Probabilities – a key solution for tomorrow real-time compositional frameworks

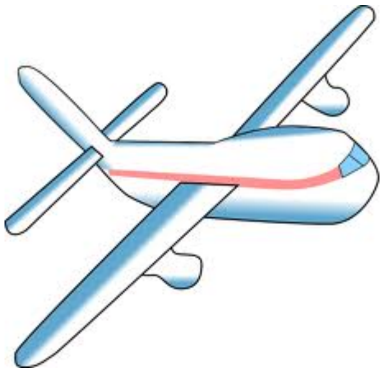
L. Cucu-Grosjean and co-authors

Outline

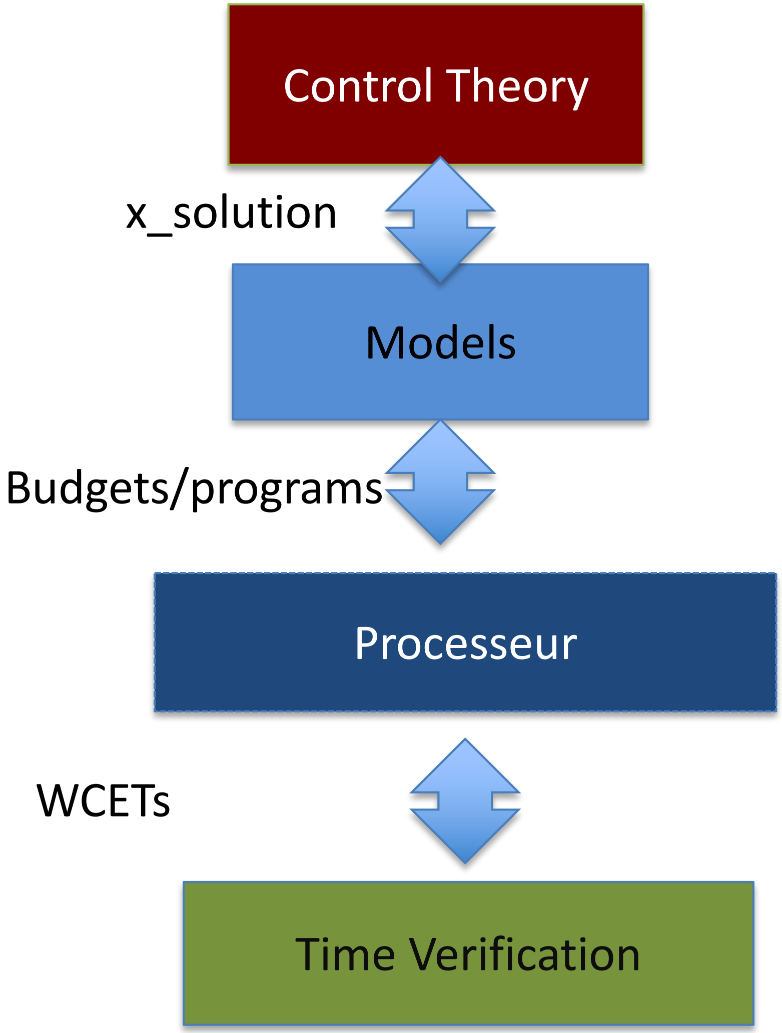
- Design of a physical system with time constraints
  - Verification of time constraints
- Probabilities: how do we compose?
- Measurement-based approaches
  - The (missunderstood) independence
  - The impact of the measurement protocol
- Analytical vs. measurement-based
- Back to models to solve the representativity
- Conclusion

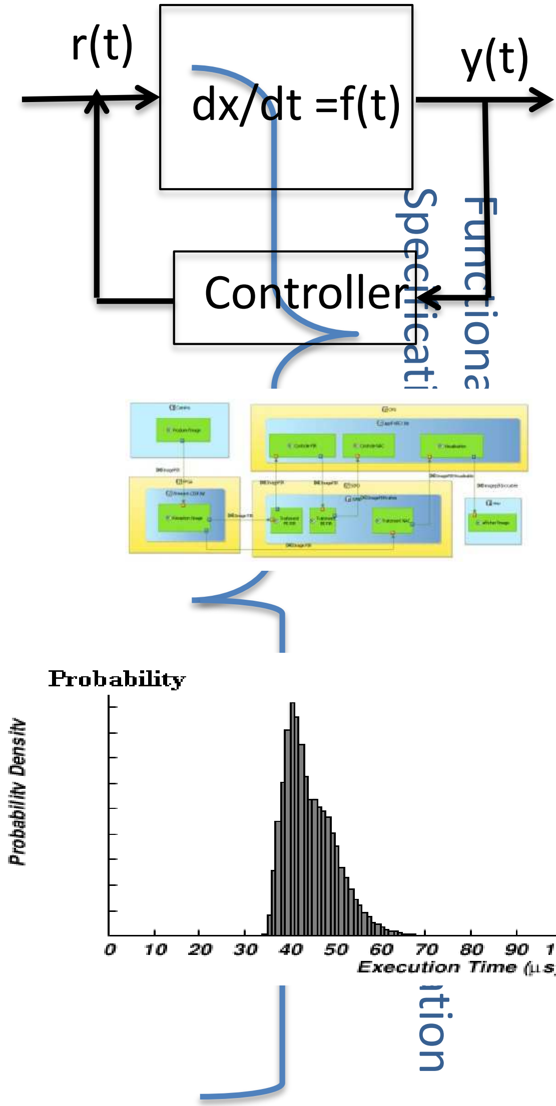
Design of a a physical system with time constraints

- Real-**time** systems
- Cyber-**physical** systems



Design of a physical system with time constraints (2)





