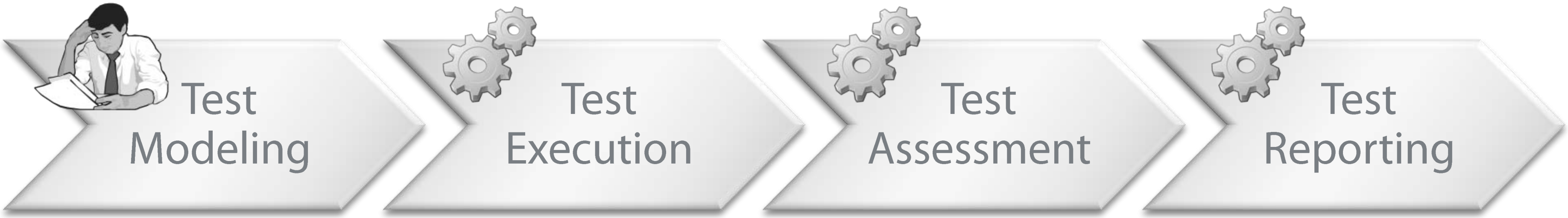
Why yet another testing tool? – Why using TPT?

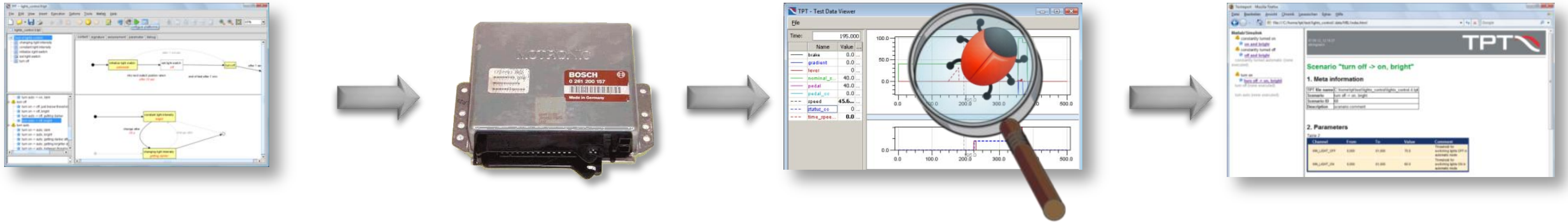
- **Effective and efficient** testing can be achieved by:
  - Test automation
  - Test frontloading
  - Test reuse
  - Tailored test models
  - Lean, consistent test processes

TPT
Time Partition Testing

TPT Test Automation

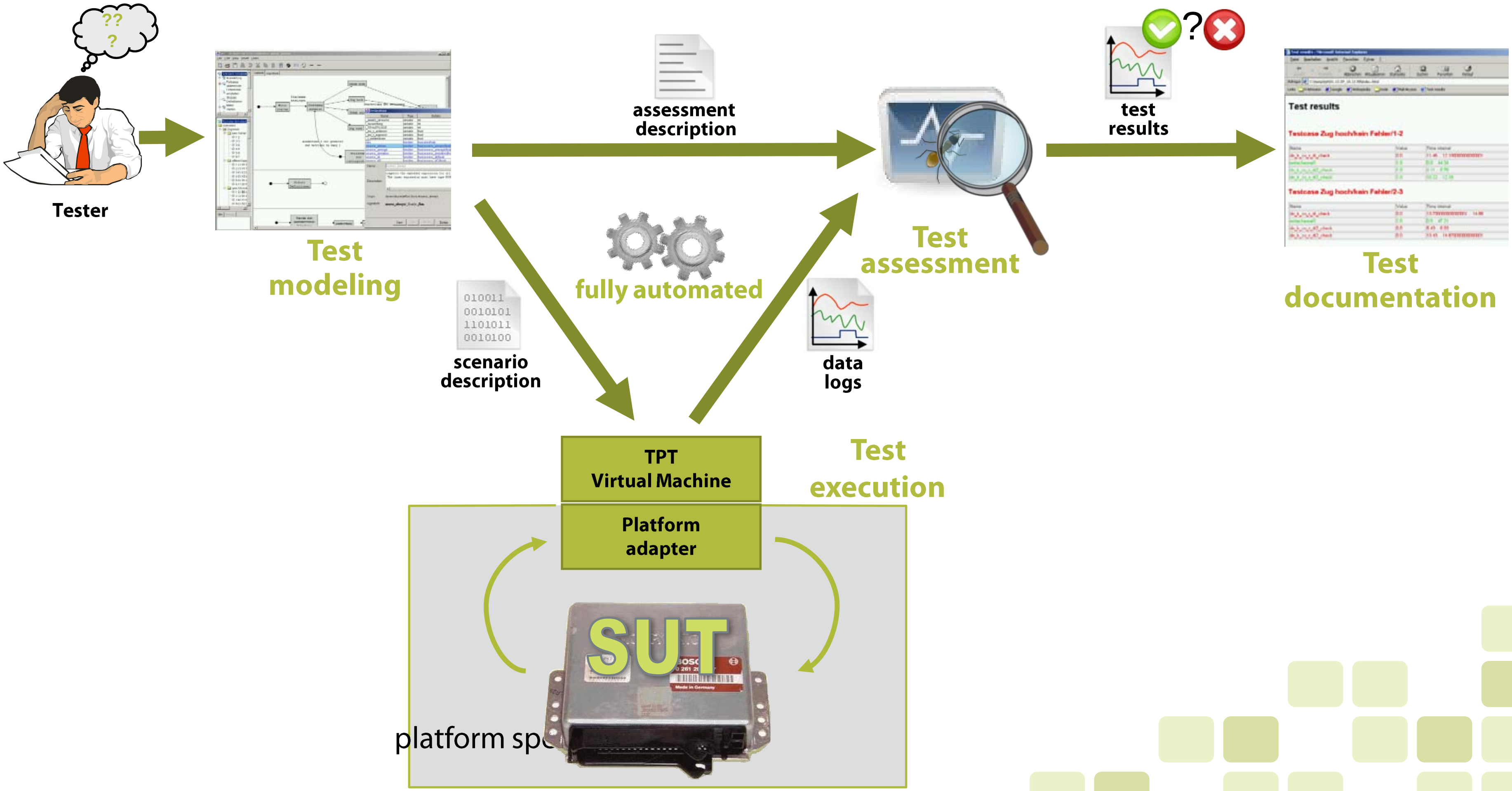
- TPT automates all steps from execution to reporting





Testing with TPT

1 Test modeling 2 Test execution 3 Assessment + Reporting



??
?

Tester

Test modeling

assessment description

fully automated

scenario description

010011
0010101
1101011
0010100

Test assessment

data logs

test results

?

