4th UCAAT User Conference on Advanced Automated Testing

ETSI World Class Standards

TEST CASES TO FIND THE BEST ARCHITECTURE IN TERMS OF PERFORMANCE

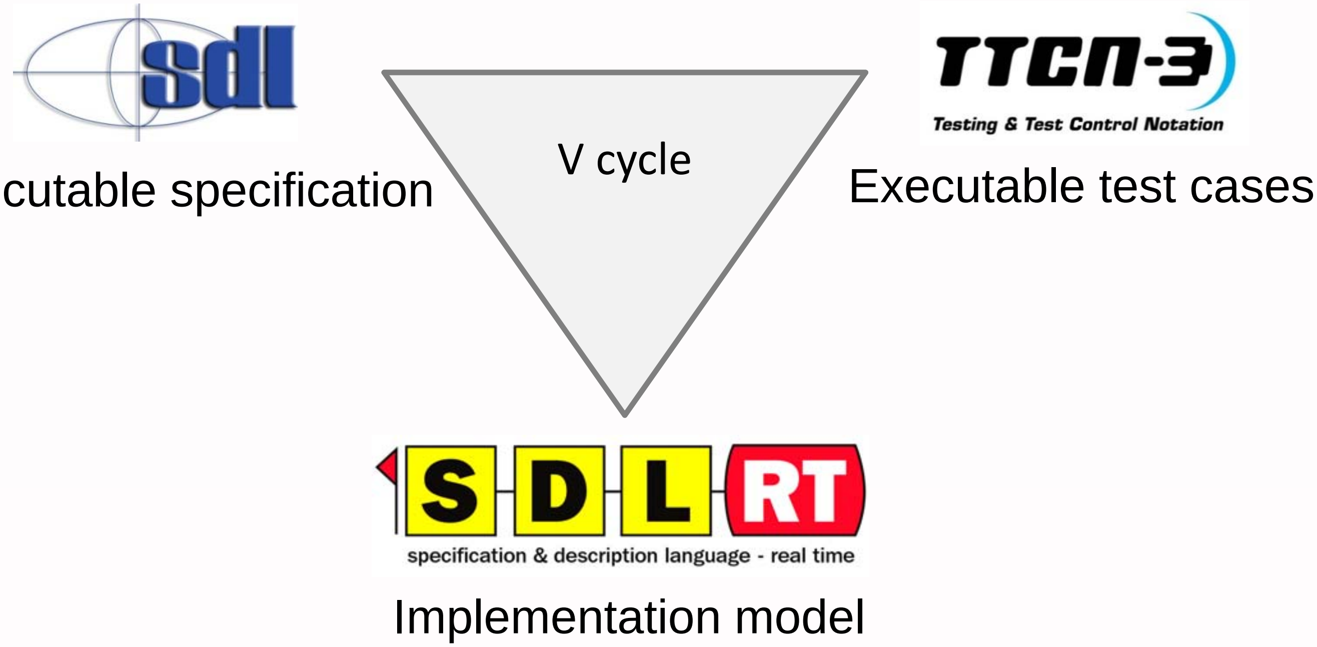
mihal.brumbulli@pragmadev.com

emmanuel.gaudin@pragmadev.com

PRAGMADEV modeling and testing tools

PragmaDev

- French software editor based in Paris
- Provides modeling and testing tools for event driven applications covering the whole lifecycle


sdl
TTCN-3
Testing & Test Control Notation
V cycle
SDL RT
specification & description language - real time

Executable specification

Executable test cases

Implementation model

Why optimizing performances

- Increasing complexity of systems
- More and more distributed systems
- Performance is a non functional aspect that is paramount but difficult to address

