Multiform requirements for critical systems : modeling, traceability, validation

Marie-Agnès Peraldi-Frati
MCF Informatique – UCA
EPI Kairos

Agenda

- Kairos research Domains : Critical embedded systems, IoT, Standardization,
- Special focus on requirements types
- Methodological needs
- DSL + multiform clocks + Formal models -> parts of our solution
- Summary & Roadmap

Application domains

How to capture, trace, validate, verify ?

- Electronic control module : **multiform time, CO2 emission, safety requirements ?**

- Autonomous vehicle : **time, space, safety, dependability requirements ?**

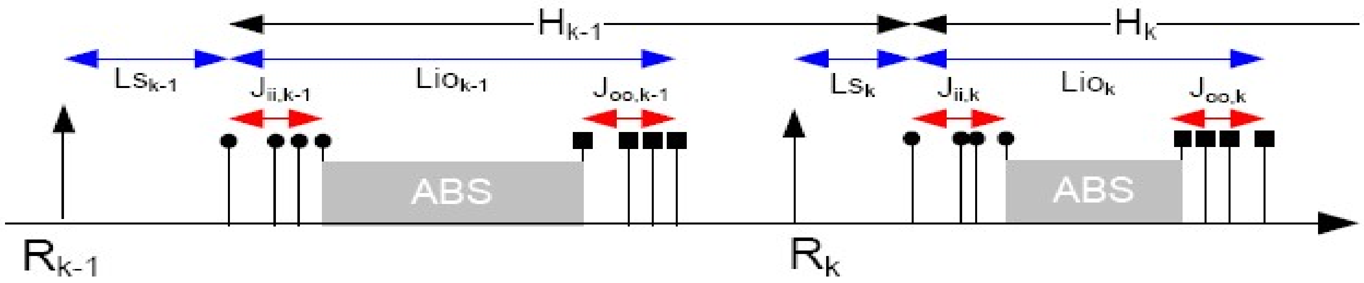
- IoT systems: **scalability, security ?**
- IoT standards : **consistency, compliance**

Agenda

- Critical embedded systems, IoT, Standardization,
- **Special focus on requirements types**
- Methodological needs
- DSL + multiform clocks + Formal models -> parts of our solution
- Summary & Roadmap

Example 1 : Temporal requirements in Anti-lock Braking System - ABS

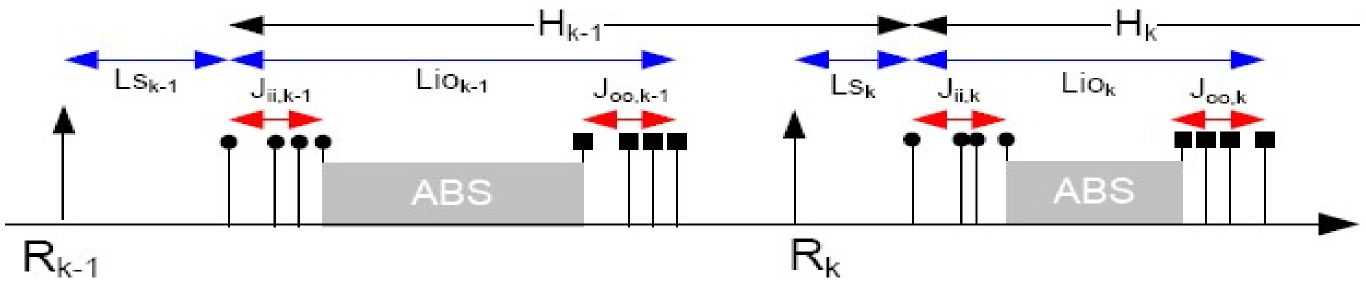
Projet ITEA Timmo2Use 2012



- **Sampling Interval** *H* : timing period for sensor sampling
- **Trigger period** *R* : the recurrence period of an ADL Function
- **Sampling Latency** *Ls*: sampling execution duration
- **Input Output Latency** *Lio*: timing duration between the first input of an ADL Function to the last output of the ADL Function.
- **Input / Output Synchronization** *Jii, Joo* : delay measured between the first event on a set of Input or output ports to the last event on this set.
- **Communication Latency** *Lc* : timing duration between an output port of an ADL Function and an input port of another ADL Function

Projet ITEA Timmo2Use 2012

Example 1 : Temporal requirements in Anti-lock Braking System - ABS



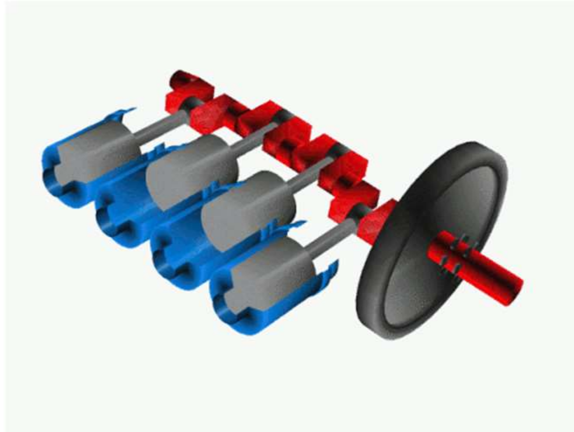
H : Sampling Interval
Ls : Sampling Latency
Lio : Input-Output Latency