Test results

Testcase Zug hoch/kein Fehler/1-2

Testcase Zug hoch/kein Fehler/2-3

Test documentation

TPT
Virtual Machine

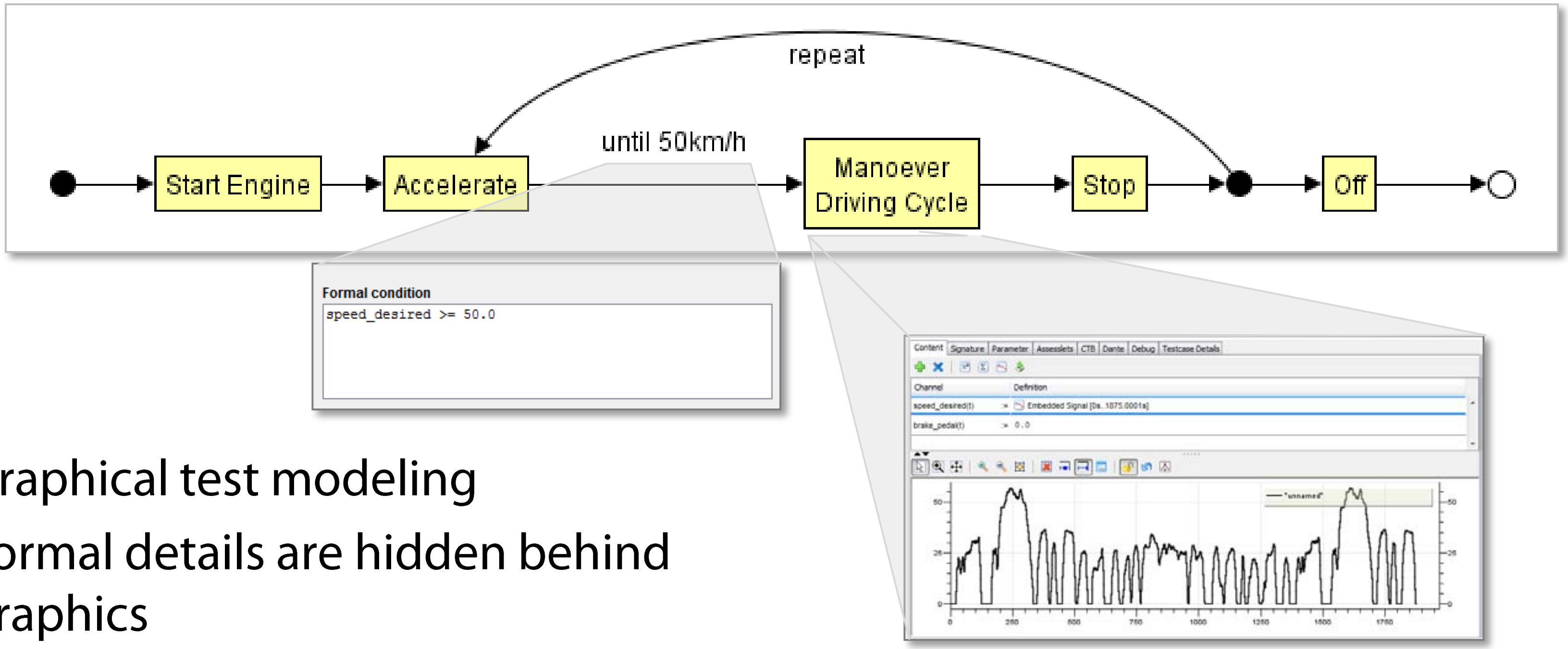
Platform adapter

Test execution

SUT

platform spe

Tailored test models



Formal condition

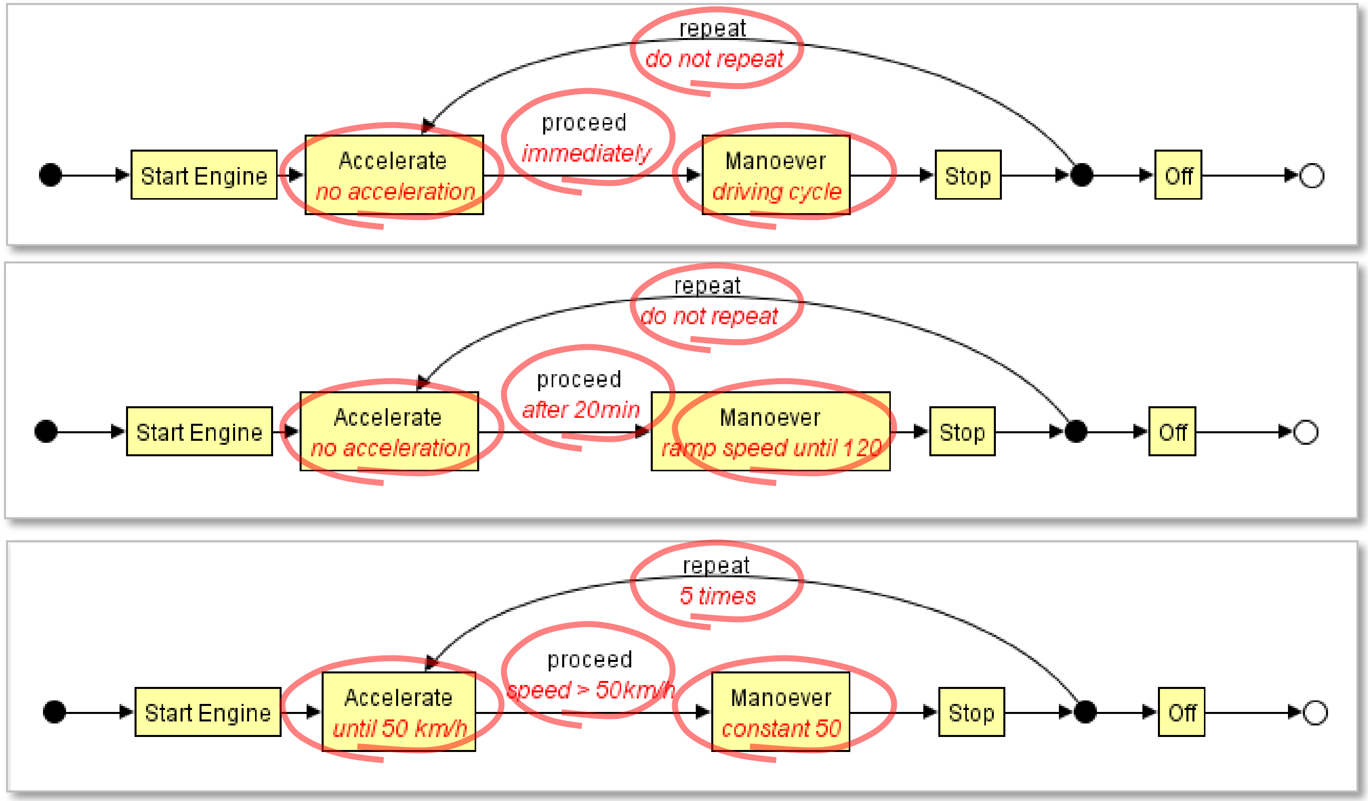
```
speed_desired >= 50.0
```

Channel	Definition
speed_desired(t)	Embedded Signal [0s..1875.0001s]
brake_pedal(t)	0.0