➢ It would be interesting to use a real functional model

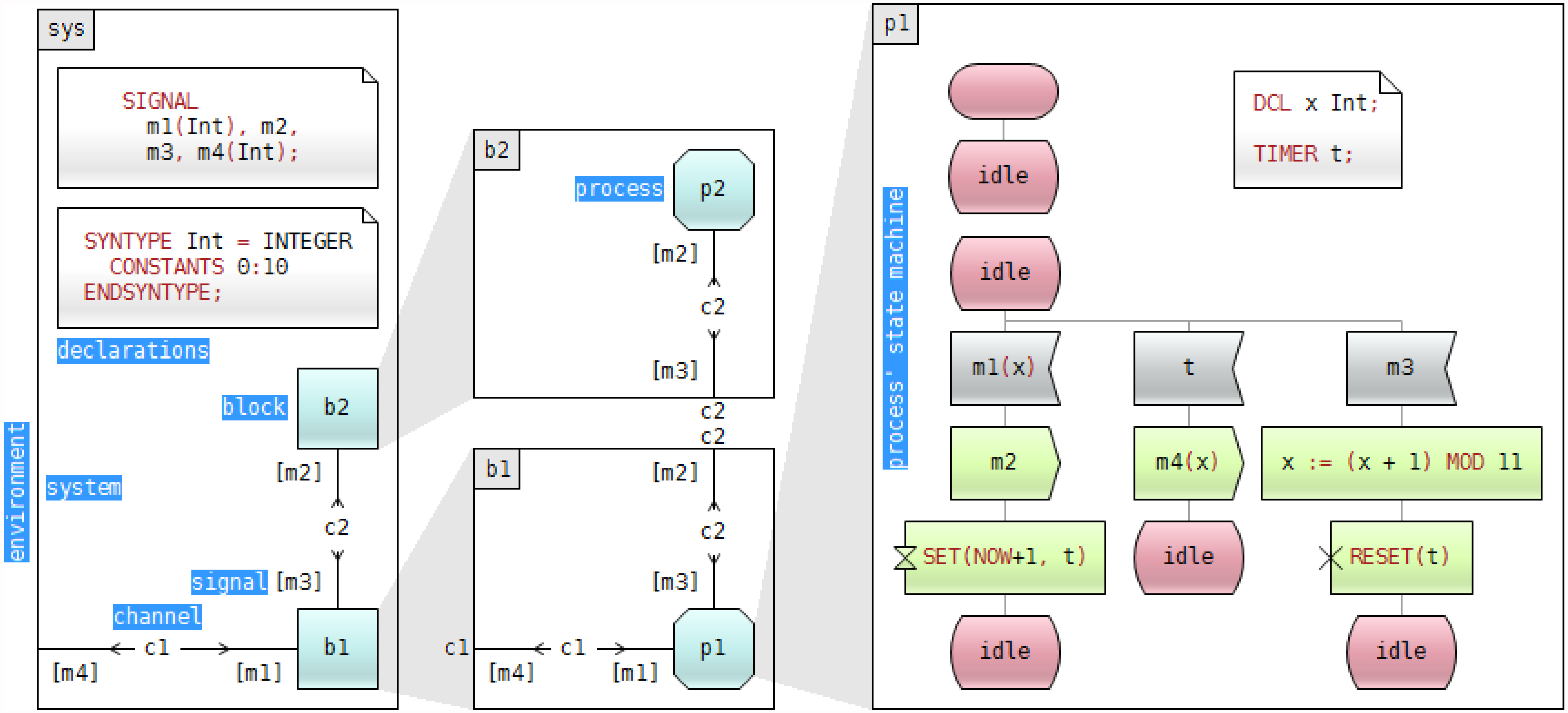
➢ It would be interesting to use real scenarios

Architecture and allocation

- SDL technology allows to design an executable model very early in the development process of a system
  - ➢ Functional requirements can be verified on the model

- SDL architecture is abstract, there is no information regarding the final allocation
- SDL execution takes no time, or an undefined time to execute
  - ➢ There is no help in the model to find the best architecture in terms of time or energy consumption

Functional model



```
SIGNAL
  m1(Int), m2,
  m3, m4(Int);
```

```
SYNTYPE Int = INTEGER
  CONSTANTS 0:10
ENDSYNTYPE;
```

```
DCL x Int;

TIMER t;
```