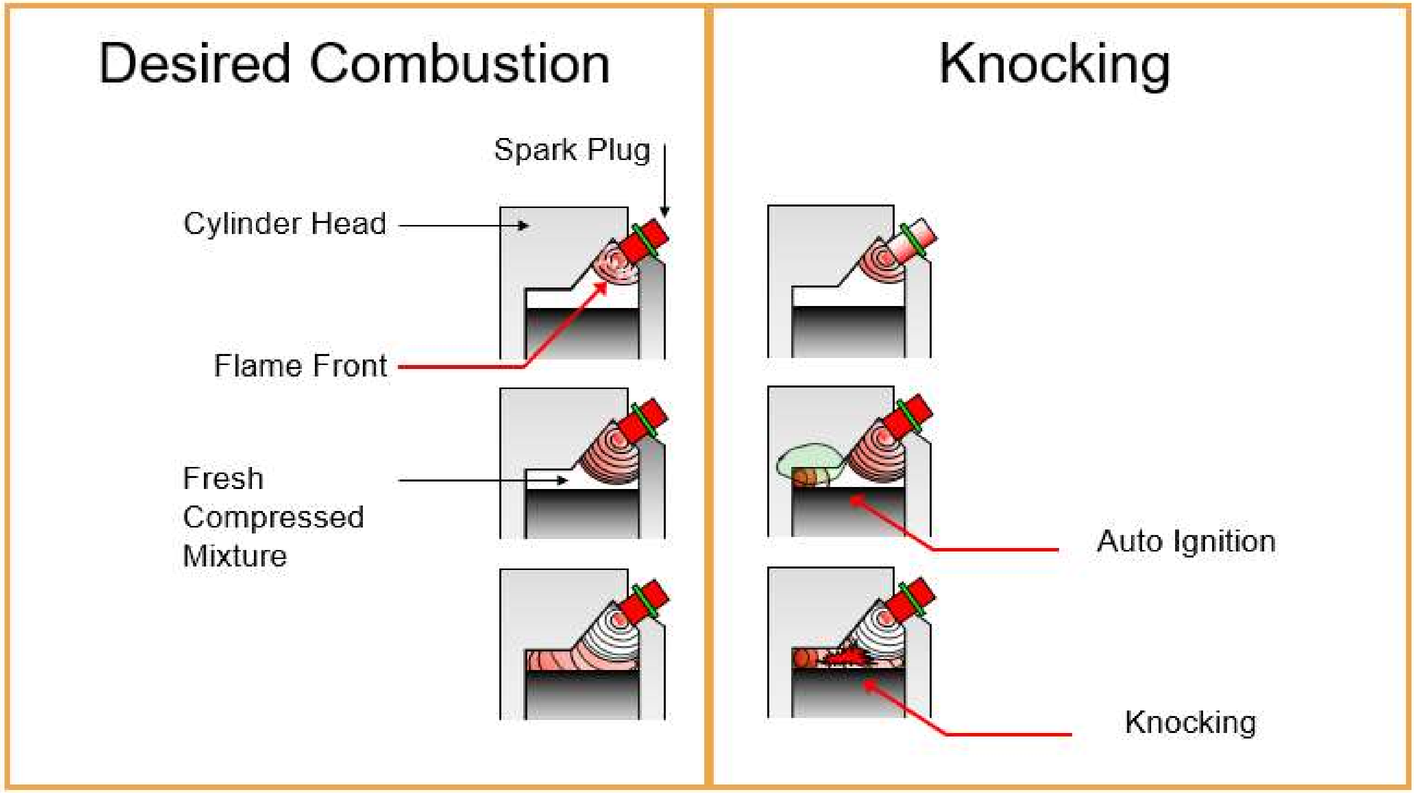
Jii : Input Synchronization
Joo : Output Synchronization
R : Trigger Period

Parameter	nominal	upper	lower	jitter
R	5 ms	-	-	1 ms
L_s	-	3 ms	-	2 ms
J_{oo}	-	0.5 ms	-	-
J_{ii}	-	0.5 ms	-	-
L_{io}	-	5 ms	-	2 ms
L_{ispeed}	-	5 ms	-	3 ms

Example2 : Safety requirements for engine control system

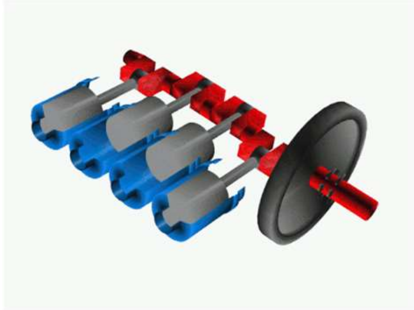
Projet ANR Memvatex 2009





Projet ANR Memvatex 2009

Example2 : Safety requirements for engine control system



Q : When to produce the spark to avoid misfiring and a potential cylinder damage ?

A: @ the top dead center instant (TDC)? Angular parameter corresponding to a piston position mesured in angular degre on the crankshaft

Related requirements :

- If a knock event is detected, an instantaneous Spark delay correction corresponding to this knock event shall be ensured.
- The knock correction value should be within the interval [-15°,+15°] CRK .
- A total spark retard shall be calculated by the correction sub function in case of knock detected
- The acquisition of the knock signal shall be performed in 50 ms

Safety analysis -> safety requirements

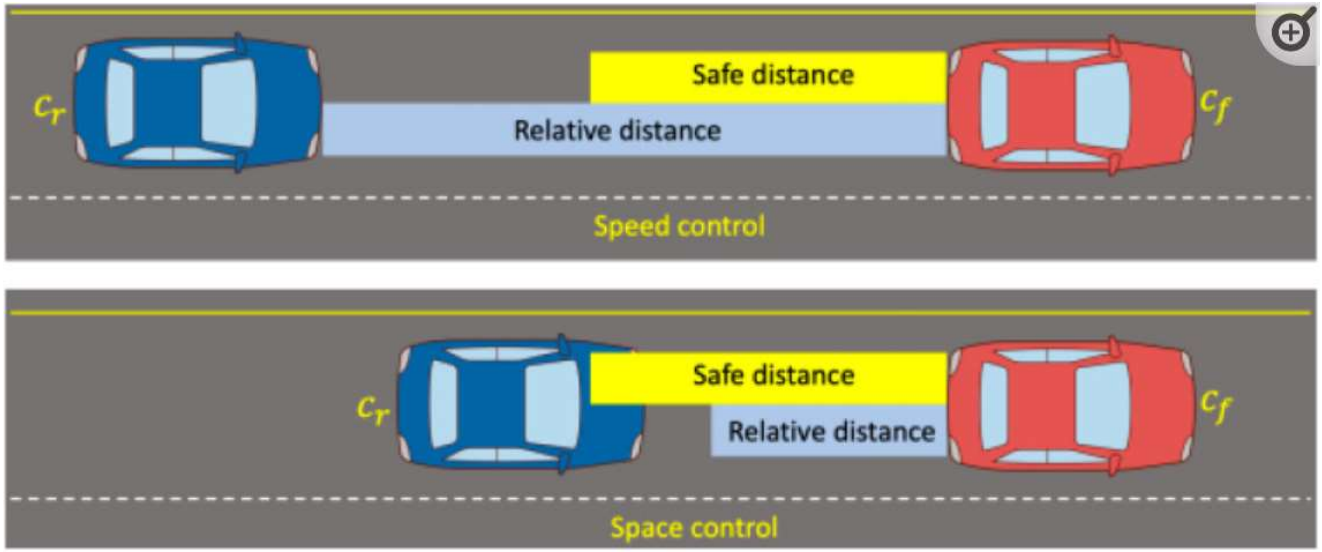
- FMEA : analyze and classify by severity level possible failures, their effects and causes