Verification of time constraints

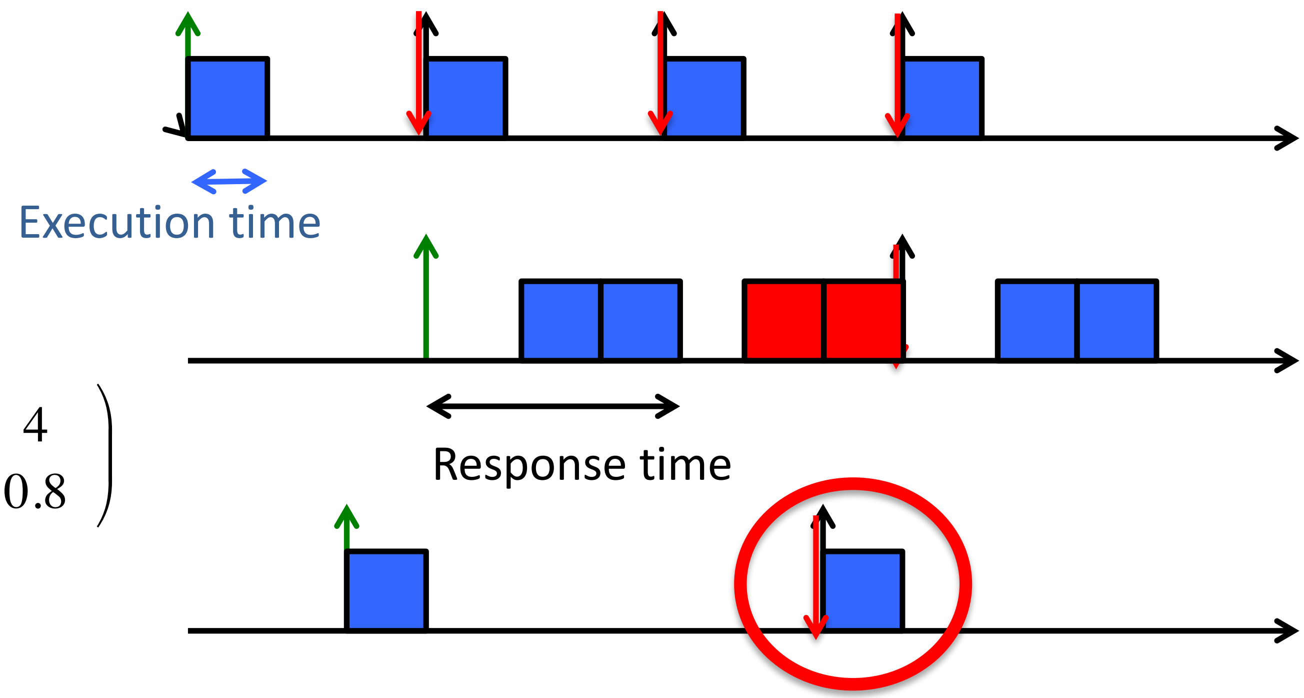
One processor, fixed-priority solution

τ_1 $(0, 1, 3, 3)$

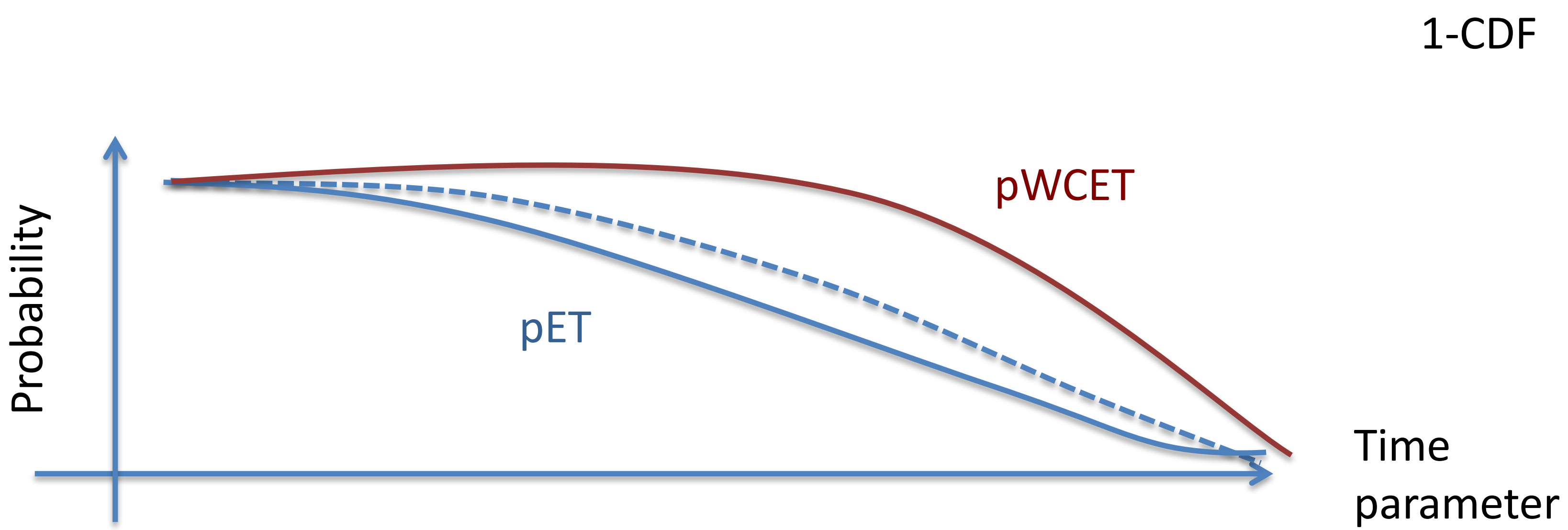
τ_2 $(3, \cancel{2}, 6, 6)$

$$\begin{pmatrix} 2 & 4 \\ 0.2 & 0.8 \end{pmatrix}$$

τ_3 $(2, 1, 6, 6)$



Probabilities: how do we compose?

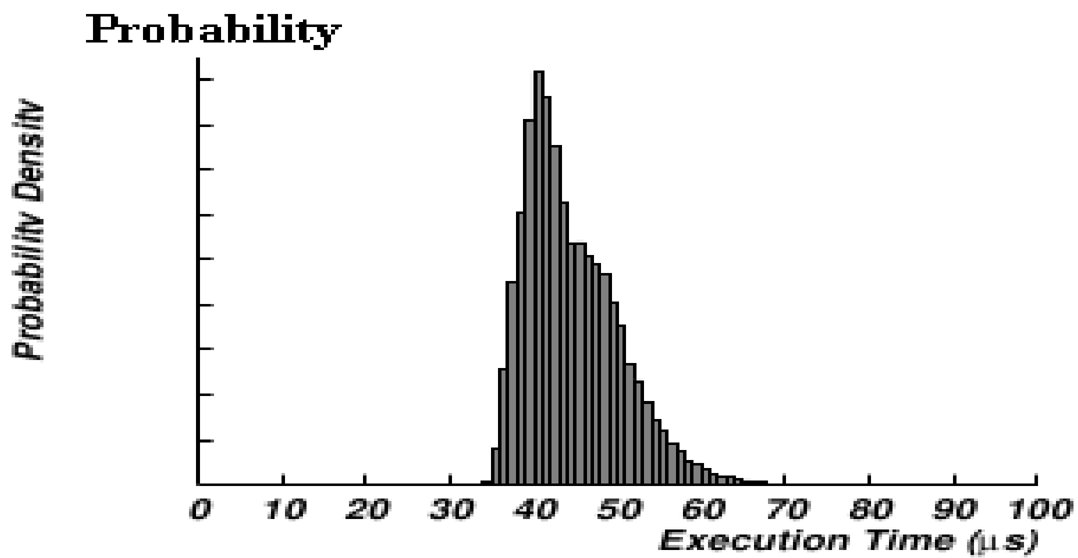


Measurements (statistical approaches)
Static analyses (probabilistic approaches)
Hybrid methods

pET: probabilistic Execution Time; pWCET: probabilistic Worst Case ET

How do we deal with probabilities?

For a program and a processor the execution time extremes are bounded by a Extreme Value Theory Distribution [Edgar et Burns at RTSS2001]

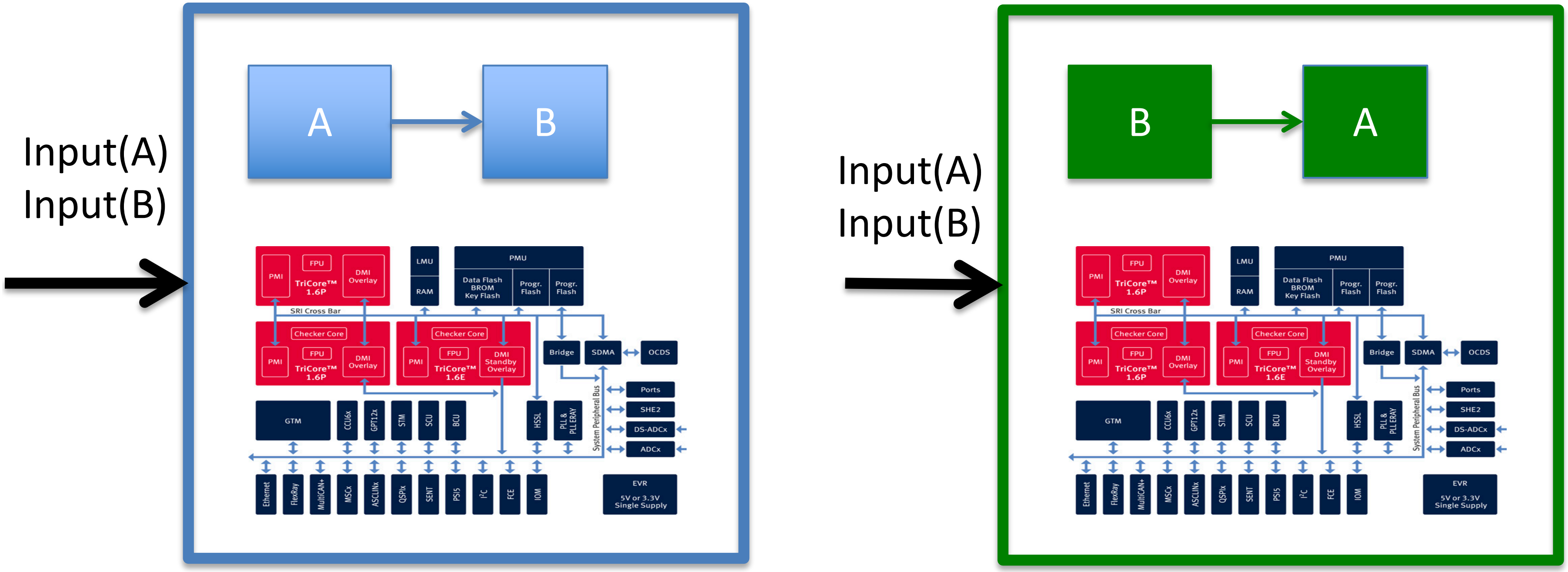


- Independence hypothesis
- Identically distributed hypothesis

Classes of independence

- (Functional) Independence between programs
- Statistical independence
- Probabilistic independence

Functional independence between programs



$$C_A = C_A \text{ and } C_B = C_B$$

Statistical dependence

{ 5, 18, 21, 27, 28, 30}

```
if y odd then
        { x = 2*y
         wait(y)
        }
else    {x =y + 3
        wait(x)
        }
```

$\xrightarrow{x}$

{ 10 21 42 54 31 33 }

```
for i= 1 to x
  wait (1)
```

{ 9 19 25 31 29 31 }

{ 10 21 42 54 31 33 }

The two sets of execution times are dependent

Two programs with (functional) dependences

Statistical independence

{ 68, 59, 84, 94, 100, 57}

```
if y odd then
        { x = 2*y
         wait(y)
        }
else    {x =y + 3
        wait(x)
        }
```

