PHYLOG certification methodology: a sane way to embed multi-core processors

Frédéric Boniol, Youcef Bouchebaba, Julien Brunel, Kevin Delmas, Thomas Loquen, Alfonso Mascarenas Gonzalez, Claire Pagetti, Thomas Polacsek, Nathanaël Sensfelder

ERTS, January 30th, 2020

ONERA

THE FRENCH AEROSPACE LAB

retour sur innovation

Introduction

Context: avionic systems

Topic:

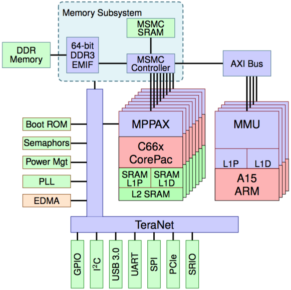
- MultiCore Processors (MCP)
- Certification: MCP-CRI standard

Observation: certification is a difficult task because of

- internal complexity of MCP
- complexity of MCP-CRI objectives

Phylog contribution: a framework to ease the certification of MultiCore Processors for avionic systems

What is a multi-core processor (MCP)?



= Complex architecture composed of

- computing cores, signal processing cores, DMAs,
- caches, memories,
- buses, IO devices...

(Pro) Allows multiple functions to be executed in parallel

(Cons) High integration density

⇒ hard to master the internal normal / abnormal behavior

Parallelism + shared resources

→ risk of interference

→ risk of delays and non-determinism (due to interference)

⇒ key issues for certification

Certification. . .

Certification =

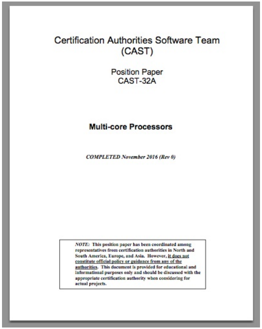
- evaluation of an argumentation, to convince that a system (i.e., its architecture, its settings, including mitigation means. . .) satisfies certification objectives

⇒ Certification objectives for MCP?

[1] "Certification Review Item for Multi-Core (MCP-CRI)" (nov. 2016)

⇒ defines 9 certification objectives about

- SW development and verification planing
- resources settings
- resource usage and interference handling
- safety handling. . .



Certification Authorities Software Team
(CAST)

Position Paper
CAST-32A

Multi-core Processors

COMPLETED November 2016 (Rev 0)

NOTE: **This position paper has been coordinated among representatives from certification authorities in North and South America, Europe, and Asia. However, it does not constitute official policy or guidance from any of the authorities. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.**

PHYLOG approach…

Phylog ideas

1. transcription of the MCP-CRI objectives in a more (pseudo-)formal graphical way

2. use of formal methods to support
   - ⇒ interference analysis
   - ⇒ safety analysis

3. use of models to support analyses and to ease dialogues between applicants and certification authorities

Agenda

1. Transcription of MCP-CRI objectives. . .
2. PML: a meta model certification-oriented for MCP
3. Interference analysis
4. Conclusion and future work

Transcription of MCP-CRI objectives. . .

Why: to clarify what to do and how to organize the arguments

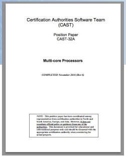
How: Argumentation patterns

- close to GSN, CAE notations
- organize in diagram form the various elements, formal and informal, that contribute to the justification of a result (such as safety, security, correctness)

Idea: define an argumentation pattern per MCP-CRI objective

⇒ Example: Resource Usage 3 (interference identification and mitigation)

Example: Resource Usage 3 (RU3)



Certification Authorities Software Team
(CAST)

Position Paper
CAST-32A

Multi-core Processors

COMPLETED November 2016 (Rev 0)