Function	Ignition Control System
Mode	Normal
Failure mode	Shockwaves in the cylinders
Effect	Cylinder destruction
Severity	10
Cause 1 **Cause 2**	No instantaneous spark delay correction Wrong spark delay correction

Example 3 : Spatial & timing requirements
Automatic Preventive Braking
- autonomous vehicle

Thèse Cifre Renault 2022
& Thèse Maksym
Labhzaniia 2022-2025

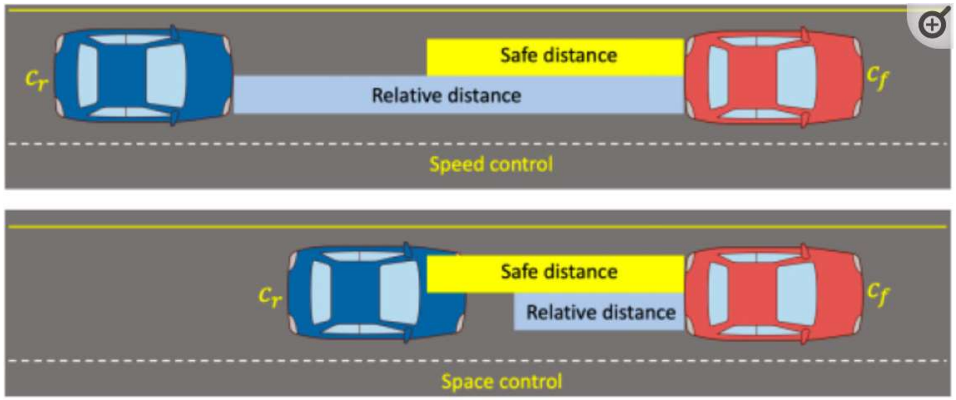
- Autonomous vehicles also introduce new safety risks, such as potential technology malfunctions that can cause car accidents



Example 3 : Spatial & timing requirements Automatic Preventive Braking - autonomous vehicle

Thèse Cifre Renault 2022
& Thèse Maksym
Labhzaniia 2022-2025

Rule 1 Safe Distance* : Une distance longitudinale entre une voiture qui roule derrière une autre voiture, où les deux voitures roulent dans la même direction, est sûre par rapport à un temps de réponse ρ si, pour tout freinage d'au plus amax,brake, effectué par cf, si cr accélère d'au plus amax.accel pendant le temps de réponse et, à partir de là, freine d'au moins amin,brake jusqu'à l'arrêt complet, alors il n'entre pas en collision avec cf.



* Kim MJ, Yu SH, Kim TH, Kim JU, Kim YM. On the Development of Autonomous Vehicle Safety Distance by an RSS Model Based on a Variable Focus Function Camera. Sensors (Basel). 2021 Oct 11;21(20):6733. doi: 10.3390/s21206733. PMID: 34695946; PMCID: PMC8539732.

oneM2M
The IoT Standard

AUTOSAR

EAST-ADL Automotive

ISO 26262-1:2018
Véhicules routiers — Sécurité fonctionnelle

Actions standardisation Kairos

Example 4 : Conficts & Consistency of requirements - Standards

A standard specifies requirements.

System implementing a standard has to met these requirements.

Standards specifications :

- Requirements should be **clearly identified** as mandatory or optional;
- **Means of conforming** to the standard should be clearly identified; and
- Definitions and requirements should **not conflict with each other**.

Potential issues :

- Requirements that are **incomplete** or omitted from the specification
- Requirements that **conflict with other** requirements in the same standard
- **Incorrect semantics** used in a specification language
- Requirements that specify unnecessary restrictions on the implementation of the standard (over-engineering)