{39, 27, 39, 36, 34, 41}

```
for i= 1 to x
  wait (1)
```

{69, 63, 85, 95, 101, 61}

{39, 27, 39, 36, 34, 41}

The two sets of execution times are independent

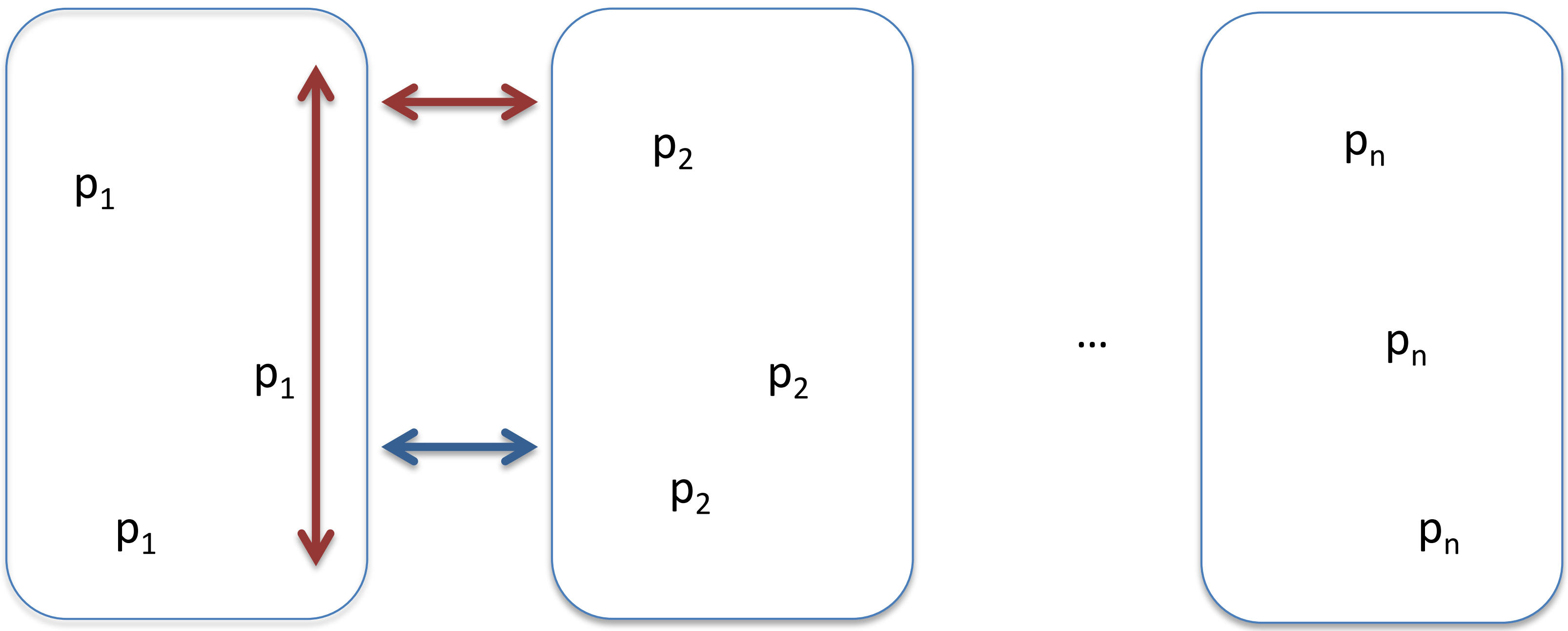
Two programs that have (functional) dependences

Multi-path programs

- The execution times are obtained per path and studied in different buckets

- All execution times are in one single bucket

Multi-paths and dependences



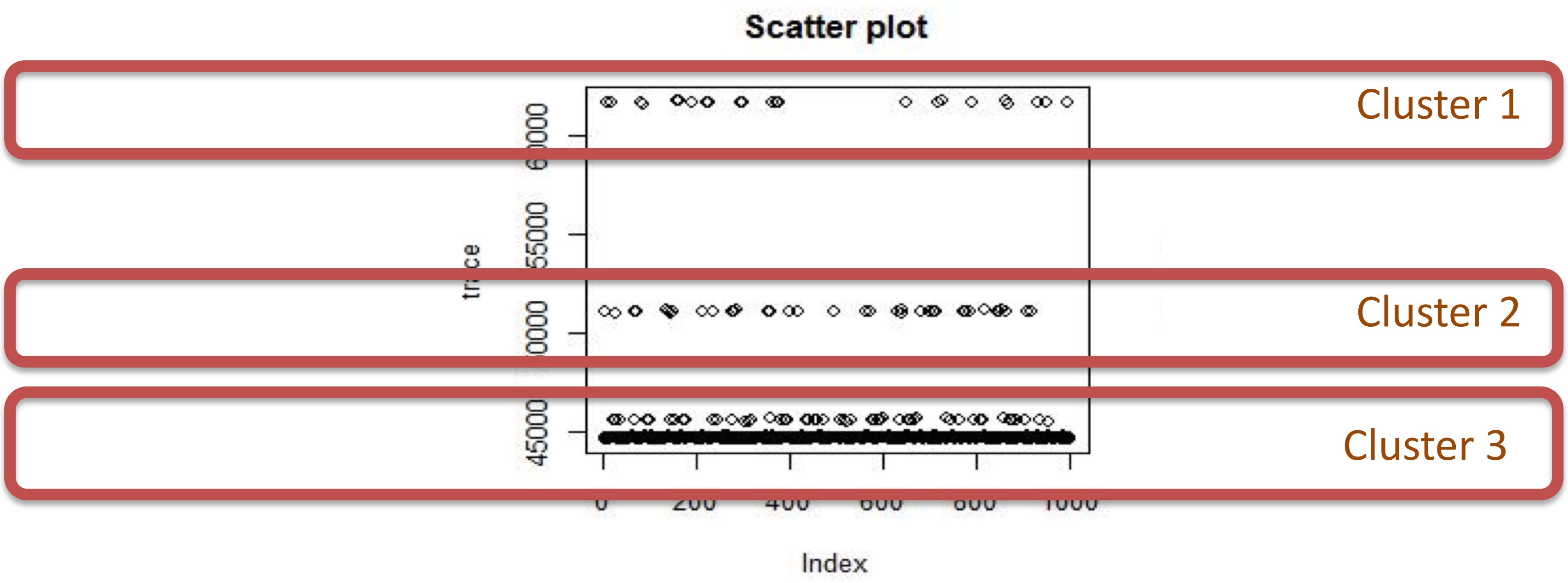
✓ Railway case:
Inter- and intra- bucket

✓ Avionics case :
Intra - bucket

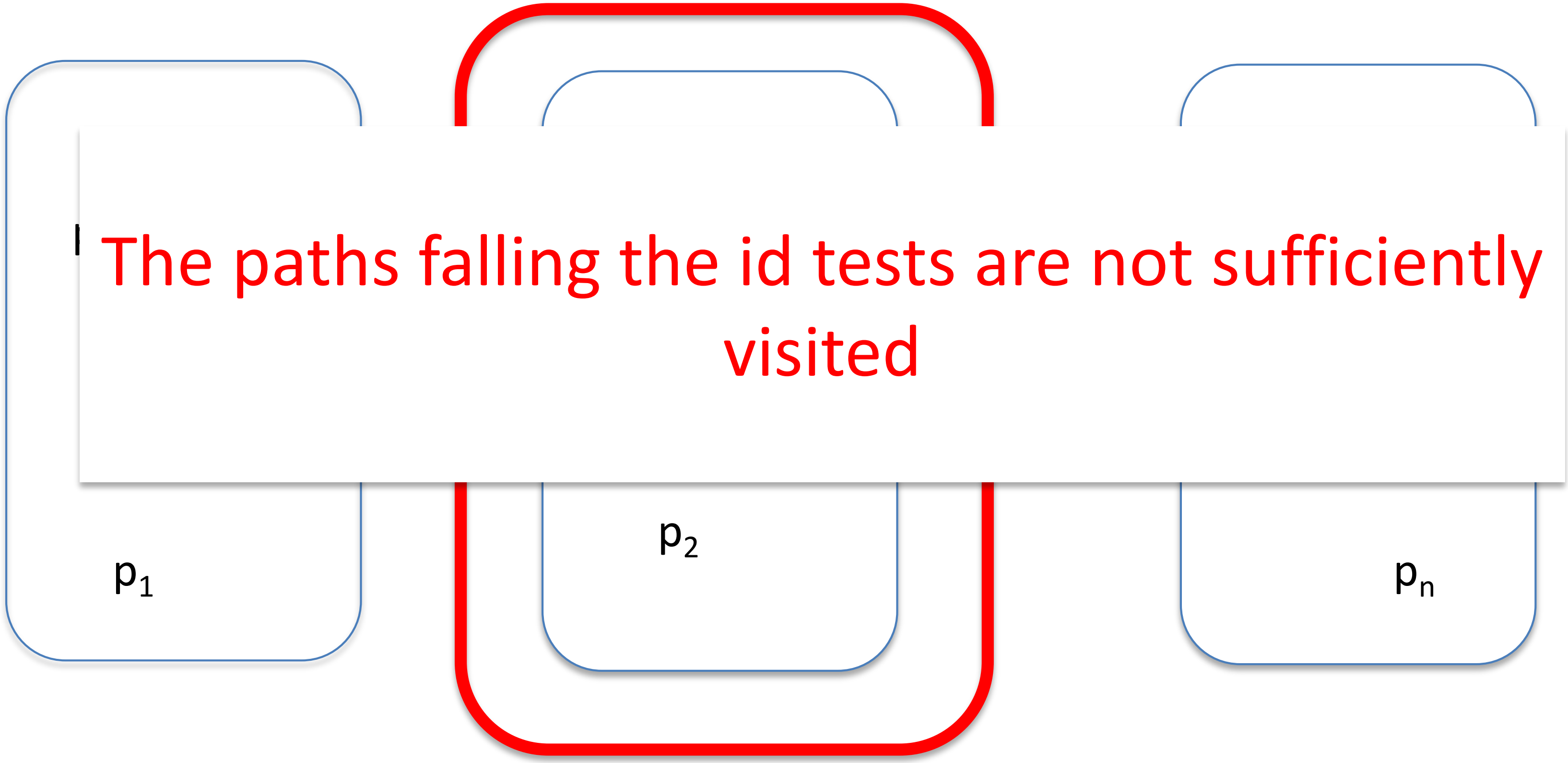
Dependences

- Decreasing the number of dependences is good, hoping to make them disappear is not realistic
- In presence of dependences, the order of execution times becomes important
  - A WCET measurement-based estimator should come with its own measurement protocol
- Manipulating the input execution times has a direct impact on the estimated pWCET
  - Monotonic property
  - Shuffling the input execution times

What dependences ?