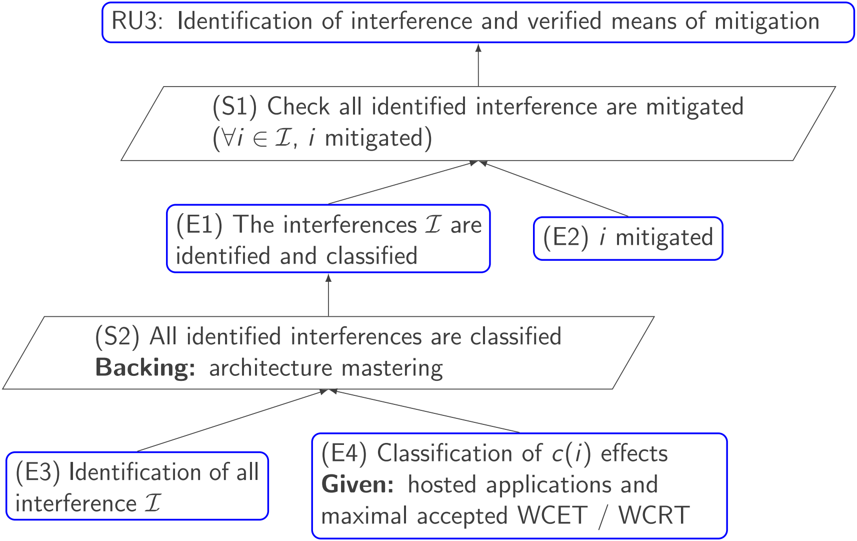
NOTE: This position paper has been coordinated among representatives from certification authorities in North and South America, Europe, and Asia. However, it does not constitute official policy or guidance from any of the authorities. This document is provided for educational and informational purposes only and should be discussed with the appropriate certification authority when considering for actual projects.

Resource Usage 3 (RU3) (MCP-CRI, page 13)

"The applicant

- *has identified the interference channels that could permit interference to affect the software applications hosted on the MCP cores,*
- *and has verified the applicant's chosen means of mitigation of the interference."*

Resource Usage 3 (RU3) objective



RU3: Identification of interference and verified means of mitigation

(S1) Check all identified interference are mitigated ($\forall i \in \mathcal{I}$, i mitigated)

(E1) The interferences $\mathcal{I}$ are identified and classified

(E2) i mitigated

(S2) All identified interferences are classified **Backing:** architecture mastering

(E3) Identification of all interference $\mathcal{I}$

(E4) Classification of $c(i)$ effects **Given:** hosted applications and maximal accepted WCET / WCRT

Resource Usage 3 (RU3) objective

Next issue: How to fulfill the leaves of the argumentation patterns

RU3 example

- how to identify / enumerate the interference (E3)?
- how to classify the interference (E4)?
- in a feasible way?

⇒ **Idea:** automatic computation

⇒ **Needs:** models (of the internal architecture of the MCP and its configuration).

Agenda

1 Transcription of MCP-CRI objectives. . .

2 PML: a meta model certification-oriented for MCP

3 Interference analysis

4 Conclusion and future work

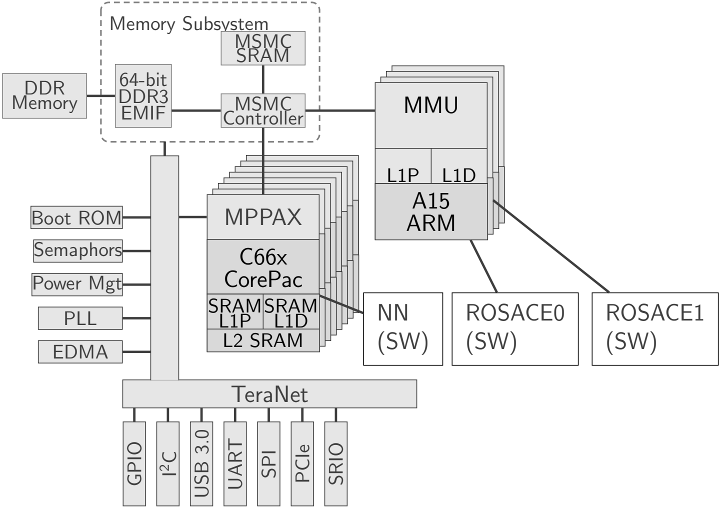
Why a specific meta-model?