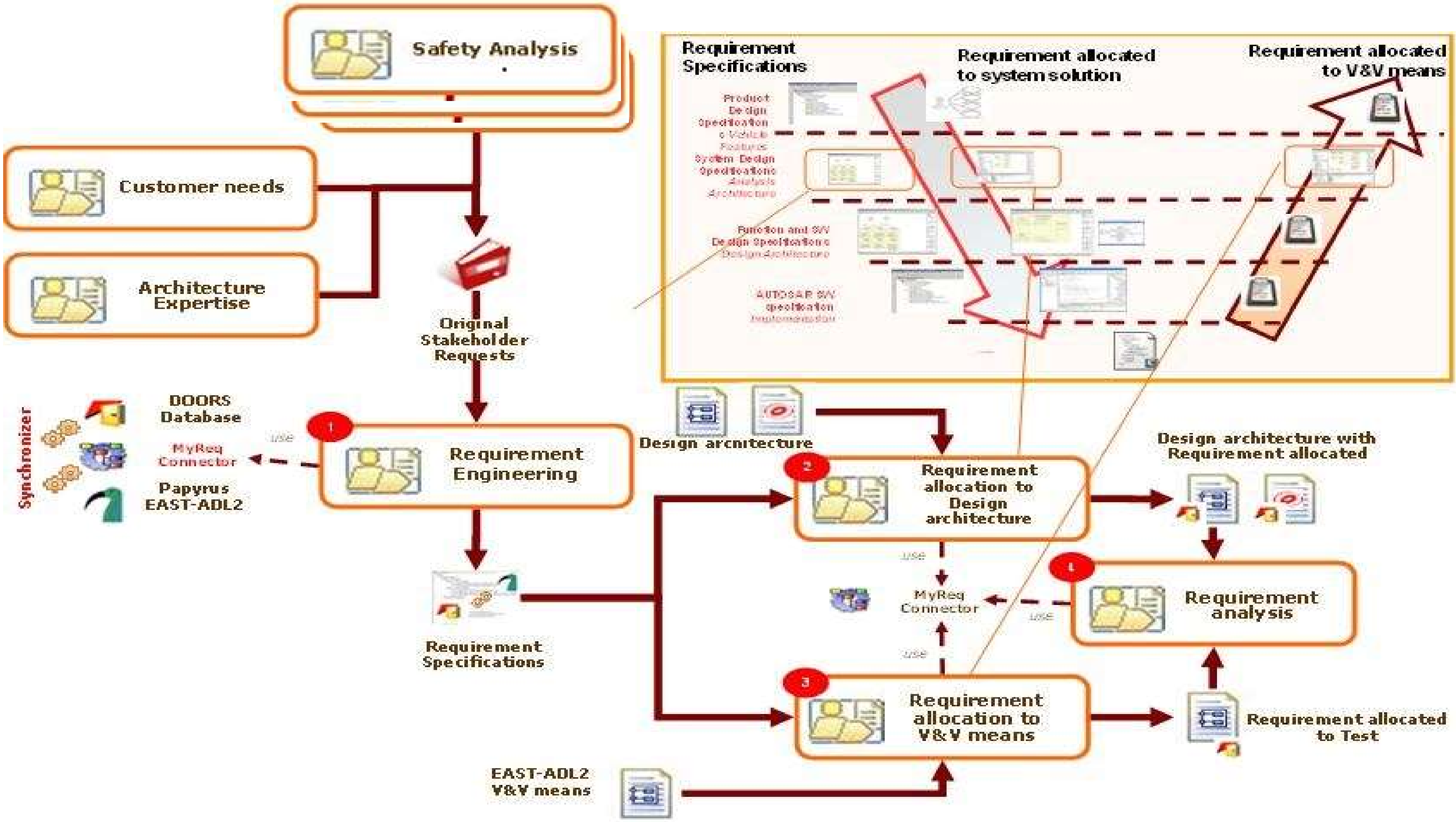
RÉPUBLIQUE FRANÇAISE

UNIVERSITÉ CÔTE D'AZUR

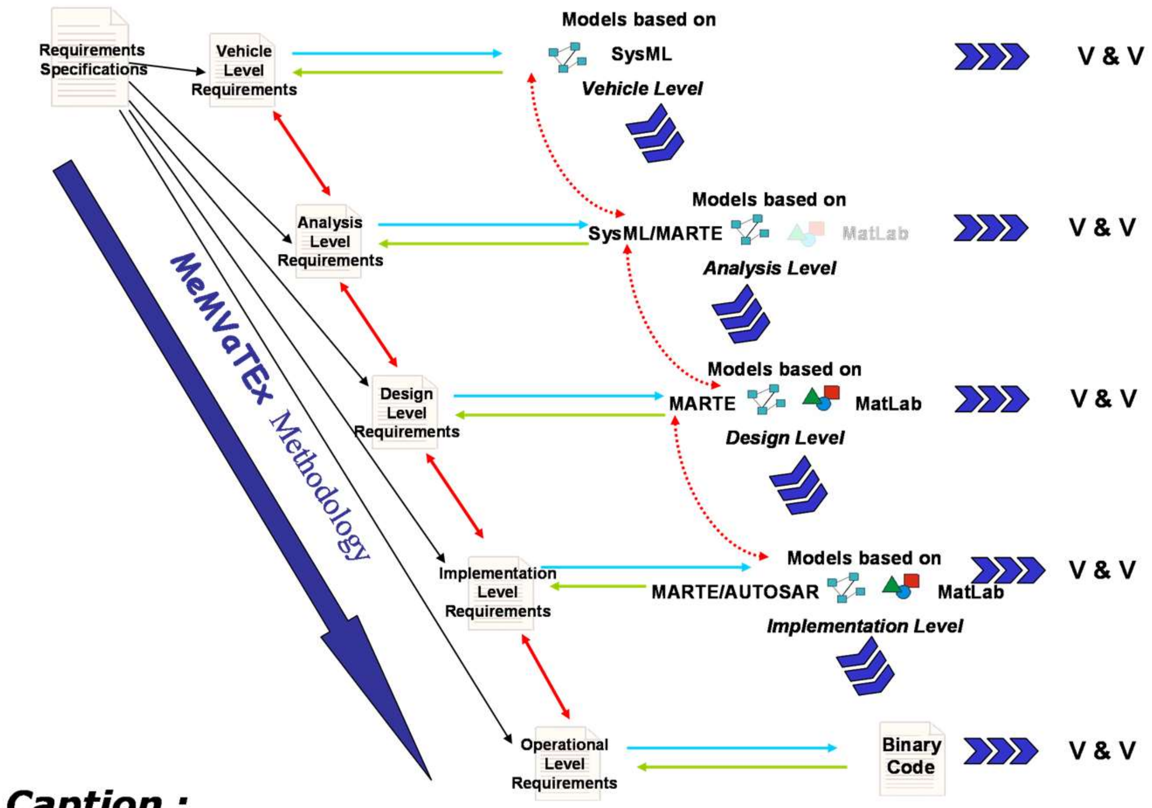
Agenda

- Critical embedded systems, IoT, Standardization,
- Special focus on requirements types
- **Methodological needs**
- DSL + multiform clocks + Formal models -> parts of our solution
- Summary & Roadmap