- Graphical test modeling
- Formal details are hidden behind graphics
- Real-time enabled
- Closed loop (reactive) testing
- Clear structured and easy to learn
- Compact (low complexity)

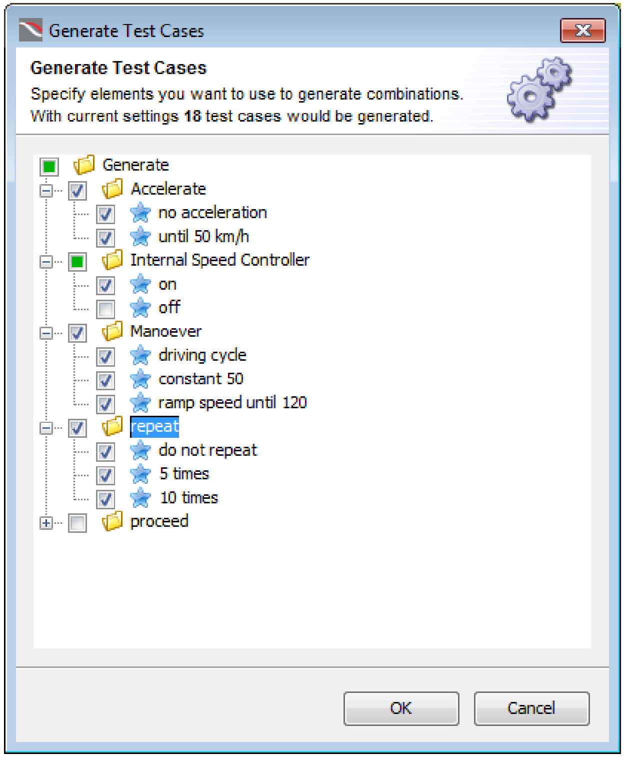
Variant modeling in TPT

- Generate individual tests by combination of variants



Test case generation in TPT

- Test cases can be generated automatically

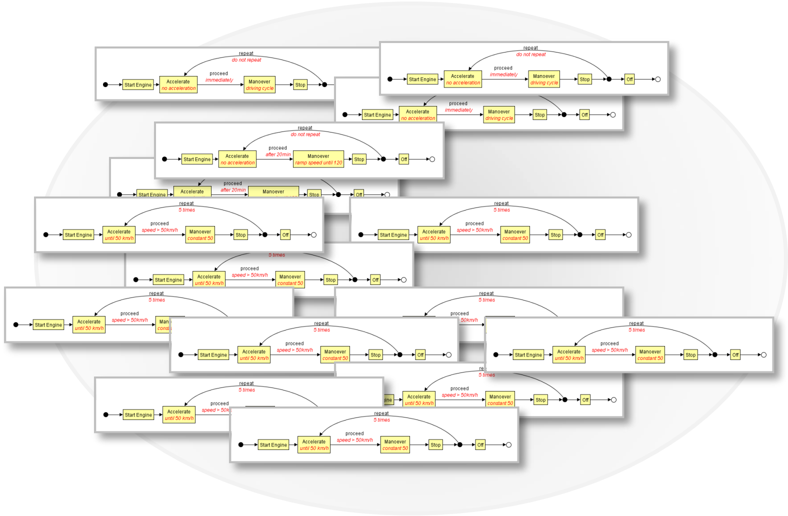


Generate Test Cases

Generate Test Cases

Specify elements you want to use to generate combinations.
With current settings **18** test cases would be generated.

- 🟩 Generate
  - ☑ Accelerate
    - ☑ no acceleration
    - ☑ until 50 km/h
  - 🟩 Internal Speed Controller
    - ☑ on
    - □ off
  - ☑ Manoever
    - ☑ driving cycle
    - ☑ constant 50
    - ☑ ramp speed until 120
  - ☑ repeat
    - ☑ do not repeat
    - ☑ 5 times
    - ☑ 10 times
  - □ proceed

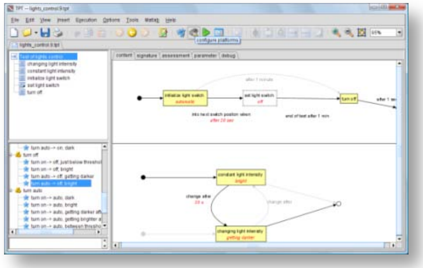


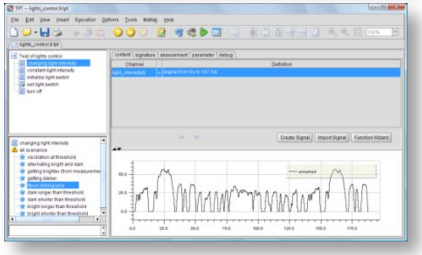
TPT

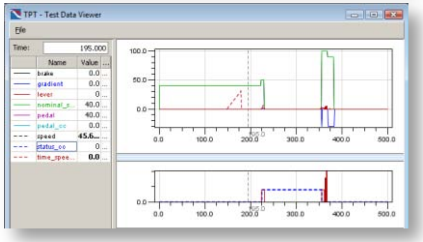
TPT

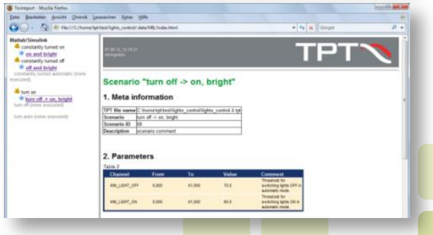
Time Partition Testing

TPT is a test tool for testing control and feedback control systems


TPT – lights_control 8.tpt
File Edit View Insert Execution Options Tools Matlab Help
content signature assessment parameter debug
initialize light switch
set light switch
turn off
constant light intensity
changing light intensity


TPT – lights_control 8.tpt
content signature assessment parameter debug
Channel
Definition
Create Signal
Import Signal
Function Wizard