Needs:

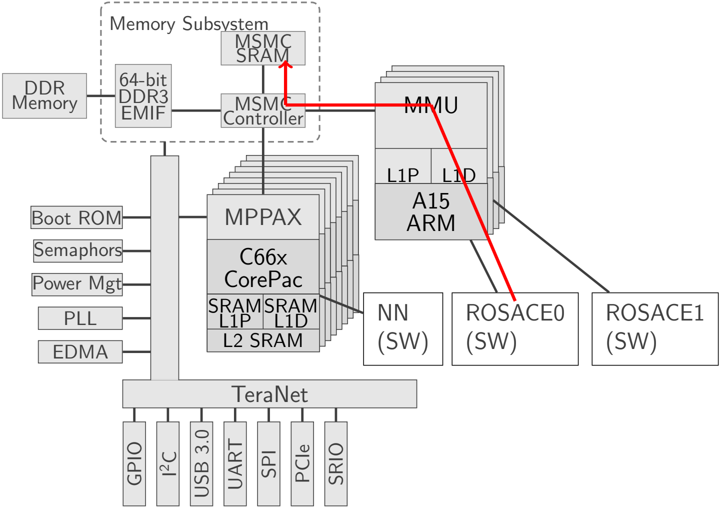
- an accurate abstraction able to capture the concepts mentioned in the MCP-CRI
- as simple as possible
  - only for certification concerns (not for design)

⇒ **Question:** what is MCP-CRI talking about?