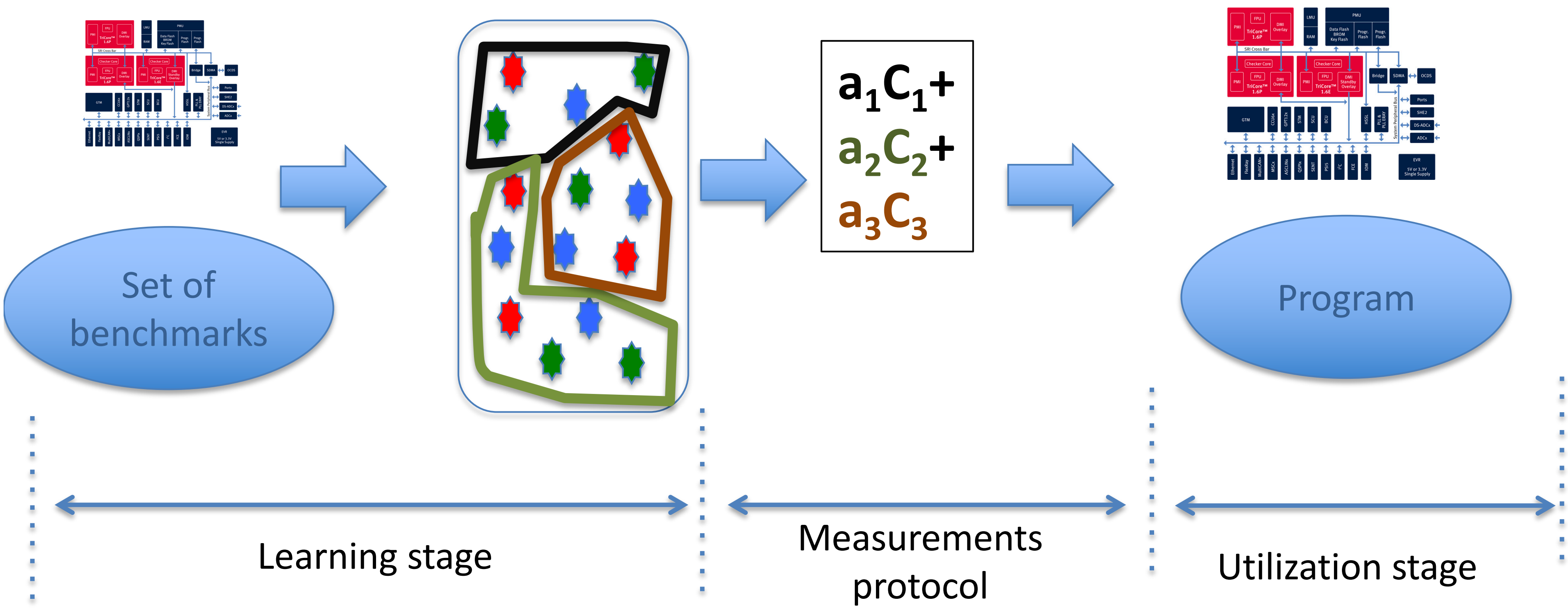


Multi-paths and identically distributed



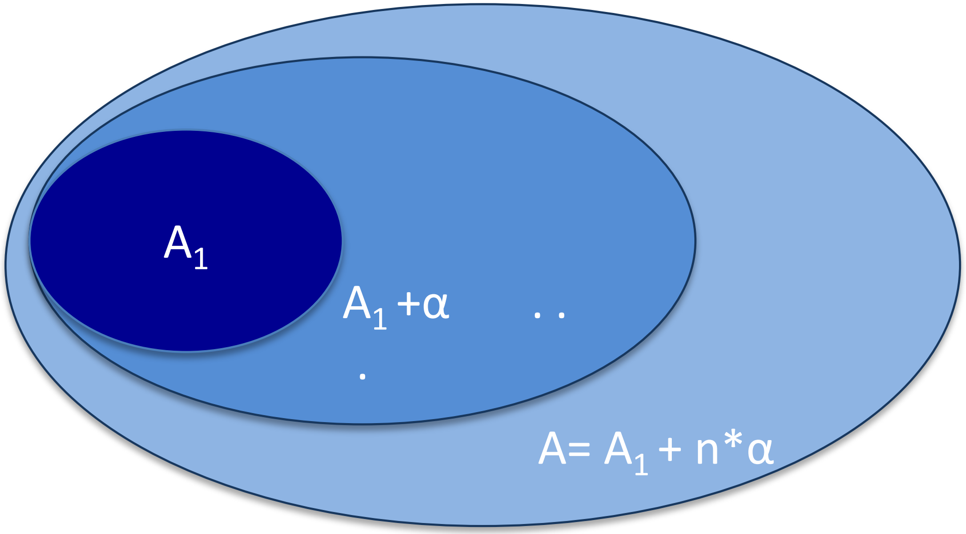
- ✓ Both railway and avionics: Within bucket
  - When identically distributed test is succesful, it is succesful for all paths
  - When it fails, it fails only for some buckets

Composing probabilities - a representativity concern?



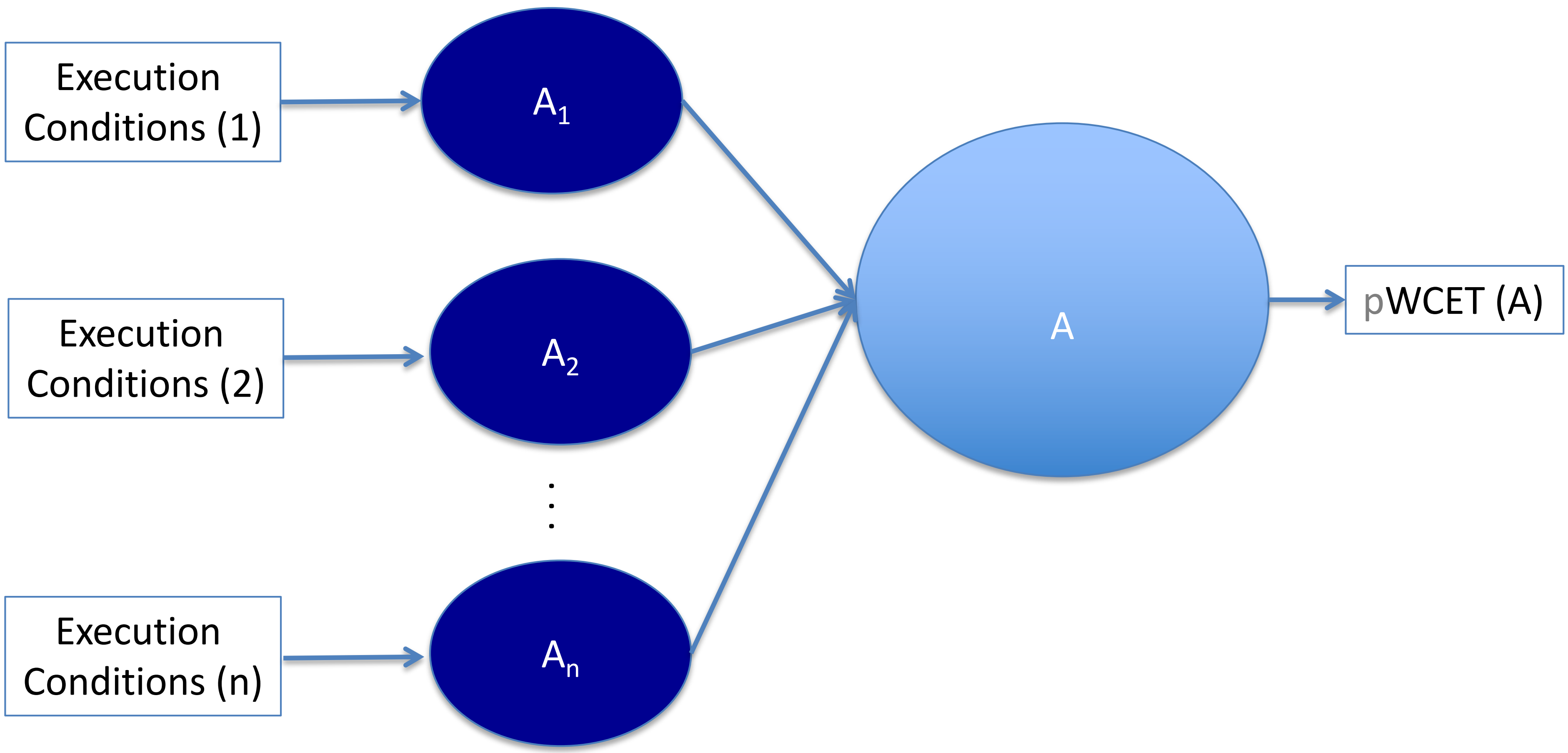
A proof of representativity requires elements from the other design levels