Process flow for collecting requirements



Design Methodology



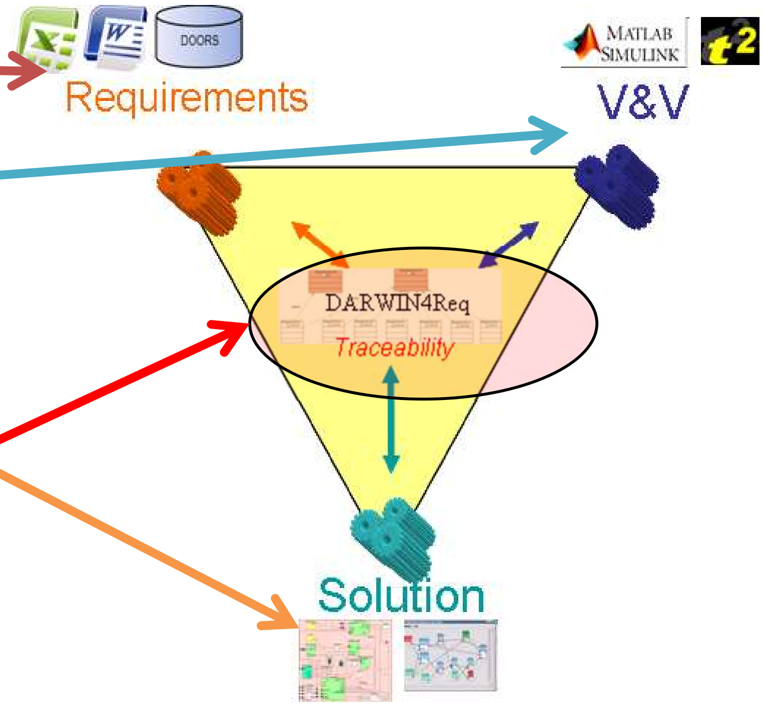
Caption :

- Consideration of requirements at models or code
- Return on the requirements after modeling and V&V
- Traceability of the requirements
- Follow-up of the requirements by models translation
- Model transformation
- **V & V** Verification & Validation

Design Methodology

- **A design methodology** :
  - Requirement modeling
  - Design solution
  - V&V analysis
- **Separation of concerns :**

- **Traceability model as a glue**
  - UML based metamodel
  - defines the semantics of links
  - Compatible with SysML MARTE profiles
  - Integrates heterogeneous DSL (EAST_ADL, Autosar) & tools (Simulink, Syndex, Timesquare …)