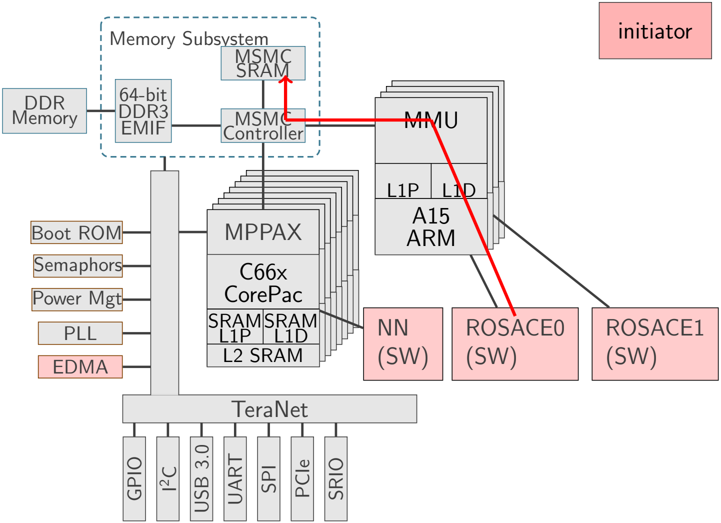
PML (1/3): Keystone example



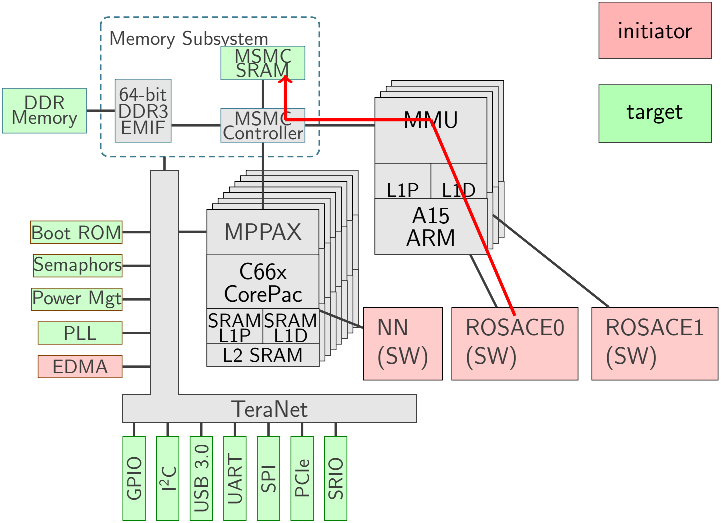
PML (1/3): Keystone example



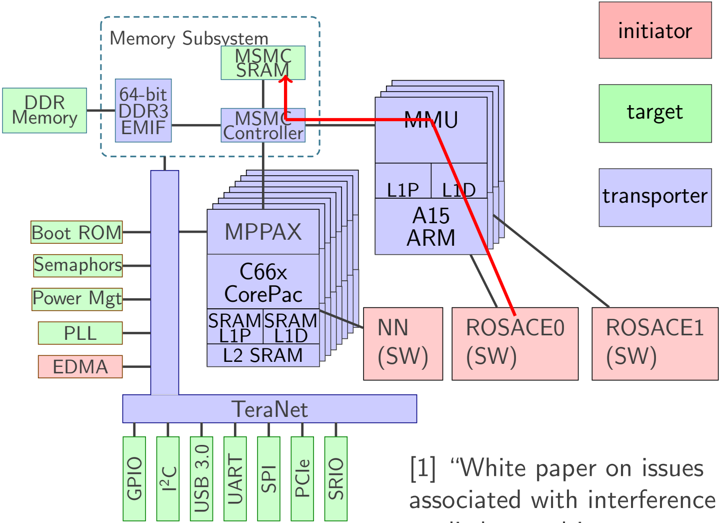
PML (1/3): Keystone example



PML (1/3): Keystone example



PML (1/3): Keystone example

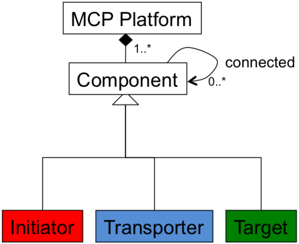


[1] "White paper on issues associated with interference applied to multi-core processors". X. Jean et al., 2016

PML (1/3)

⇒ **1st Idea:** MPC platform = organised set of

- initiators
- targets
- transporters



PML (2/3)

2d Idea: characterize each component with the services it provides

- to capture the normal / abnormal behavior of the platform

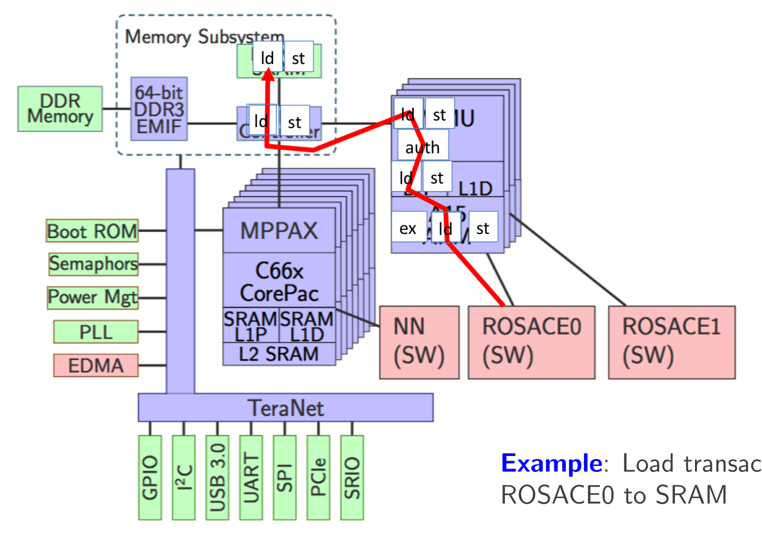
⇒ **6 types of services**

- execute (ex), load (ld), store (st), authorize (auth), dispatch (dp), copy (cp)

⇒ **transaction =**

- is a request of type T
- from 1 iniator
- to n target services of type T
- through a path of transporter services of type T