Representativity requires convergence



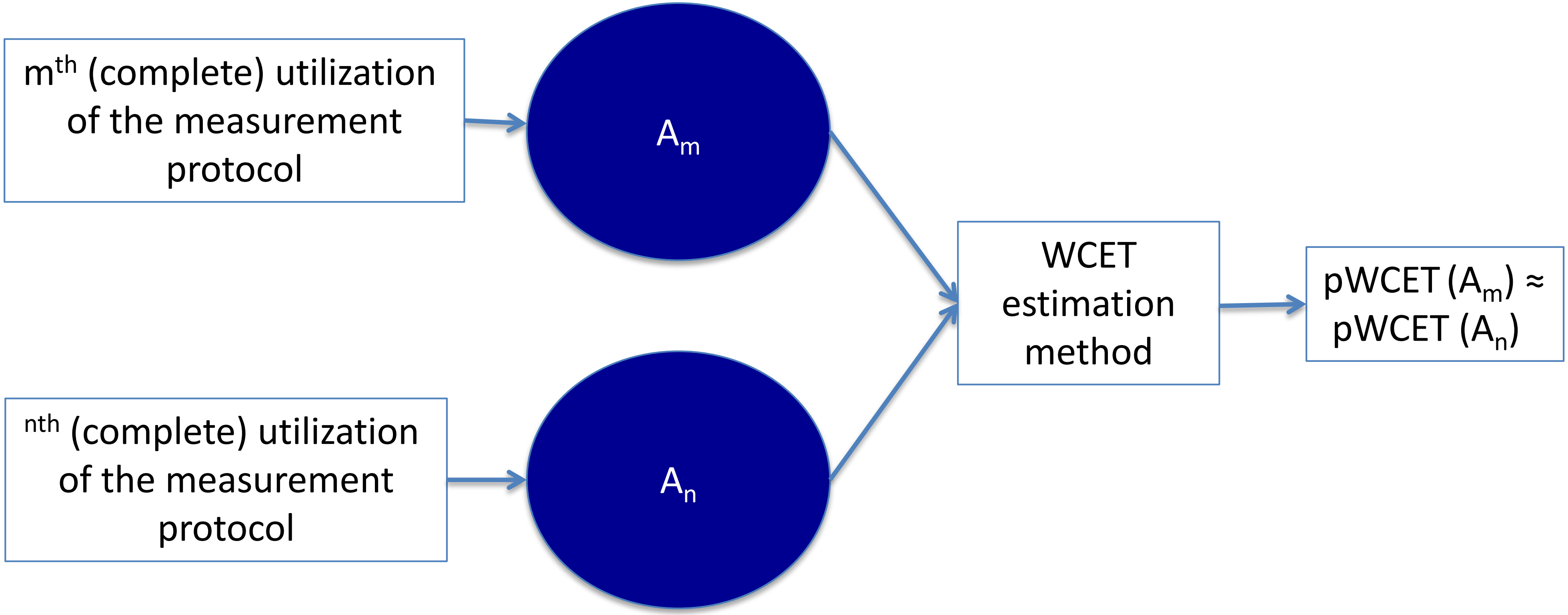
The statistical methods estimating extremes are not monotonic

The measurement protocol and the representativeness



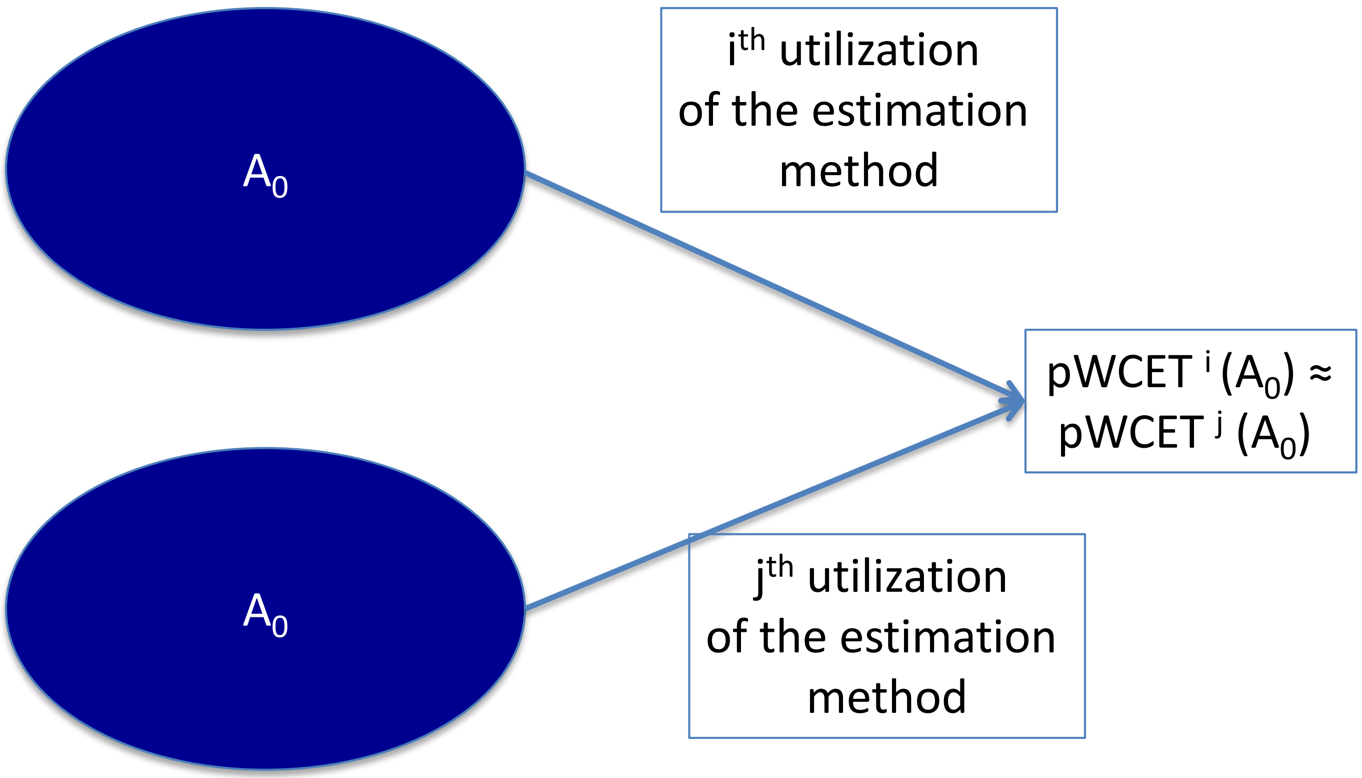
A_i is representative with respect to A if pWCET (A) is close to pWCET(A_i)

The reproducibility of the measurement protocol



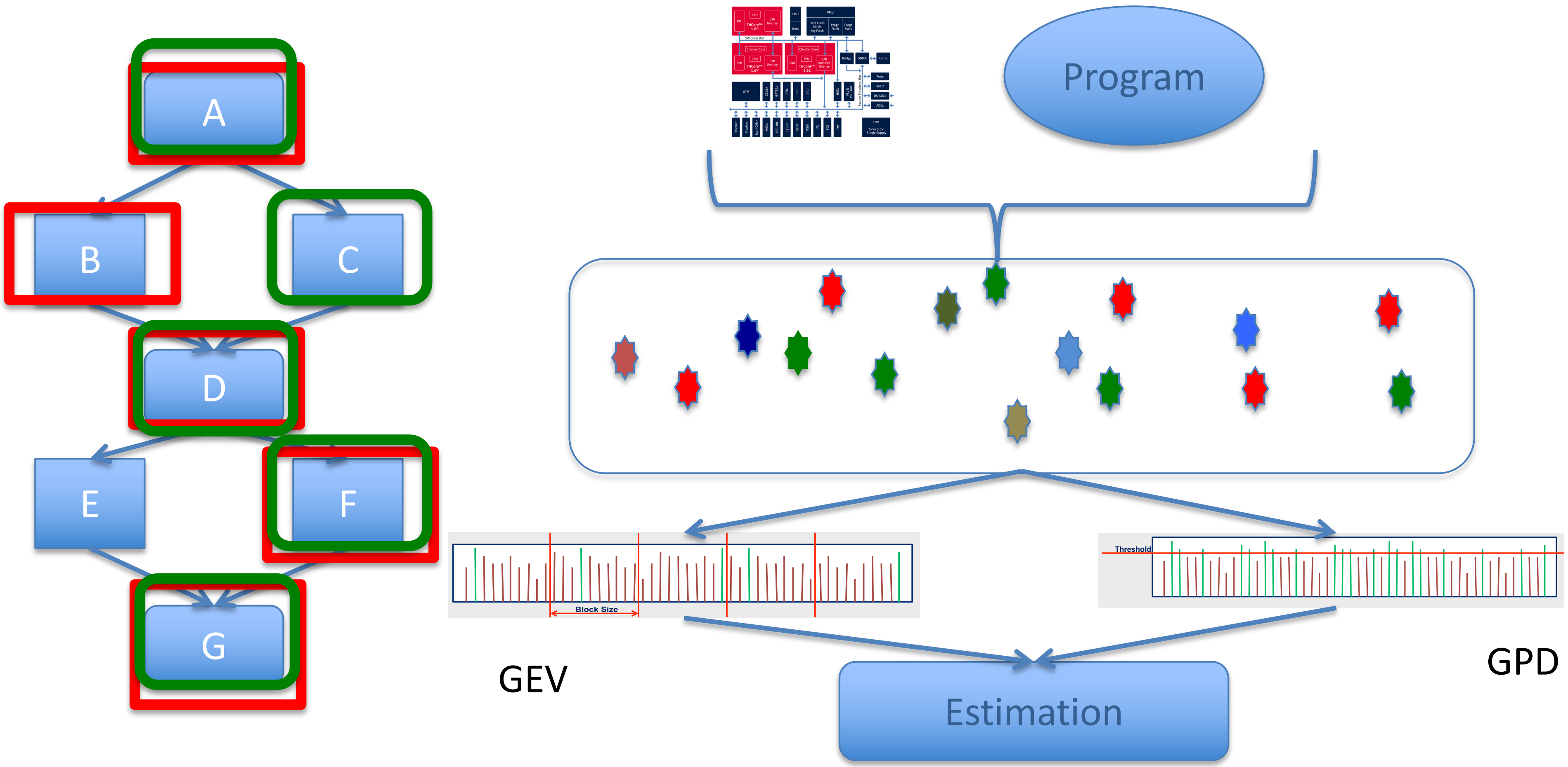
Any two different (and complete) utilizations of the measurement protocol from the same set of execution conditions should provide the same pWCET estimate