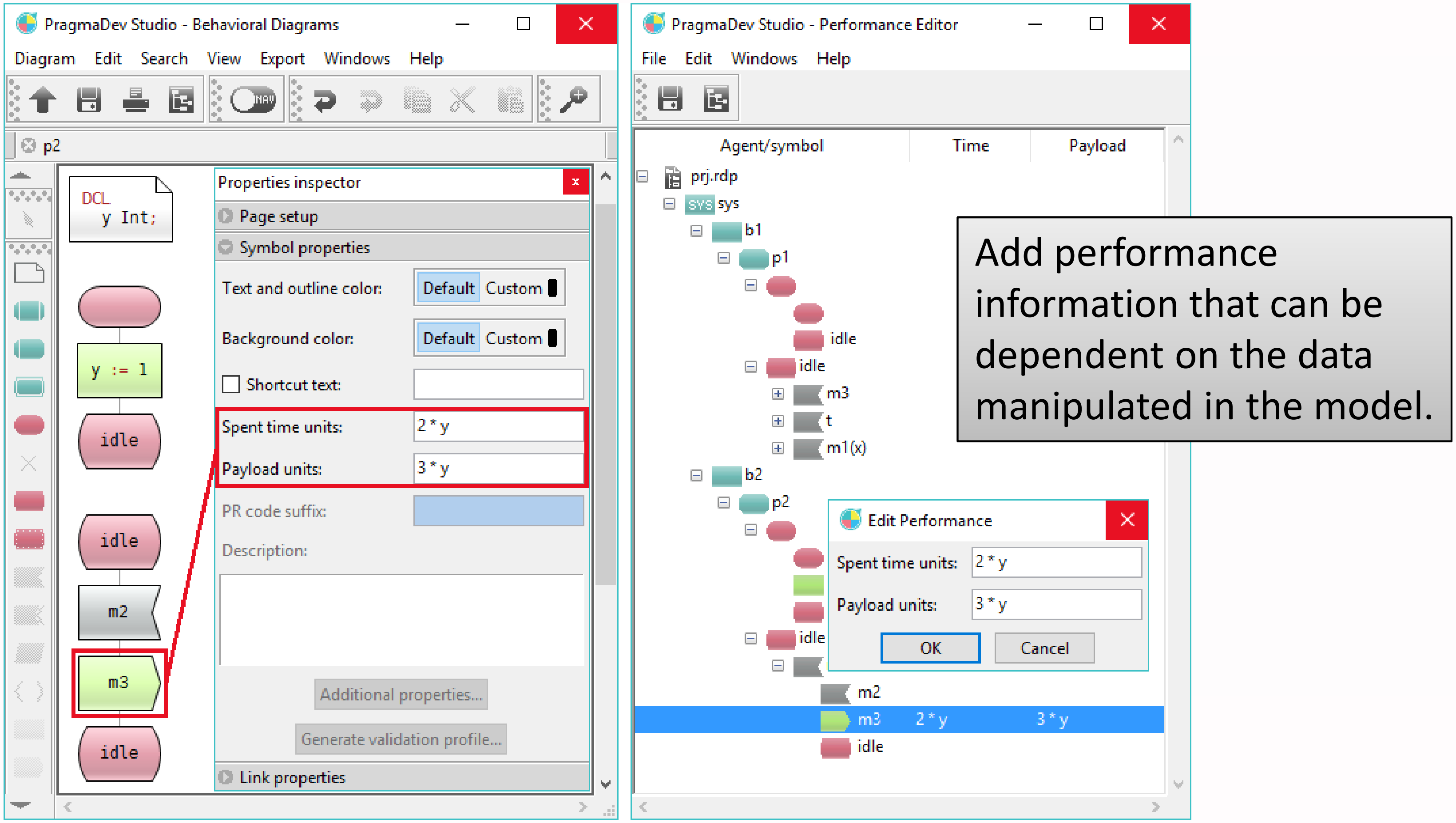
```
x := (x + 1) MOD 11
```

```
SET(NOW+1, t)
```

```
RESET(t)
```

- Architecture
- Communication
- Behavior

Adding performance information



Spent time units: 2 * y

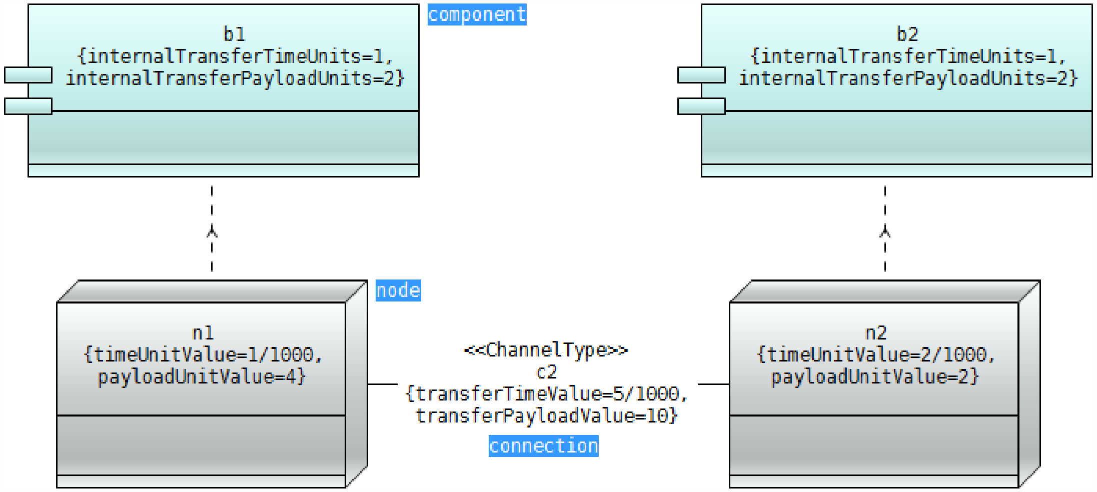
Payload units: 3 * y

Agent/symbol	Time	Payload
m3	2 * y	3 * y

Add performance information that can be dependent on the data manipulated in the model.