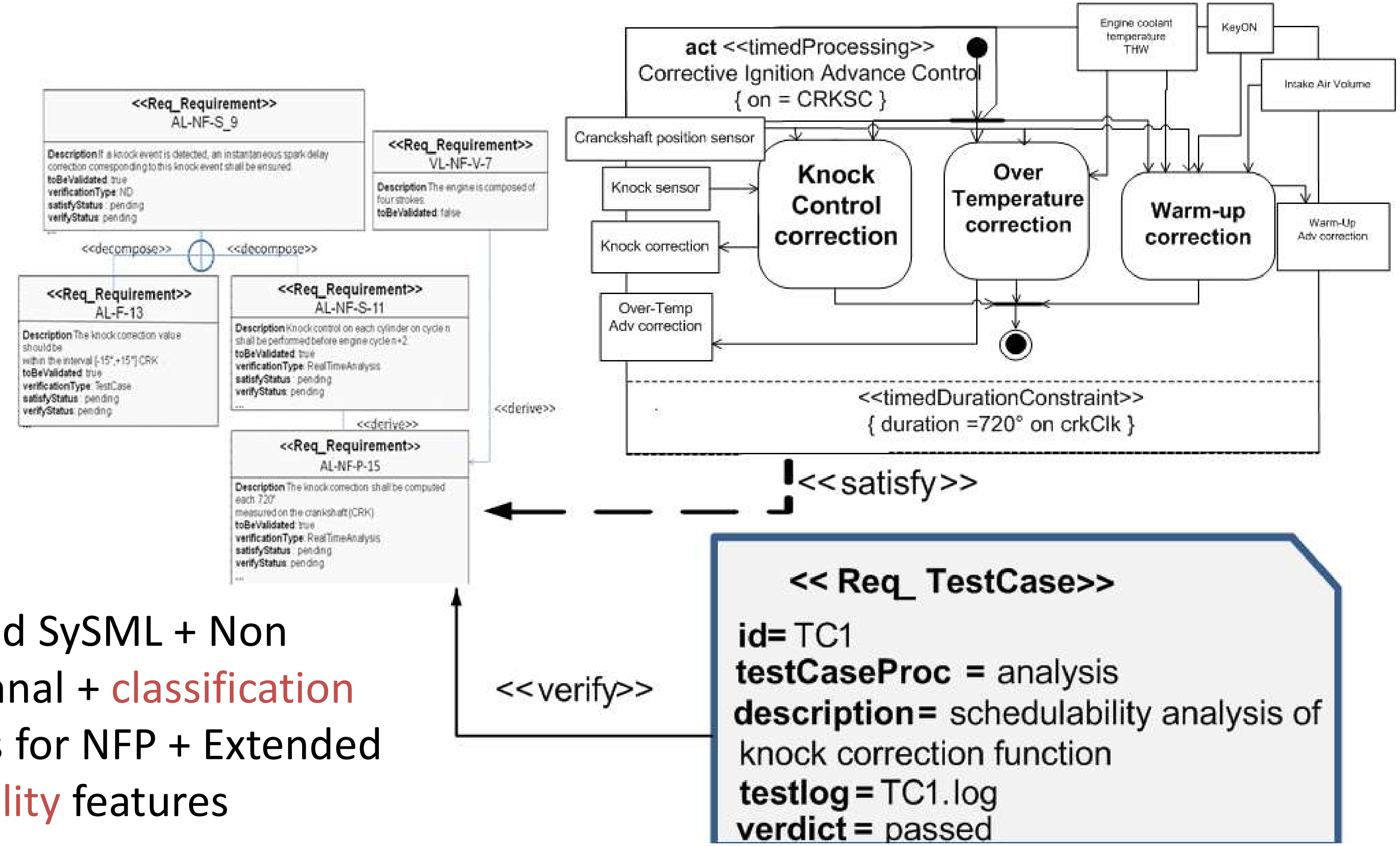


Traceability Definition

Link between Solution model, V&V model and Requirement model for the knock

Requirement model

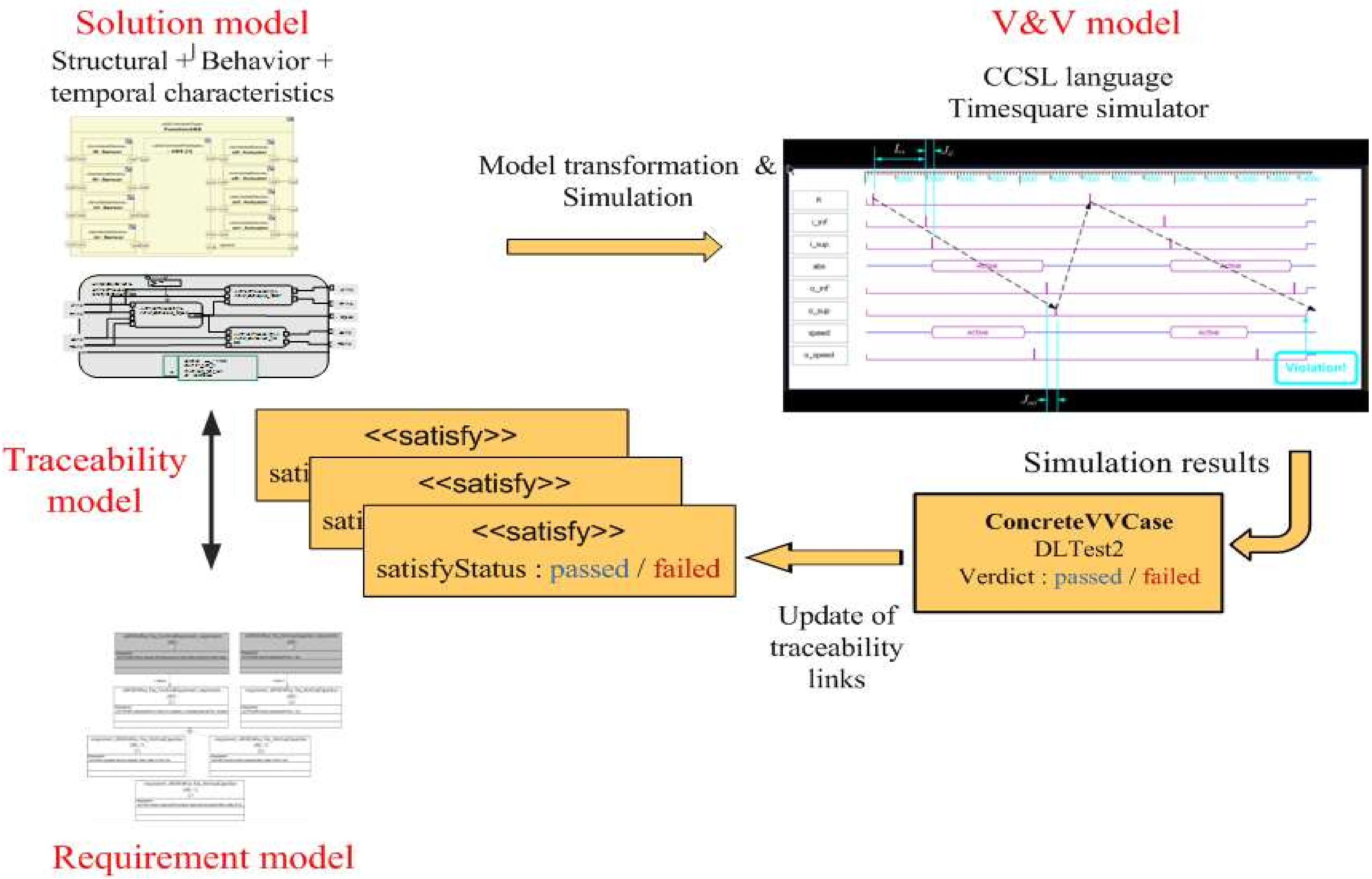
Solution model



V&V model

Extended SysML + Non functionnal + classification features for NFP + Extended traceability features for V&V

Focus on validation



RÉPUBLIQUE FRANÇAISE

Published in ECMDA IES, ERTS, ISORC, RTSS, MODRE cf. https://webusers.i3s.unice.fr/~map/publications.FR.html

UNIVERSITÉ CÔTE D'AZUR

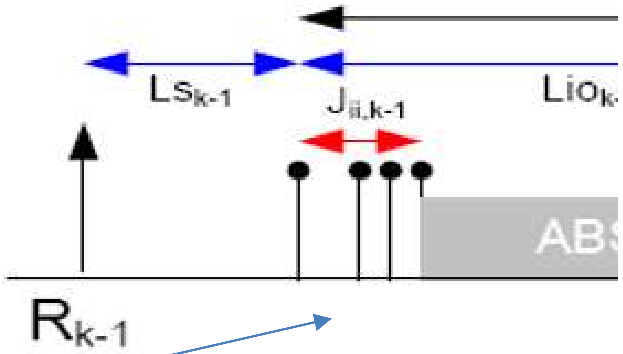
Agenda

- Critical embedded systems, IoT, Standardization,
- Special focus on requirements types
- Methodological needs
- **DSL + multiform clocks + Formal models -> parts of the solution & V&V means**
- Summary & Roadmap

RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

UNIVERSITÉ CÔTE D'AZUR

DSL example : Time Augmented Description Language




```
SynchronizationConstraint tc10 {
  events firstWheelBrakeActuation,
   secondWheelBrakeActuation, thirdWheelBrakeActuation,
   fourthWheelBrakeActuation
   tolerance = (5.0 ms on universal_time) }
```

Formal model of TADL specifications

TADL Clocks based/event-based specifications

```
SynchronizationConstraint tc10 {
  events firstWheelBrakeActuation ,
   secondWheelBrakeActuation , thirdWheelBrakeActuation ,
   fourthWheelBrakeActuation
   tolerance = (5.0 ms on universal_time)  }
```