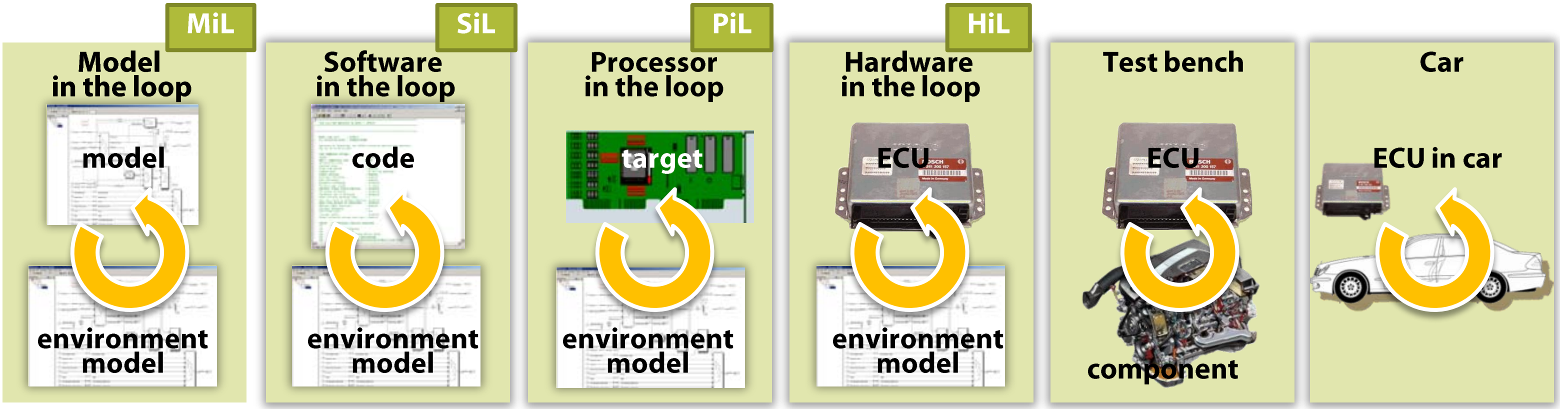


Test challenges

- Clear test case description (Modeling)
- Continuity and consistency at all test platforms
- Automated Execution, Assessment, Reporting
- Real-time behavior
- Coverage and tracing of Requirements (ISO 26262)

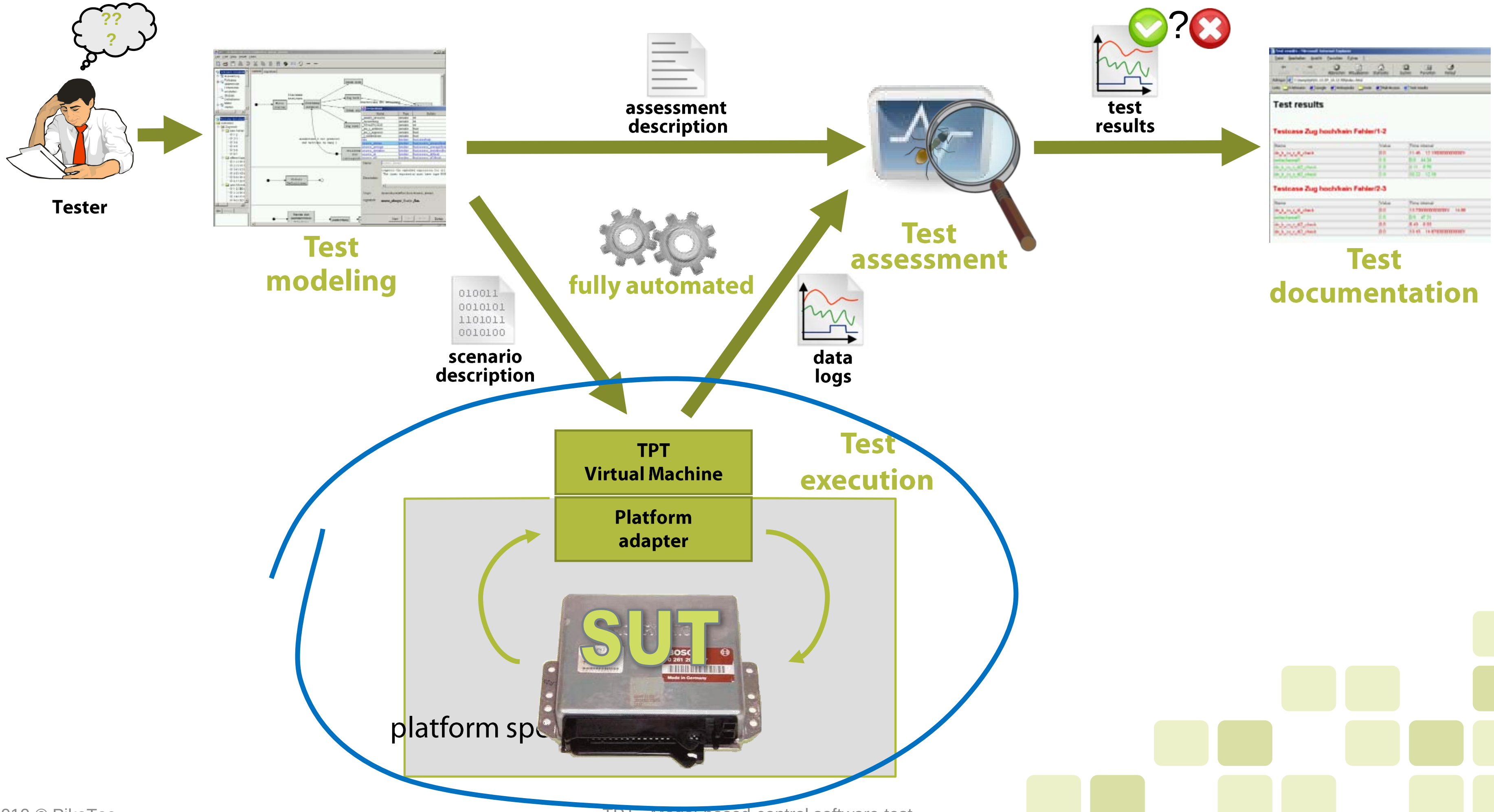
TPT test execution

Reactive real time test execution everywhere



- Using the same test cases on different platforms
  - MATLAB/Simulink
  - C-Code
  - HiL
  - Vehicle e.g. via CAN
- Real-time enabled
- Reactive tests