The reproducibility of the pWCET estimation method

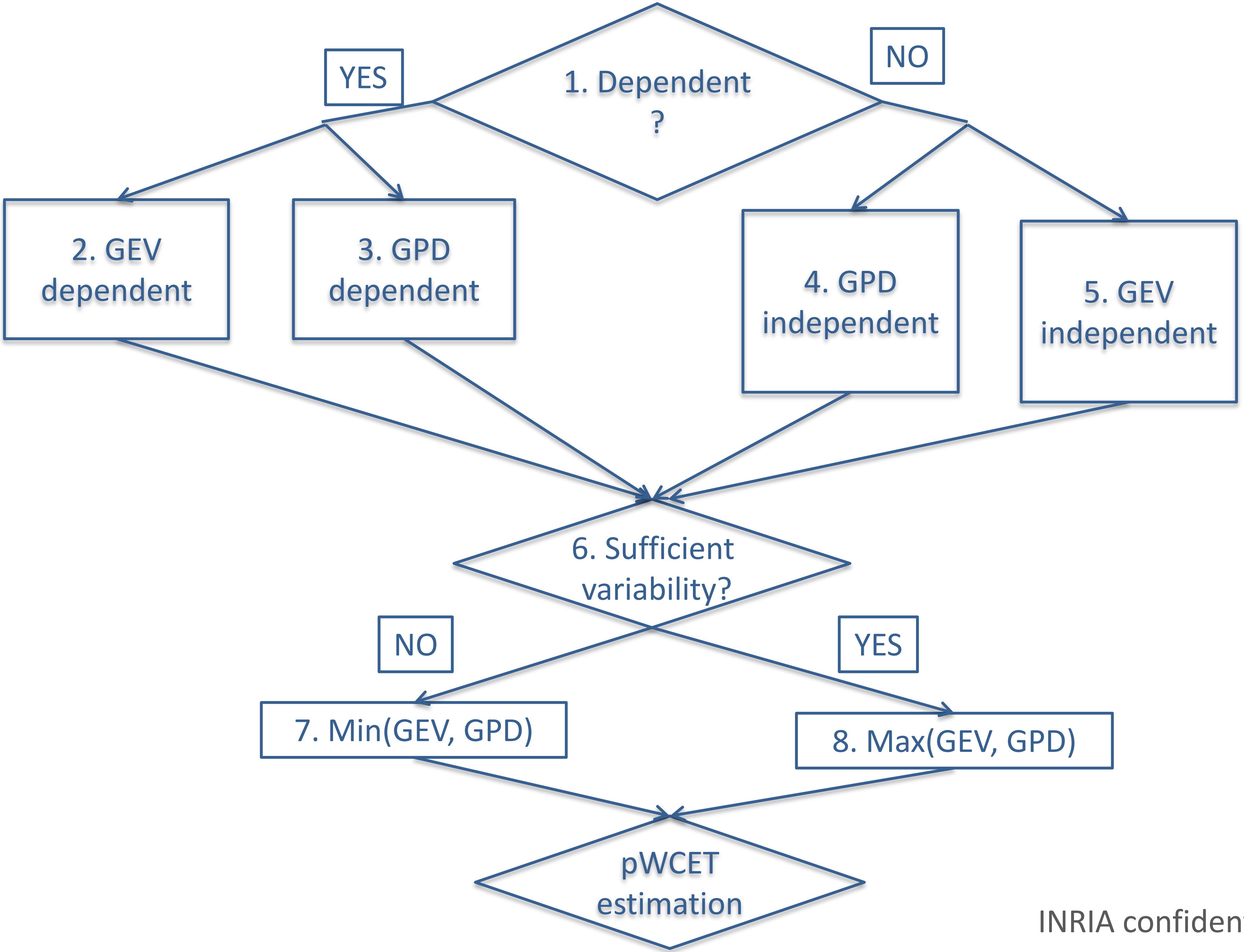


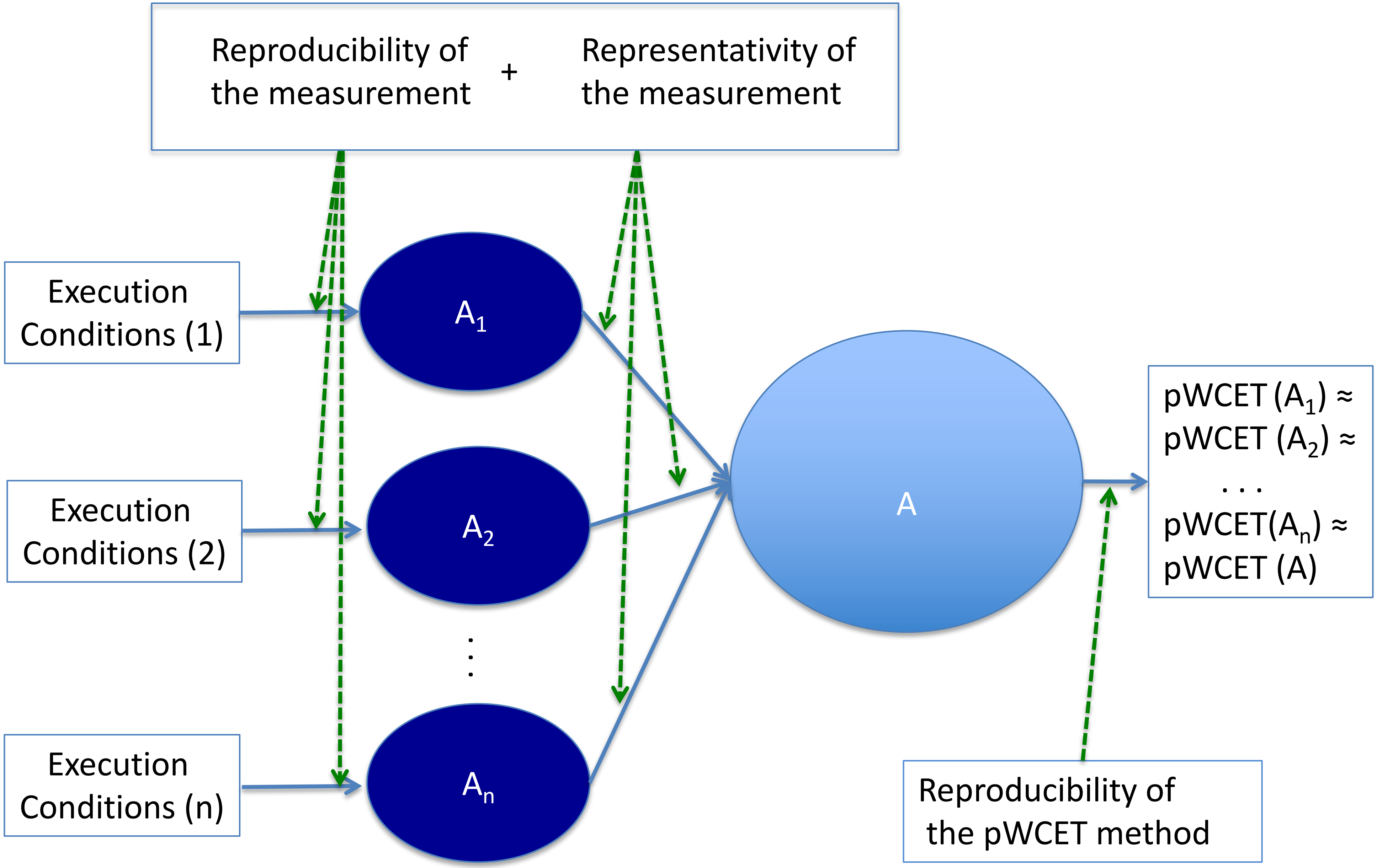
Any two different applications of the same set of execution conditions should provide the same pWCET estimate

Validation of a statistical test



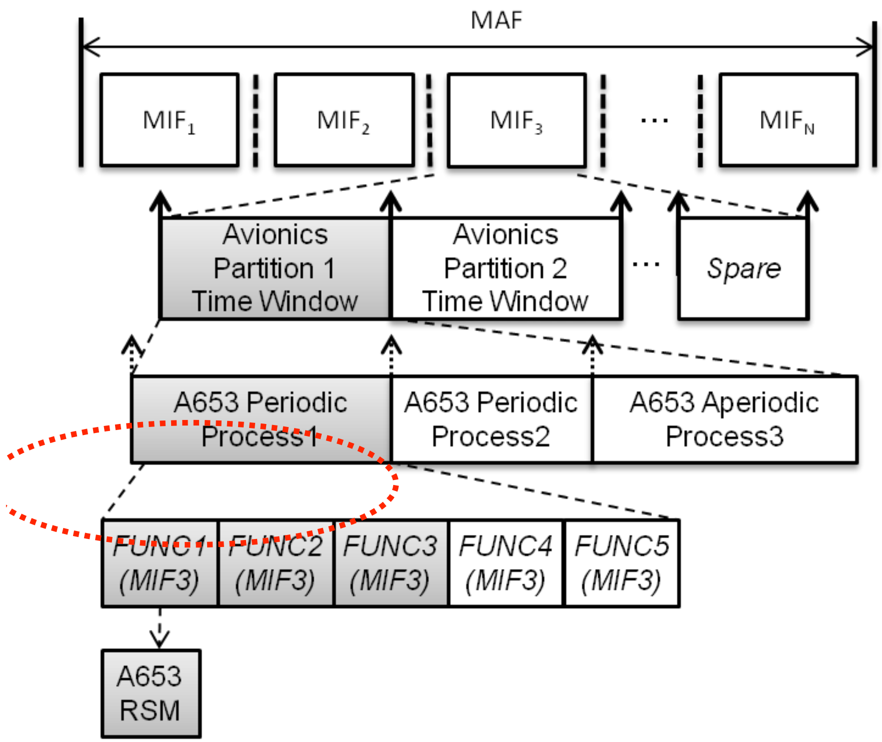
- Arguments complaint DO178B and IEC-61508