Associated CCSL Clock Constraint Specification

$$
\begin{aligned}
fastest\_tc10 \; = \; & \mathsf{Inf}(firstWheelBrakeActuation, \\
& secondWheelBrakeActuation, \\
& thirdWheelBrakeActuation, \\
& fourthWheelBrakeActuation) \\
slowest\_tc10 \; = \; & \mathsf{Sup}(firstWheelBrakeActuation, \\
& secondWheelBrakeActuation, \\
& thirdWheelBrakeActuation, \\
& fourthWheelBrakeActuation)
\end{aligned}
$$

$\mathsf{Integer}\ tolerance\ \ tc10 = 5000$

Filling the gap between requirements and solution models

Example of oneM2M IoT systems

Model Driven Engineering approach for Rapid prototyping of embedded systems and applications

- ***Development of a DSL (Domain Specific Language)*** *for oneM2M to define:*
  - High level modeling of IoT application -> the logical infrastructure of the system
  - Compliance with oneM2M Standard -> The oneM2M nodes & services (CRUD)
  - Performance Evaluation -> object and communications behaviors
  - Scalable modeling -> programmatic definition of the infrastructure
- ***Evaluation of a oneM2M system By simulation***
  - Automatic Generation of executable models (Omnet++ simulator)
  - Application behavior (periodic sporadic calls to services)
  - Communication latencies
  - Effective Performances of oneM2M platform implementations

oneM2M
executable specifications

DSL expressivness coverage

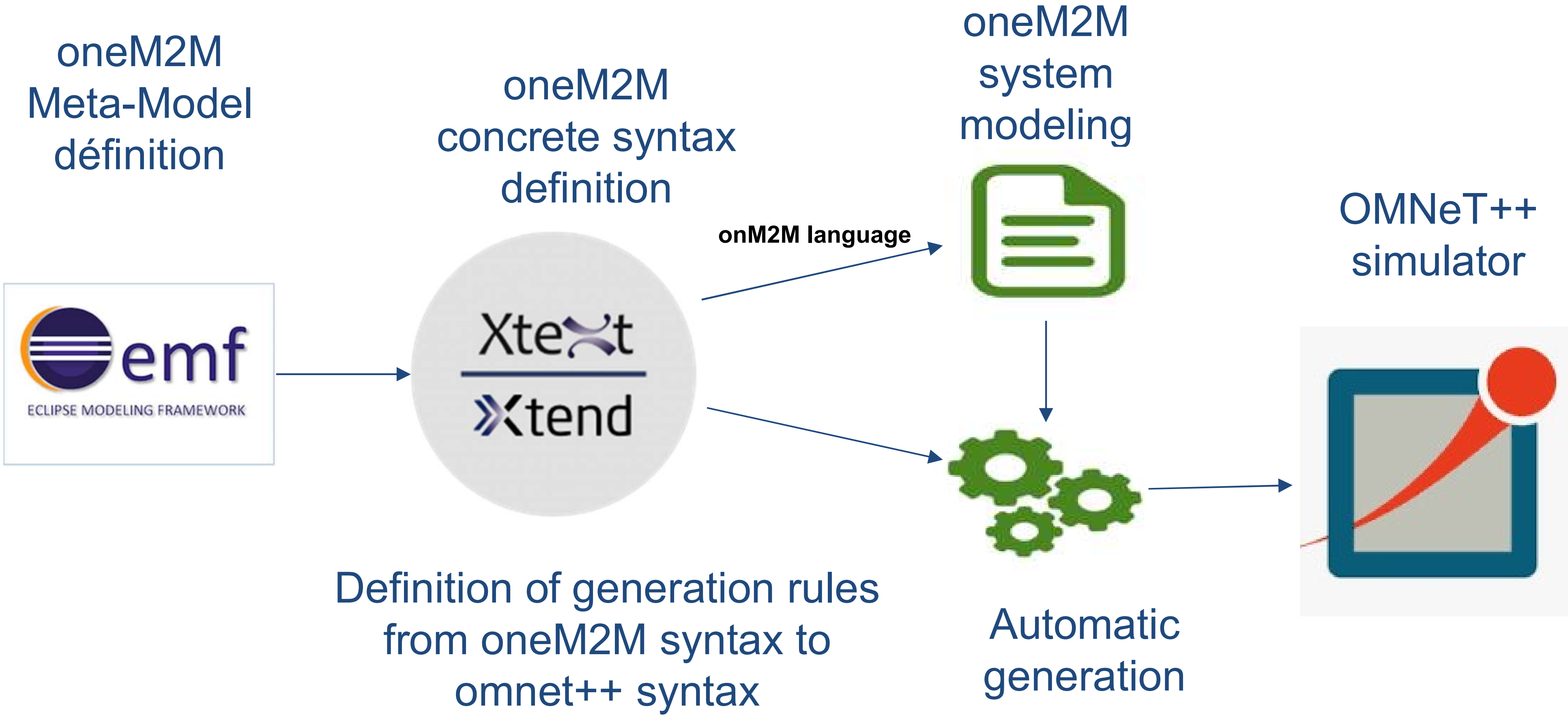
IoT system implementation

- Effective IoT system infrastructure
  - Decentralized nodes
  - Gateways (edge computing, semi-local constraints)
  - Server on the cloud
  - Applications : apps, servers, objects

- Heterogeneous requirements on application
  - Sensing and actioning on real-time information / physical environment
  - Embedded sensing, computing,
  - Decentralized and static/mobile services
  - Data integrations over different environments

- Heterogeneous requirements on deployment
  - functionnal : functional description, Privacy and Security Challenge
  - Non-functionnal : power consumption, time , memory, processing resources