Allocation

- Allocation with UML deployment diagram
- Component is named after the SDL agent with 2 properties:
  - internalTransferTimeUnits
  - internalTransferPayloadUnits
- Execution node property:
  - timeUnitValue
  - payloadUnitValue
- Connection properties to replace component ones when inter node communication:
  - transferTimeValue
  - transferPayloadValue

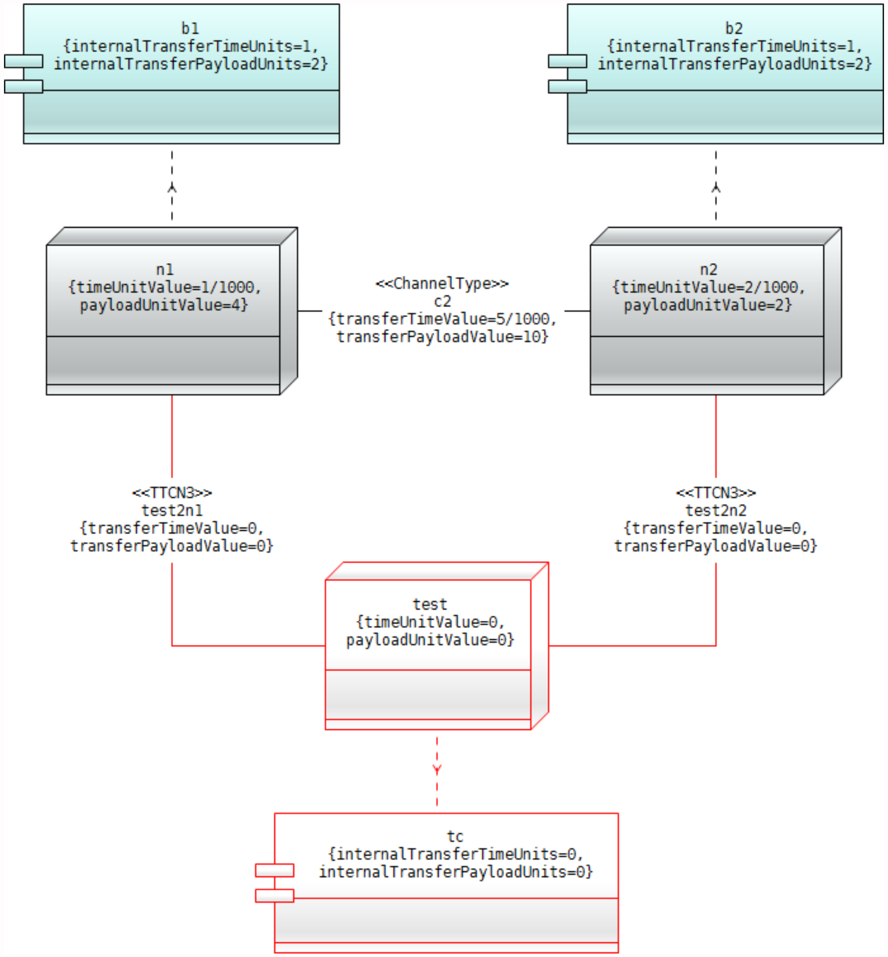


➢ Timer runs is parallel

➢ Payload adds up

Stimuli

- Use real TTCN-3 test cases
- Abstract and executable like SDL model
- Note the test case execution has no impact on the performance