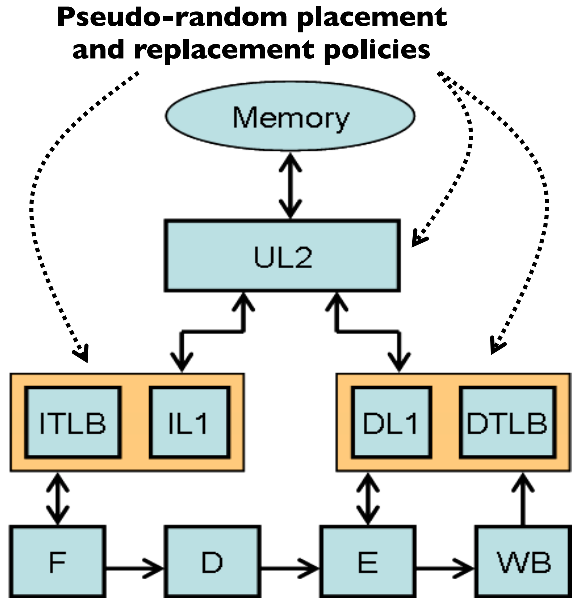

1. Dependent ?
YES
NO
2. GEV dependent
3. GPD dependent
4. GPD independent
5. GEV independent
6. Sufficient variability?
NO
YES
7. Min(GEV, GPD)
8. Max(GEV, GPD)
pWCET estimation


Reproducibility of the measurement
+
Representativity of the measurement
Execution Conditions (1)
Execution Conditions (2)
Execution Conditions (n)
A_1
A_2
A_n
A
pWCET (A_1) ≈
pWCET (A_2) ≈
. . .
pWCET(A_n) ≈
pWCET (A)
Reproducibility of the pWCET method

Avionics case study

- FP7 STREP PROARTIS Case study
  - IMA application performing maintance of the flight control computers
  - Randomized cache remplacement policies

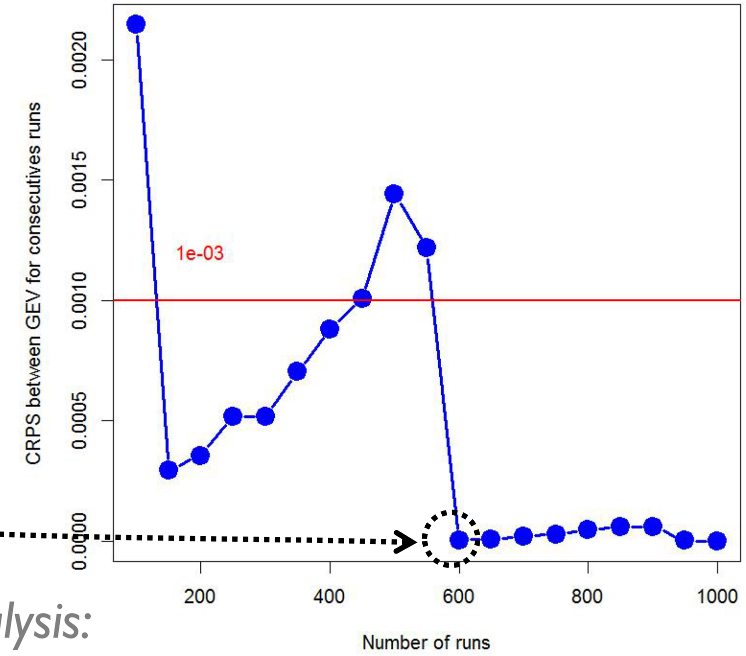




Avionics case study (2)

- Less than 5 minutes to provide a pWCET estimation

Function	FUNC 1	FUNC 2	FUNC 3
Collected Values	600	250	300
Iterations	12	5	6



F. Wartel et al., *Measurement-Based Probabilistic Timing Analysis: Lessons from an Integrated-Modular Avionics Case Study, SIES 2013*

Outline

- ✓ Design of a physical system with time constraints
  - o Verification of time constraints
- ✓ Probabilities: how do we compose?
- ✓ Measurement-based approaches
  - o The (missunderstood) independence
  - o The impact of the measurement protocol
- ❑ Analytical vs. measurement-based
- ❑ Back to models to solve the representativity
- ❑ Conclusion

Average versus worst case

What is the impact on an analysis?

- Average number of arrivals within a time interval

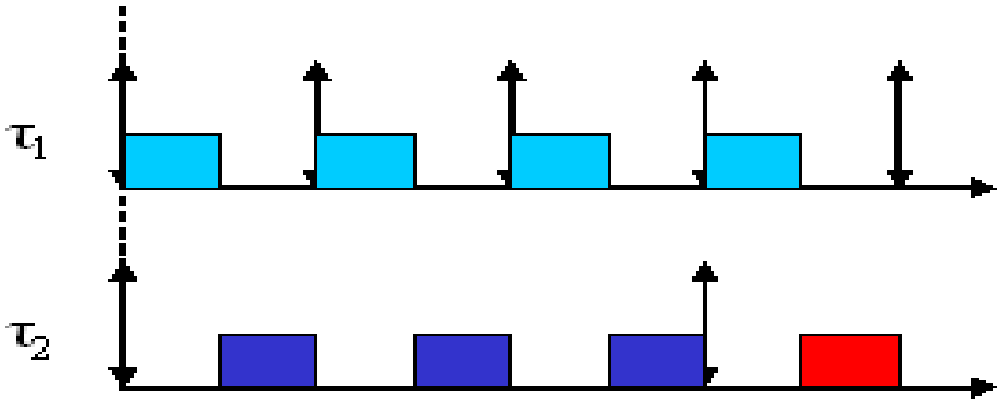
$$\tau_1 = \begin{pmatrix} 1 & 2 & 4 \\ 0.4 & 0.3 & 0.3 \end{pmatrix}, \text{ for } t_\Delta = 12$$

- Minimal inter-arrival times between two consecutive arrivals

$$\tau_1^* = \begin{pmatrix} 5 & 10 \\ 0.3 & 0.7 \end{pmatrix}$$

Optimal fixed-priority scheduler

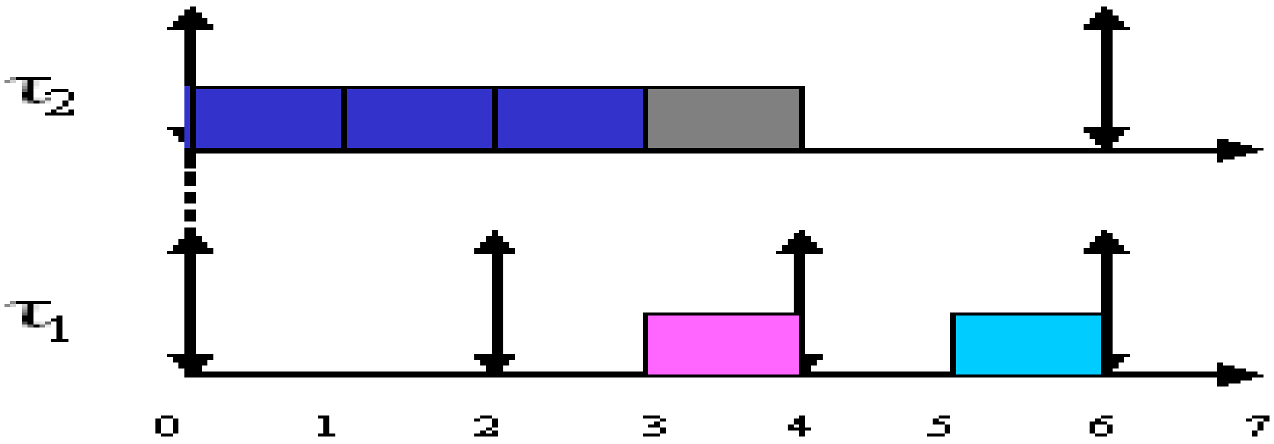
- Rate Monotic is not optimal



$$\tau_1 = \left(\begin{pmatrix} 1 \\ 1 \end{pmatrix}, 2, 2, 40\% \right) \qquad \tau_2 = \left(\begin{pmatrix} 3 & 4 \\ 0.5 & 0.5 \end{pmatrix}, 6, 6, 30\% \right)$$

Optimal (task) fixed-priority scheduler (2)

- A feasible task fixed-priority assignement



$$\tau_1 = \left(\begin{pmatrix} 1 \\ 1 \end{pmatrix}, 2, 2, 40\% \right) \qquad \tau_2 = \left(\begin{pmatrix} 3 & 4 \\ 0.5 & 0.5 \end{pmatrix}, 6, 6, 30\% \right)$$

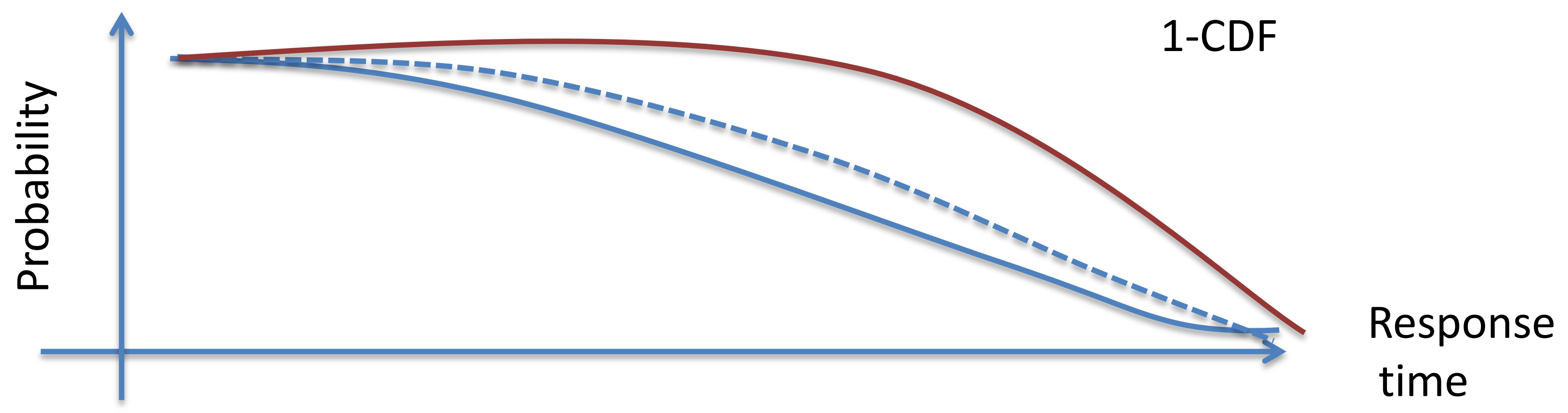
Optimal (task) fixed-priority scheduler (3)