TPT Test Process

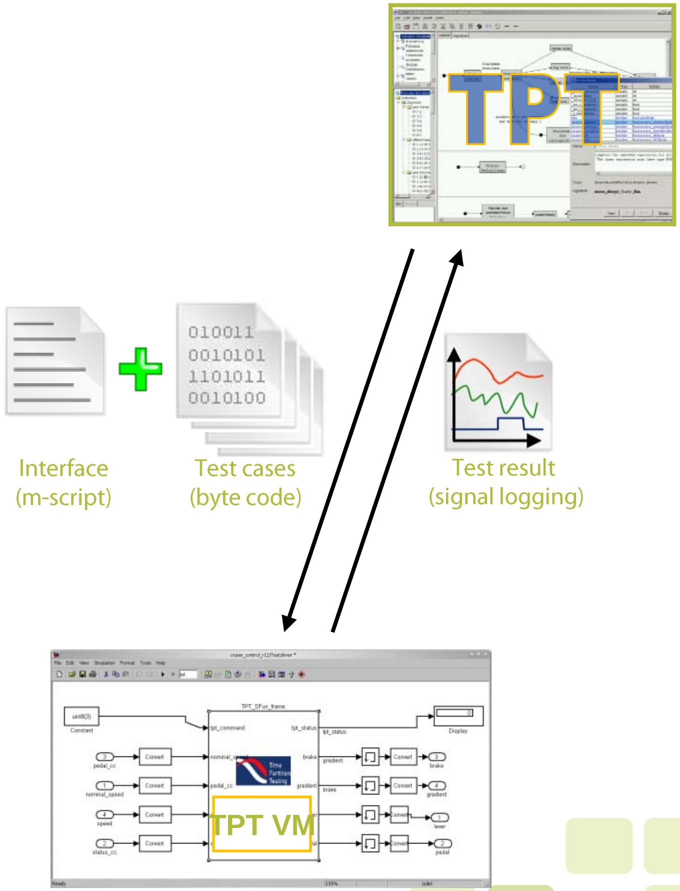


Test execution

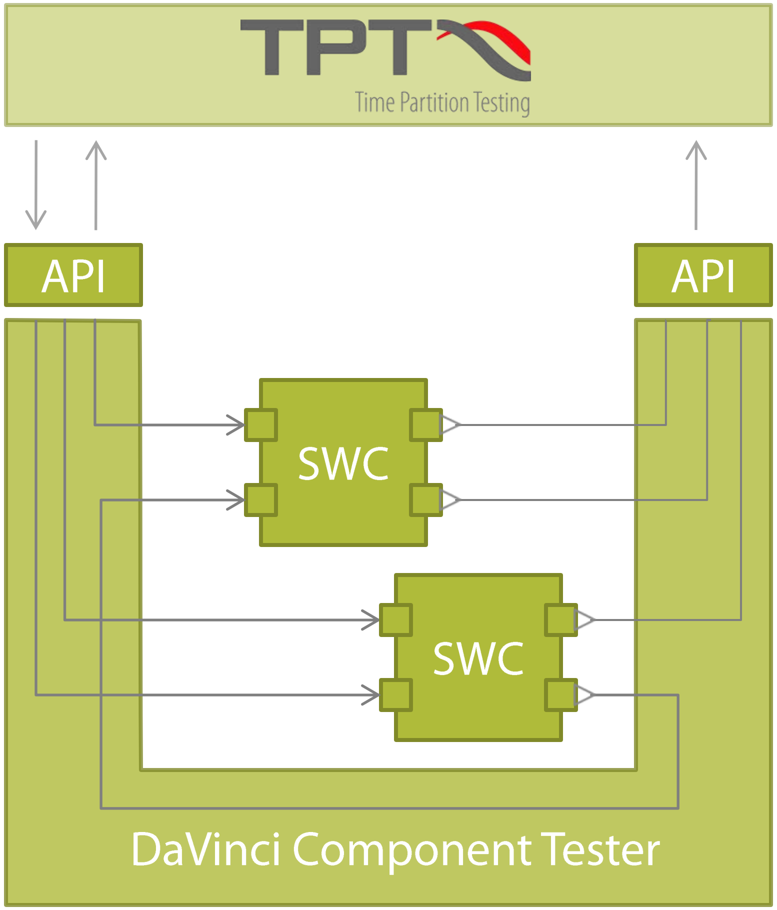
- Based on TPT-VM integrated in a test platform
- TPT-VM is real-time enabled
- Currently supported test platforms include:
  - MIL
    - MATLAB/Simulink/Stateflow/Targetlink
  - SiL
    - C-Code
    - Tessy
    - Customer specific SiL-Environments
  - HIL
    - ProveTECH:TA (Engineering Solution)
    - MESSINA
    - CameLView
    - MLBA4
    - Customer specific HiL-Environments
- Integration in new platforms requires between 5 days to 3 month of work (depending on integration requirements and openness of the platform to integrate)

Example: Matlab/Simulink environment

- Automated test execution
- TPT VM is embedded into a Simulink S-function
- Communication with ECU code and environment (car model) is managed by Simulink
- Interface is specified by a generated m-script
- Test cases are hand over by means of workspace variables
- RTW and Targetlink enabled