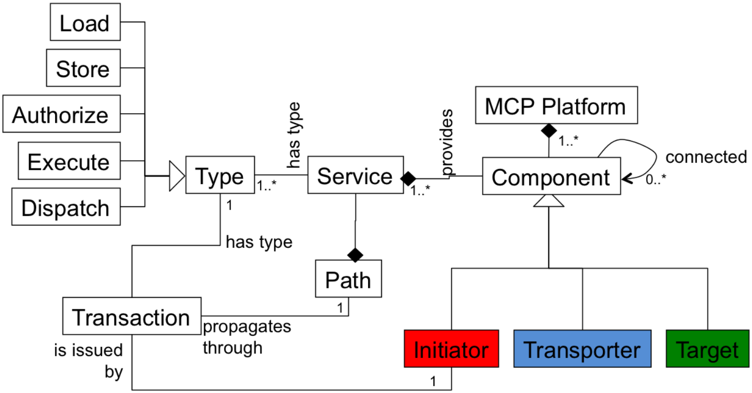
PML (2/3): Keystone example



Example: Load transaction from ROSACE0 to SRAM

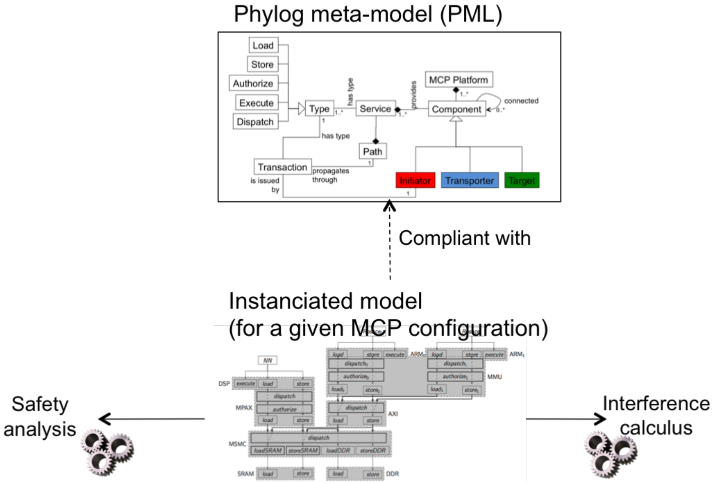
PML (3/3)

PML (simplified view)



PML: a meta model certification oriented for MCP

⇒ allows export to dedicated view points: interference analysis, and safety analysis.



Phylog meta-model (PML)

Compliant with

Instanciated model
(for a given MCP configuration)

Safety analysis

Interference calculus

Agenda

1 Transcription of MCP-CRI objectives…

2 PML: a meta model certification-oriented for MCP

3 Interference analysis

4 Conclusion and future work

Interference definition

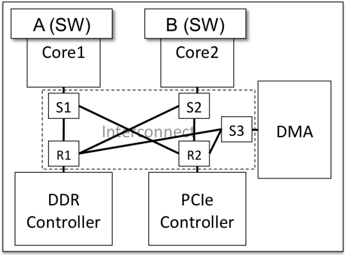
⇒ **Interference definition**

⇒ **Method to enumerate all interference**

Interference definition

Interference scenario

- let A and B two initiator components
- let t_A and t_B two "transactions" issued by A and B
- let $P(t_A)$ and $P(t_B)$ the paths of t_A and t_B (i.e., the services crossed by t_A and t_B)



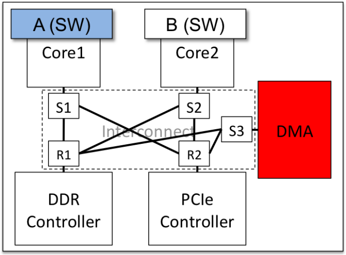
$\Rightarrow$ if there exists a service $r \in P(t_A) \cap P(t_B)$, then

$\boxed{\langle t_A || t_B \rangle \text{ is an interference scenario on } r}$

Interference definition

Interference scenario

- let A and B two initiator components
- let t_A and t_B two “transactions” issued by A and B
- let $P(t_A)$ and $P(t_B)$ the paths of t_A and t_B (i.e., the services crossed by t_A and t_B)