- Theorem (Maxim,2011)

  The order of higher priority tasks does not have any impact on the probability of missing the deadline of a task
- Audsley reasoning may be proposed

Analytical verification of time constraints

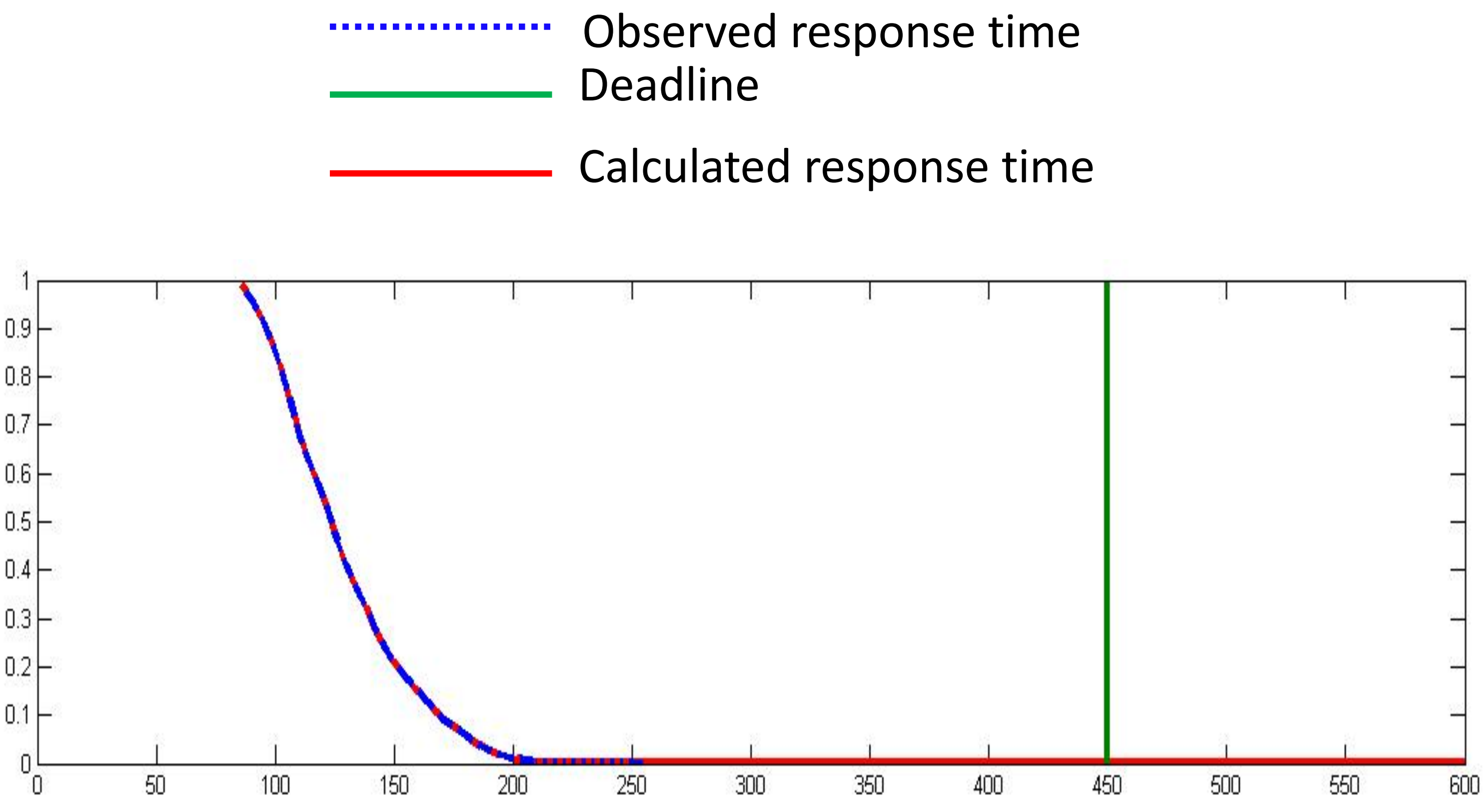
The first response time calculation for systems with multiple probabilistic parameters (DC13)



- Probabilistic independence required between the probabilistic parameters

[DC13] D. Maxim et L. Cucu-Grosjean, *Response Time Analysis for Fixed-Priority Tasks with Multiple Probabilistic Parameters*", IEEE Real-Time Systems Symposium (RTSS 2013), Vancouver, December 3-6, 2013

Analytical versus simulation



Probability of not meeting the deadline : 9.24819 x 10^{-14}

Outline

- ✓ Design of a physical system with time constraints
  - Verification of time constraints
- ✓ Probabilities: how do we compose?
- ✓ Measurement-based approaches
  - The (missunderstood) independence
  - The impact of the measurement protocol
- ✓ Analytical vs. measurement-based
- ❑ Back to models to solve the representativity
- ❑ Conclusion

Design of a physical system with time constraints

Control Theory

x_solution

Models

Programs/budgets

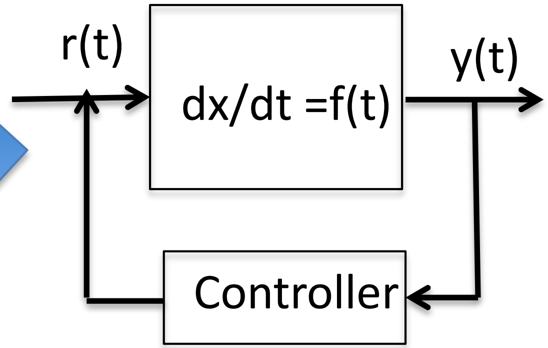
Processeur

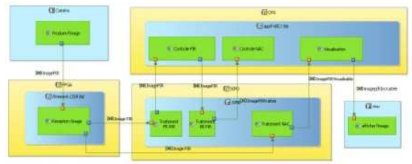
WCETs

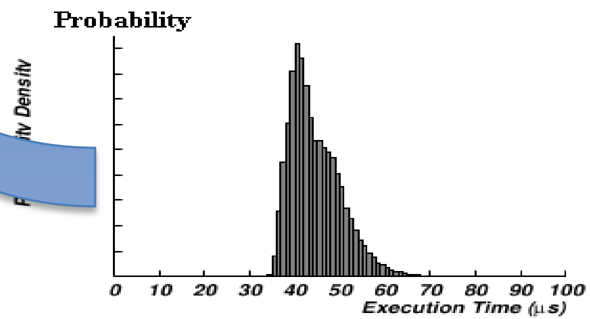
Time Verification

Probabilistic descriptions

Classification of programs

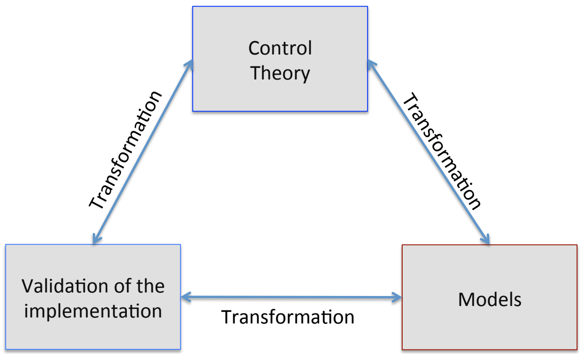






Possibles steps (and open problems)

- Worst case probabilistic models
  - Understanding the relations between different design levels
  - Choice of properties to be probabilistically described
  - Proposition of new models
- Time constraints analyses

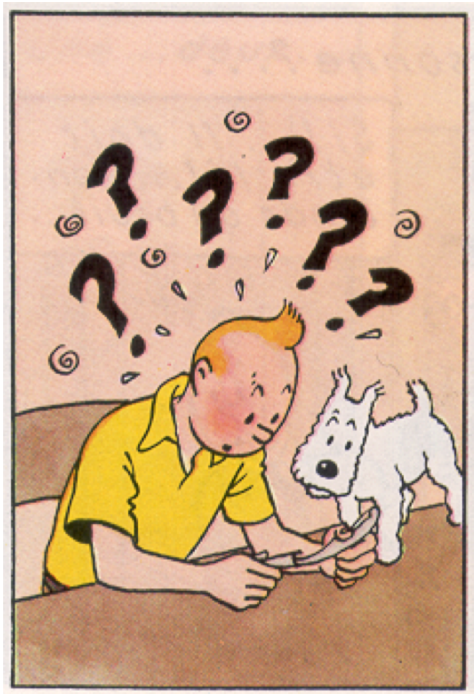


- Validation and certification of the framework
  - Proposition of a complementary transformation

CONCLUSIONS

- Time critical embedded systems are everywhere
- There is an important bareer while building tomorrow time critical embedded systems
- Proving correct such framework requires an important effort from different communities

Je vous remercie pour votre attention



liliana.cucu@inria.fr