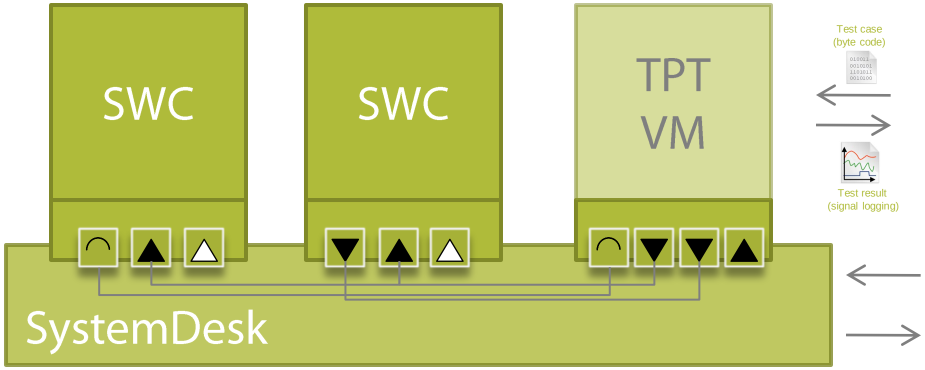


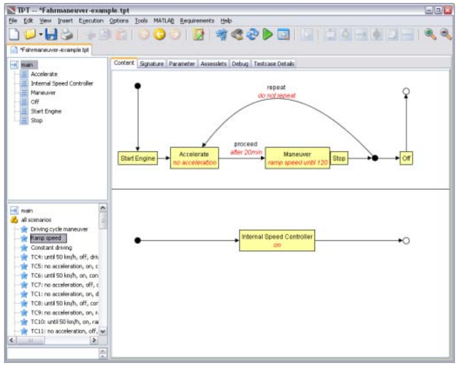
DaVinci AUTOSAR testing



SystemDesk AUTOSAR testing

- TPT VM is embedded as AUTOSAR SW-C at the VFB



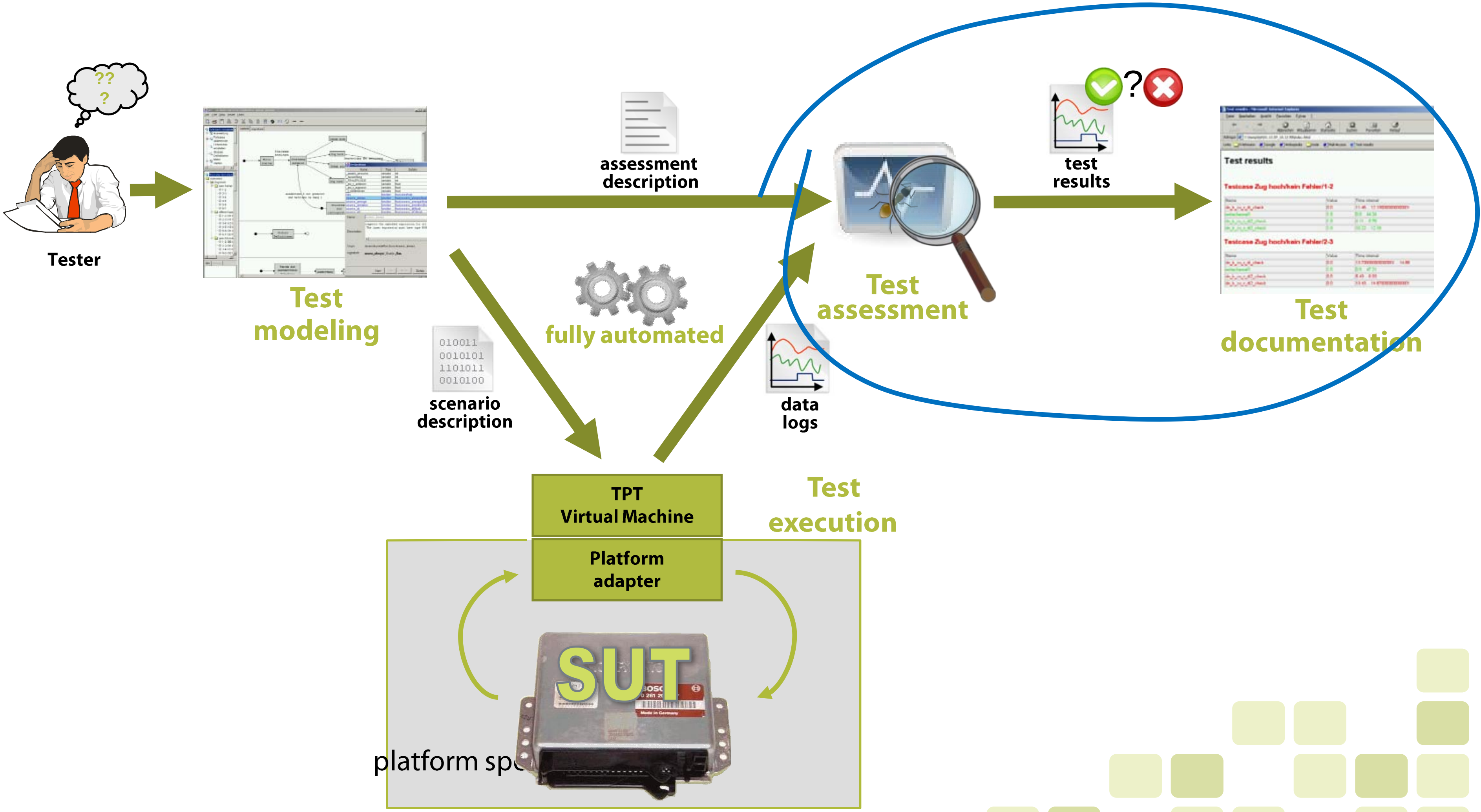


- main
  - Accelerate
  - Internal Speed Controller
  - Maneuver
  - Off
  - Start Engine
  - Stop

- main
- all scenarios
  - Driving cycle maneuver
  - Ramp speed
  - Constant driving
  - TC4: until 50 km/h, off, dri
  - TC5: no acceleration, on, c
  - TC6: until 50 km/h, on, con
  - TC7: no acceleration, off, c
  - TC1: no acceleration, on, d
  - TC8: until 50 km/h, off, cor
  - TC9: no acceleration, on, r
  - TC10: until 50 km/h, on, ra
  - TC11: no acceleration, off,

TPT Test-evaluation

TPT Test Process



Tester

Test modeling

assessment description

scenario description

fully automated

TPT Virtual Machine

Platform adapter

SUT

platform spe

Test execution

data logs

Test assessment

test results

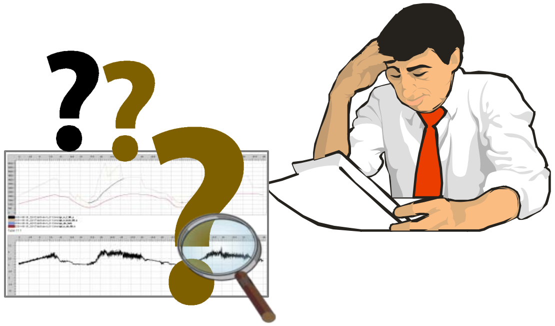
Test documentation

??
?
?
Test results
Testcase Zug hoch/kein Fehler/1-2
Testcase Zug hoch/kein Fehler/2-3
010011
0010101
1101011
0010100

Evaluation of test results

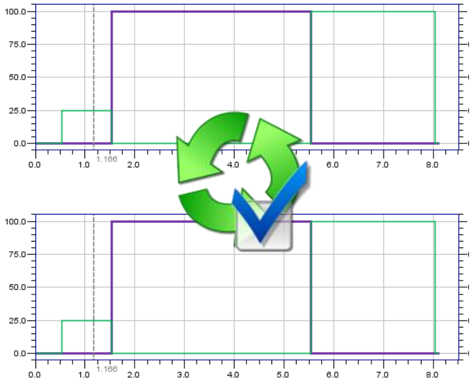
Manual evaluation

- Simple evaluation method
- Good for non-recurring tests
- Always an expert must do the evaluation



Regression test

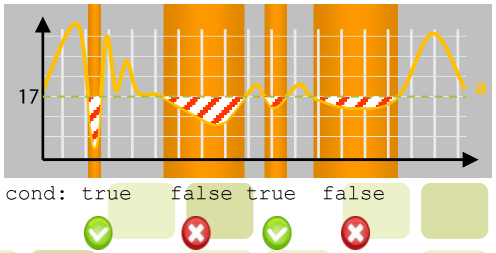
- Inital effort similar to manual evaluation
- Sensitive to parameter changes, type series and functional changes



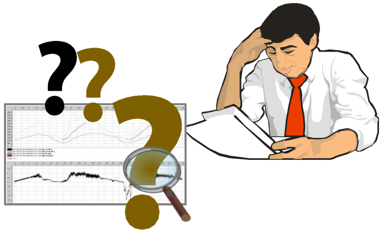
TPT assessments

- Initial effort because of programming the assessments
- Simple maintanance with less effort
- Cost reduction for recurring tests

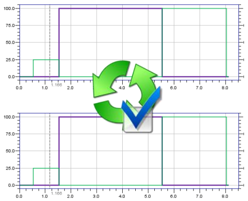
```
during TPT.regexp([a(t)<17]):
    cond := this.getLength()<10.0;
```