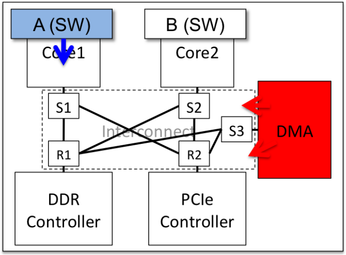
$\Rightarrow$ if there exists a service $r \in P(t_A) \cap P(t_B)$, then

$\langle t_A || t_B \rangle$ is an interference scenario on r

Interference definition

Interference scenario

- let A and B two initiator components
- let t_A and t_B two "transactions" issued by A and B
- let $P(t_A)$ and $P(t_B)$ the paths of t_A and t_B (i.e., the services crossed by t_A and t_B)



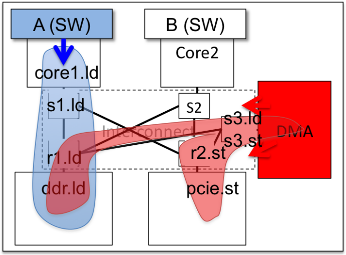
$\Rightarrow$ if there exists a service $r \in P(t_A) \cap P(t_B)$, then

$$\boxed{\langle t_A || t_B \rangle \text{ is an interference scenario on } r}$$

Interference definition

Interference scenario

- let A and B two initiator components
- let t_A and t_B two "transactions" issued by A and B
- let $P(t_A)$ and $P(t_B)$ the paths of t_A and t_B (i.e., the services crossed by t_A and t_B)



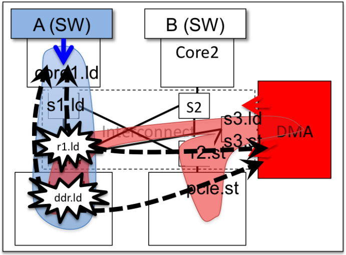
$\Rightarrow$ if there exists a service $r \in P(t_A) \cap P(t_B)$, then

$$\boxed{\langle t_A || t_B \rangle \text{ is an interference scenario on } r}$$

Interference definition

Interference scenario

- let A and B two initiator components
- let t_A and t_B two “transactions” issued by A and B
- let $P(t_A)$ and $P(t_B)$ the paths of t_A and t_B (i.e., the services crossed by t_A and t_B)



$\Rightarrow$ if there exists a service $r \in P(t_A) \cap P(t_B)$, then

$$\boxed{\langle t_A || t_B \rangle \text{ is an interference scenario on } r}$$

Interference definition

$\Rightarrow$ **Enumeration of all binary interference scenarios**

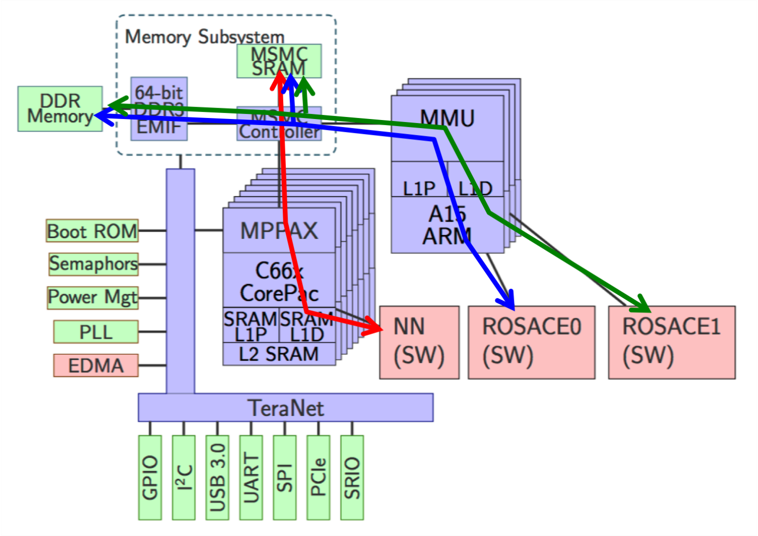
$$\mathcal{I}^2 = \left\{ \langle t_A || t_B \rangle \mid t_A, t_B : \mathit{transaction}, \exists r \in P(t_A) \cap P(t_B) \right\}$$