b1
{internalTransferTimeUnits=1,
internalTransferPayloadUnits=2}

b2
{internalTransferTimeUnits=1,
internalTransferPayloadUnits=2}

n1
{timeUnitValue=1/1000,
payloadUnitValue=4}

<<ChannelType>>
c2
{transferTimeValue=5/1000,
transferPayloadValue=10}

n2
{timeUnitValue=2/1000,
payloadUnitValue=2}

<<TTCN3>>
test2n1
{transferTimeValue=0,
transferPayloadValue=0}

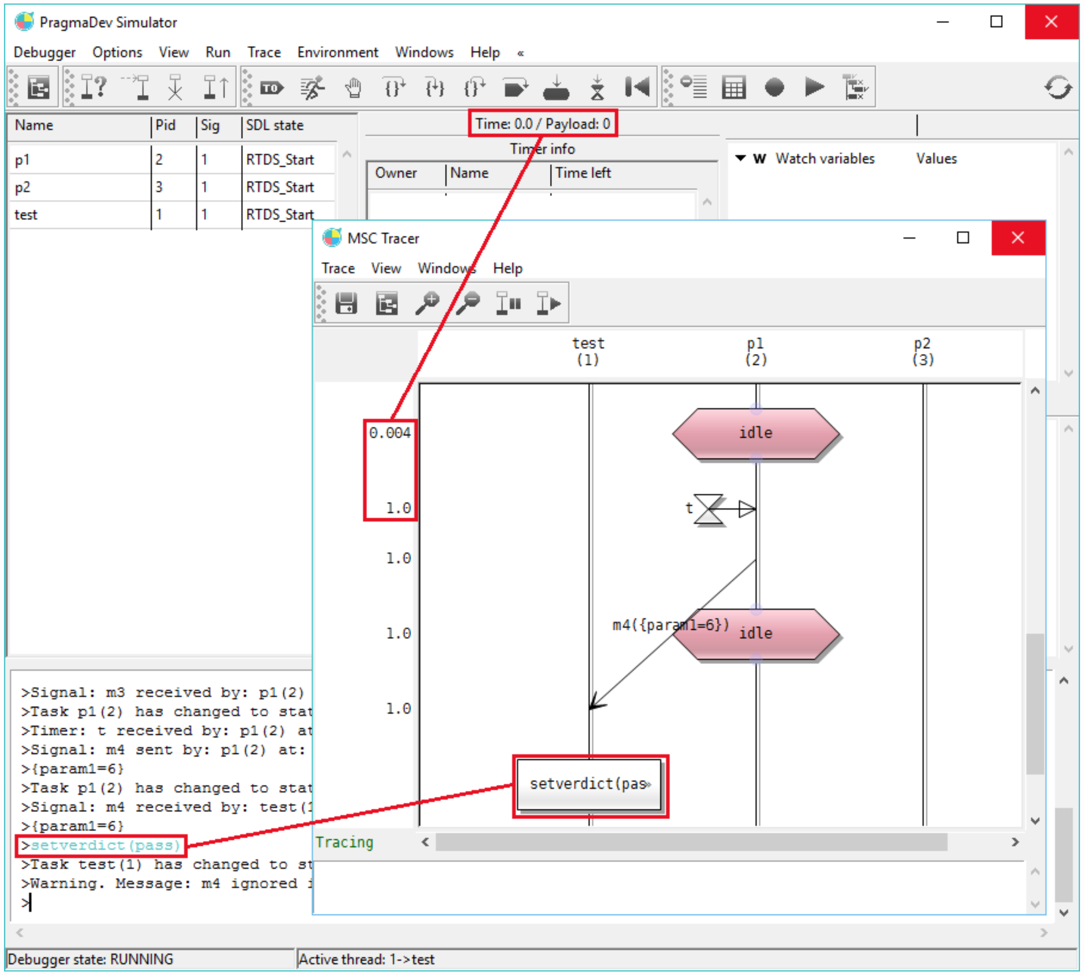
<<TTCN3>>
test2n2
{transferTimeValue=0,
transferPayloadValue=0}

test
{timeUnitValue=0,
payloadUnitValue=0}

tc
{internalTransferTimeUnits=0,
internalTransferPayloadUnits=0}

Cross simulation

Run a test case against the model including timing aspects.



Name	Pid	Sig	SDL state
p1	2	1	RTDS_Start
p2	3	1	RTDS_Start
test	1	1	RTDS_Start

```
>Signal: m3 received by: p1(2)
>Task p1(2) has changed to stat
>Timer: t received by: p1(2) at
>Signal: m4 sent by: p1(2) at:
>{param1=6}
>Task p1(2) has changed to stat
>Signal: m4 received by: test(1
>{param1=6}
>setverdict(pass)
>Task test(1) has changed to st
>Warning. Message: m4 ignored i
>
```