Possible options to evaluate test results

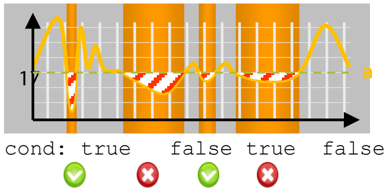


- Manual evaluation



- Regression tests (back-to-Back)

- Online evaluation



- Offline evaluation

TPT

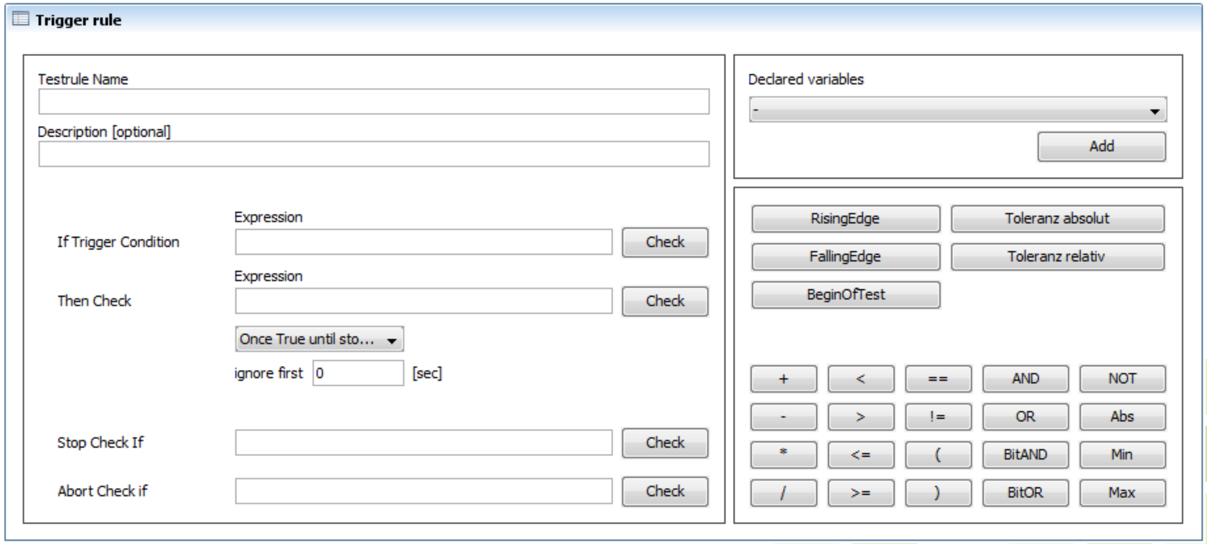
Time Partition Testing

Test assessment

- Manual analysis
  - Using the test data viewer
- Back-to-Back Analysis
  - One feature of offline assessments in TPT
- Online assessments (integrated in test models)
  - Runtime decisions
  - Abort on error
- Offline assessments
  - Minimize online efforts
  - Reference comparison
  - Access to measurements (MCD3)
  - Huge library, access to file system, external tools etc.
  - Building blocks are called **Assesslets**

Graphical test result evaluation methods - Assesslets

- GUI for commonly used test assessment methods
- MIN/MAX comparison
- Signal comparison
- Sequence check
- Trigger rules
- Script
  (Extended Python)

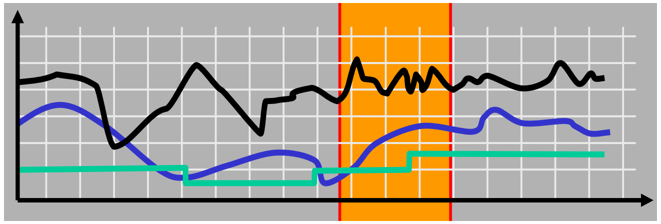


Once True until sto...

ignore first 0 [sec]

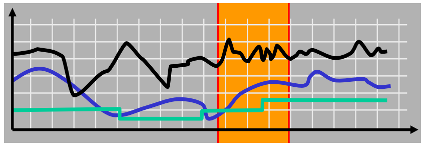
Time intervals

- In most practical cases properties to analyze focus on particular, characteristic time intervals (subsets of the overall test run).
- Special case: the test run itself



- How can such time intervals be characterized in assessments?

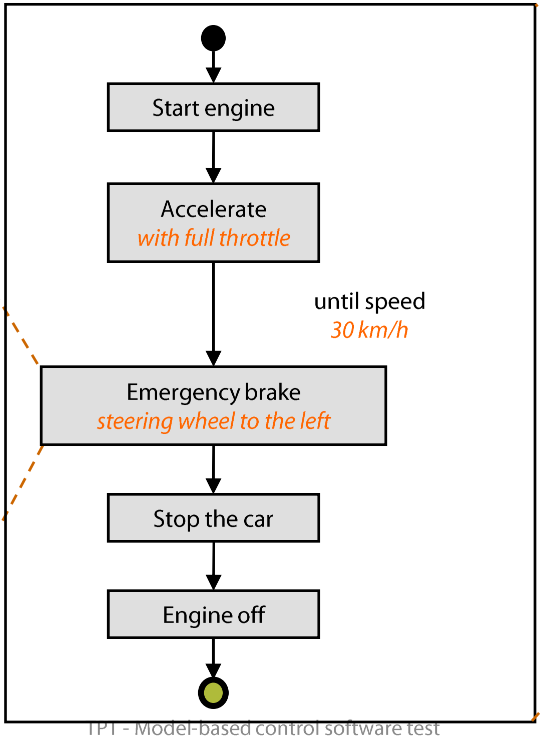
Implicit time intervals



- Each state can be used as a time interval.
- Properties that are assigned to a state A are checked in every time interval where the state A was active.

Steering wheel to the left

- Duration of the emergency brake may not exceed 10sec.
- Deceleration must be less than 1.5 m/s2.
- Revs per minute must be less than 6000 min-1.

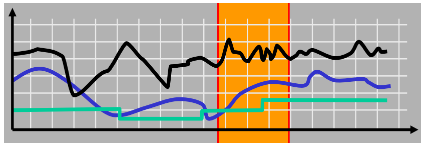


Full throttle to 30 km/h

- Failure recognition module may not detect any failure in this test case.
- The complete test run may not last longer than 30 sec.

Explicit time intervals

- Explicit time intervals can be specified to characterize more complex time interval using so called time patterns.
- Time patterns are special cases of temporal regular expressions.