$\Rightarrow$ **Enumeration of all binary interference-free scenarios**

$$\mathcal{IF}^2 = \left\{ \langle t_A || t_B \rangle \mid t_A, t_B : \mathit{transaction}, P(t_A) \cap P(t_B) = \emptyset \right\}$$

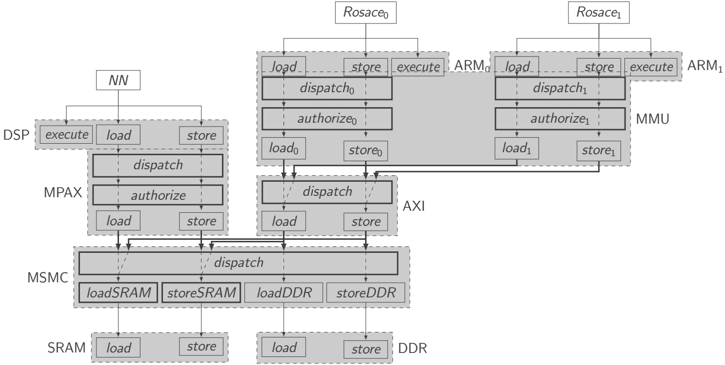
$\Rightarrow$ Can be generalized to *n*-ary interference channels / scenarios

Interference definition: Keystone example



Interference definition: Keystone example

⇒ Phylog model



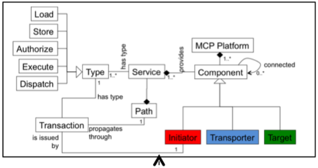
⇒ 32 binary interference scenarios

⇒ 32 ternary interference scenarios

⇒ 23 bunary interference-free scenarios

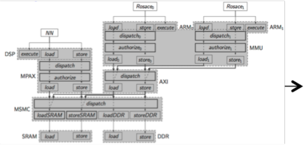
Interference argumentation: synthesis

Phylog meta-model (PML)



Compliant with

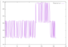
Architecture model



Interference calculus

List of
- Interference scenarios
- Interference-free scenarios

Classification benchmarks



Validation benchmarks

RU3: Identification of interference and verified means of mitigation

(S1) Check all identified interference are mitigated ($\forall i \in \mathcal{I}$, i mitigated)

(E1) The interferences $\mathcal{I}$ are identified and classified

(E2) i mitigated

(S2) All identified interferences are classified
Backing: architecture mastering

(E3) Identification of all interference $\mathcal{I}$

(E4) Classification of $c(i)$ effects
Given: hosted applications and maximal accepted WCET / WCRT

Constraint programming framework (IDP)

Agenda

1. Transcription of MCP-CRI objectives…

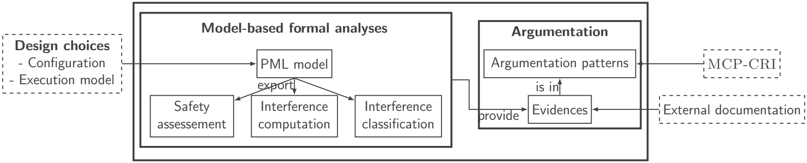
2. PML: a meta model certification-oriented for MCP

3. Interference analysis

4. Conclusion and future work

Synthesis

PHYLOG framework



- argumentation pattern per MCP-CRI objective
- PML (PHYLOG meta model)
- automatic computation with formal methods

- web site https://w3.onera.fr/phylog/
- open source results

Synthesis

Special thanks to EASA (Nicolas Chevillard, Guillaume Soudain) for fruitful discussions