Debugger state: RUNNING

Active thread: 1->test

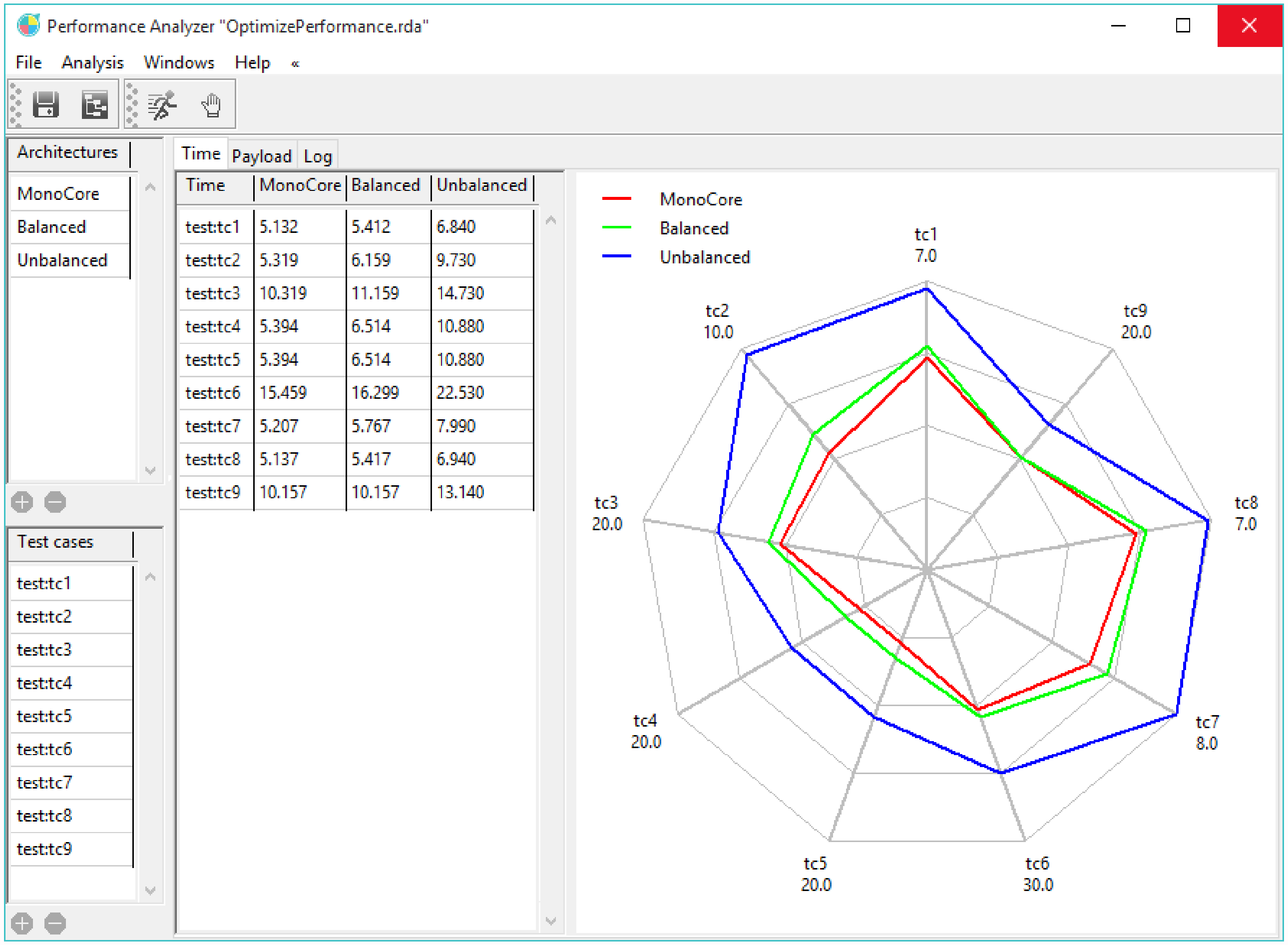
Cross architecture

Execute the model on:

- A set of **architectures**

Against:

- A set of **test cases**



Time	MonoCore	Balanced	Unbalanced
test:tc1	5.132	5.412	6.840
test:tc2	5.319	6.159	9.730
test:tc3	10.319	11.159	14.730
test:tc4	5.394	6.514	10.880
test:tc5	5.394	6.514	10.880
test:tc6	15.459	16.299	22.530
test:tc7	5.207	5.767	7.990
test:tc8	5.137	5.417	6.940
test:tc9	10.157	10.157	13.140

Test cases to find the best architecture in terms of performance

Conclusion

- Functional SDL models are unchanged
- Real TTCN-3 test cases
- Ease the trade off to find the best architecture