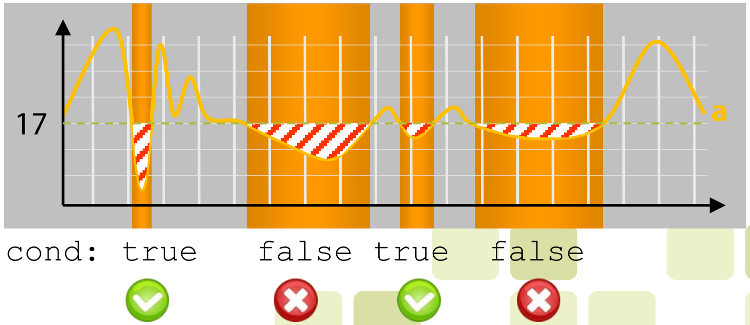
- Examples:

[v_vehicle(t) >= 100.0]	*time intervals with speed* $\geq$ *100*
[foo(t) == 1] [foo(t) == 2]	*time intervals with foo=1 followed by foo=2*
([foo(t) == 1] [foo(t) == 2])+	*time intervals with foo=1 and foo=2 alternating*

Implicit time intervals

- Common analyses are
  - Checking of signal bounds
  - Comparison with reference signals
  - Correctness of value sequence (for discrete signals)
  - Duration of particular test phases
- Analysis is always performed in context of a well-defined time interval
- Every assessment can be analyzed in multiple time intervals:

```
during TPT.regexp([a(t)<17]):
   cond := this.getLength()<10.0;
```



Feature examples

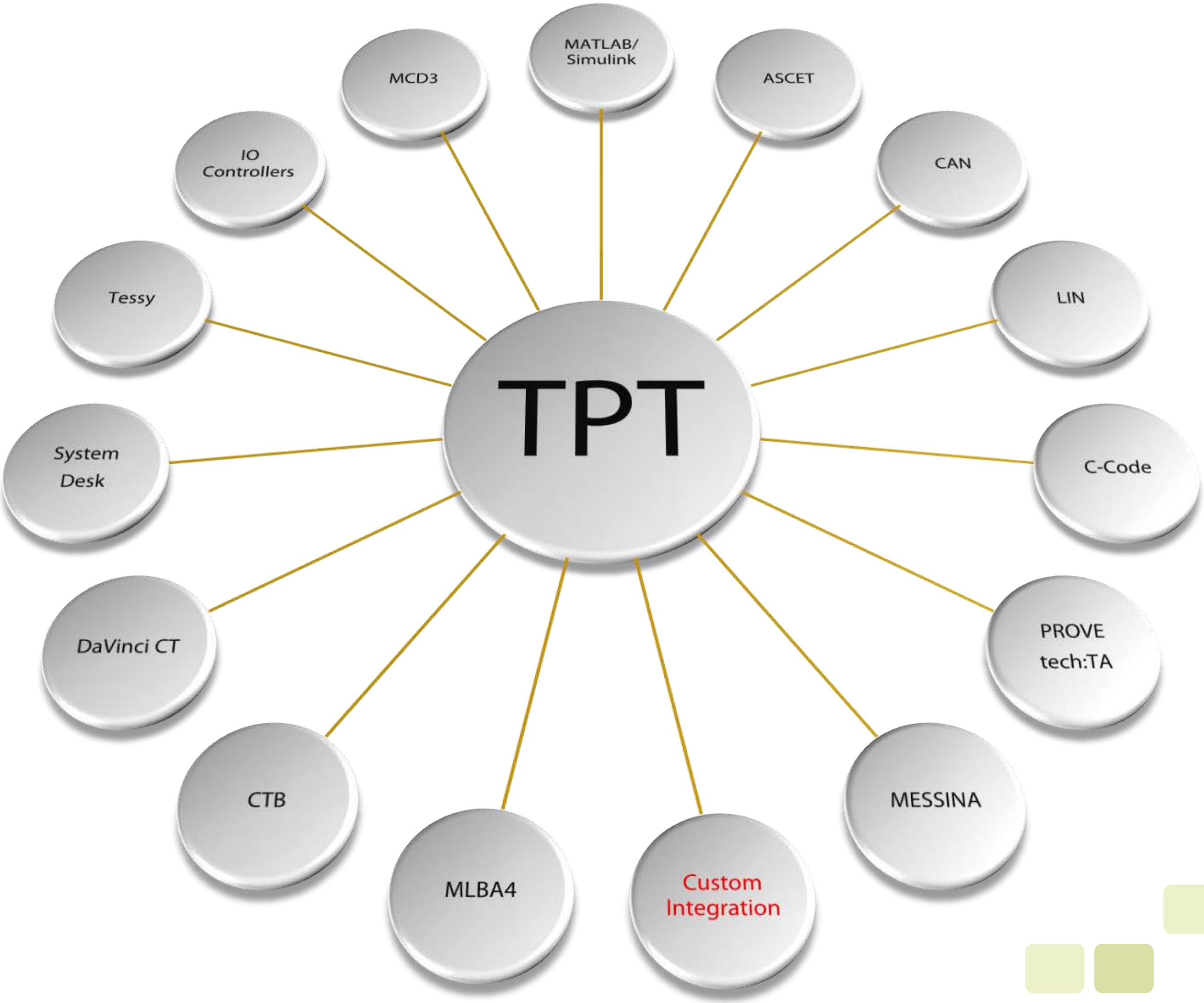
- Monotony checks
- Duration checks
- Sequence checks
- Always/Exists Checks
- Signal Filter (FIR, IIR, MA)
- Bounds checks
- Signal comparison
- Signal File Import/Export

Features and Platforms

TPT Features (Overview)

- **Modeling**
  - Graphical test models
  - Closed-loop (reactive) tests
  - Hard real-time enabled (≤100µs cycles)
  - Signal import
  - Signal editor
  - Wizard based signal creation
  - Variant handling / one model for all tests
  - Test case generation (combinatory)
- **Parameter Support**
  - Scalars, arrays, curves, and maps
  - Parameter import
  - Parameter overloading/calibration
  - Online parameter calibration
- **Assessment**
  - Online and Offline
  - Back-to-back Analysis
  - Temporal conditions
  - General constraints and analysis per scenario
  - Flexible offline concept with scripting language
- **Execution**
  - Support of many execution platforms
  - Multiple test sets and execution configurations
  - Debugger for analysis
- **Reporting**
  - HTML or MHTML
  - Highly configurable
  - Template editor for report customization
  - Additional programmable content
- **Requirements Management**
  - Import/Synchronization of requirements
  - Import/Export/Synchronization of test cases
  - Import/Export/Synchronization of links
  - Impact analysis when requirements changes
  - DOORS Integration
- **MCD3**
  - Measurements for offline assessment
  - Calibration before or during the test execution
- **Test Data Viewer and Editor**
  - Interactive and easy handling
  - Multiple viewers
  - One or two Y-axes per viewer
- **File I/O**
  - MDF, DBC, LDF, A2L, DCM, ARXML, HDR, CSV, TPTBIN

Supported Platforms by TPT



TPT Feature Summary

Time Partition Testing

Platform independent test models

Consistency from model to assessment and report

Automated tests (from test execution to test report)

Closed loop tests supported

Abstract test language

Systematic test case definition

Intuitive graphical models

Continuous behavior testing

Requirements tracing (e.g. Doors)

ASAM MCD 3 measuring