OneM2M Domain Specific Language definition & code generation

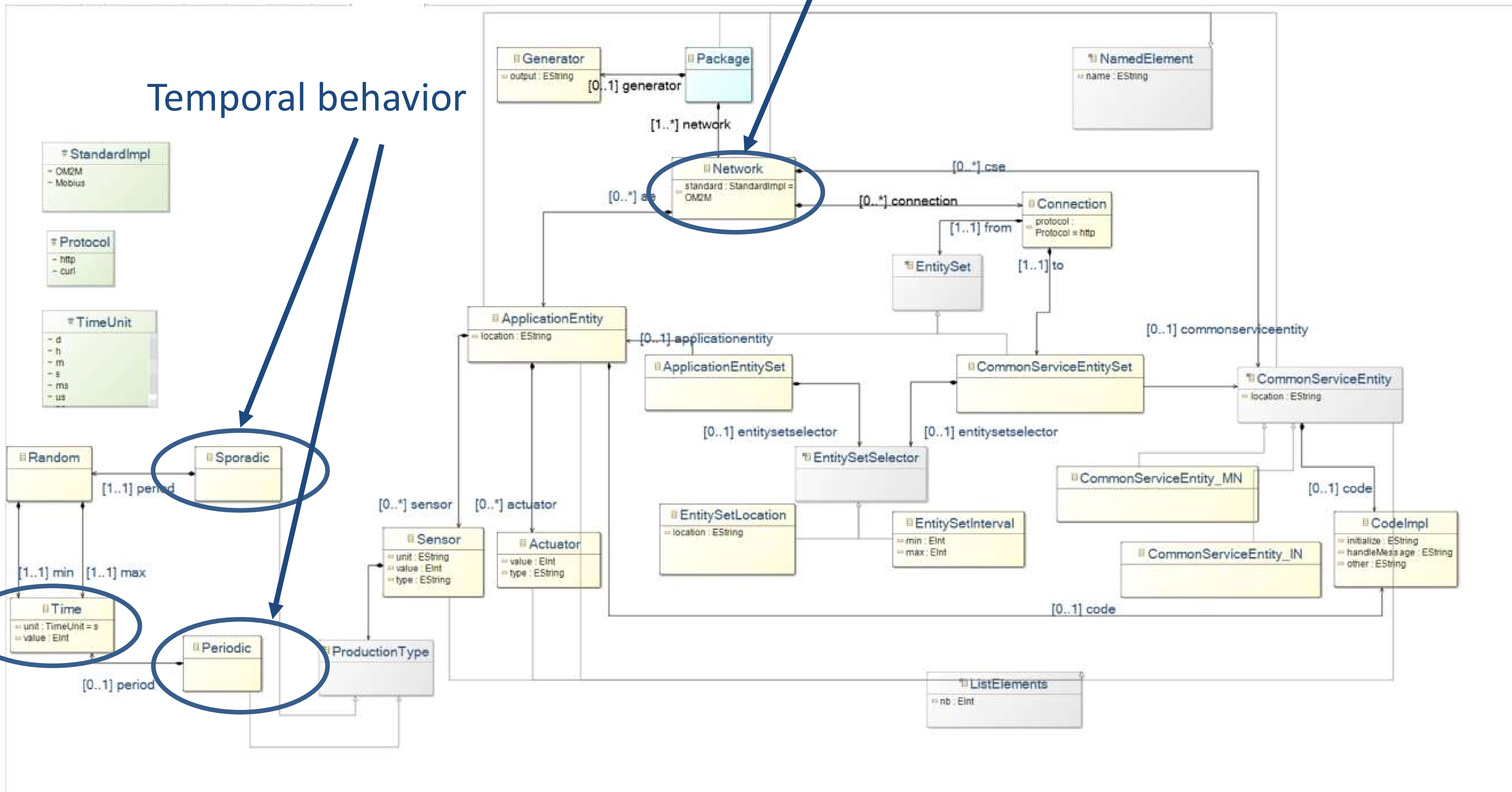


OneM2M Domain Specific Language

emf
ECLIPSE MODELING FRAMEWORK

oneM2M
Meta-Model
Application Behaviour

Deployment platform



RÉPUBLIQUE FRANÇAISE

UNIVERSITÉ CÔTE D'AZUR

OneM2M Domain Specific Language
System specification

oneM2M
concrete syntax
definition

Xtext
Xtend

```
Package exemple {

    Network Petit {
        standard Mobius

        ApplicationEntity ae1 {
            Sensor [2] {
                unit "%"
                value 5
                production Sporadic {
                    period 1us..10ms
                }
            }
        }

        CommonServiceEntity_IN cin {
            location "INRIA"
        }

        Connection {
            protocol curl
            from ae ae1
            to cse cin
        }
    }

}
```

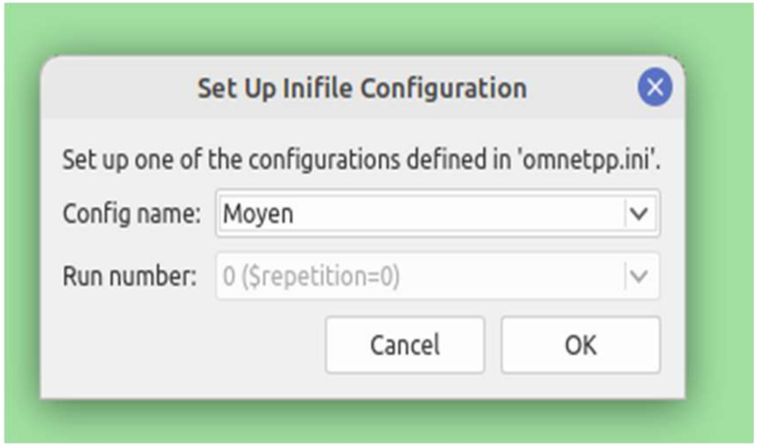
OneM2M Simulation in OMNeT++

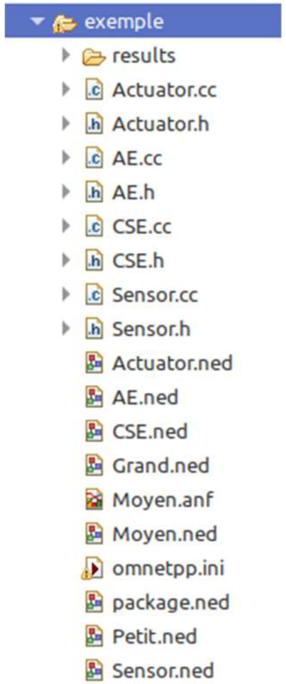
oneM2M system modeling

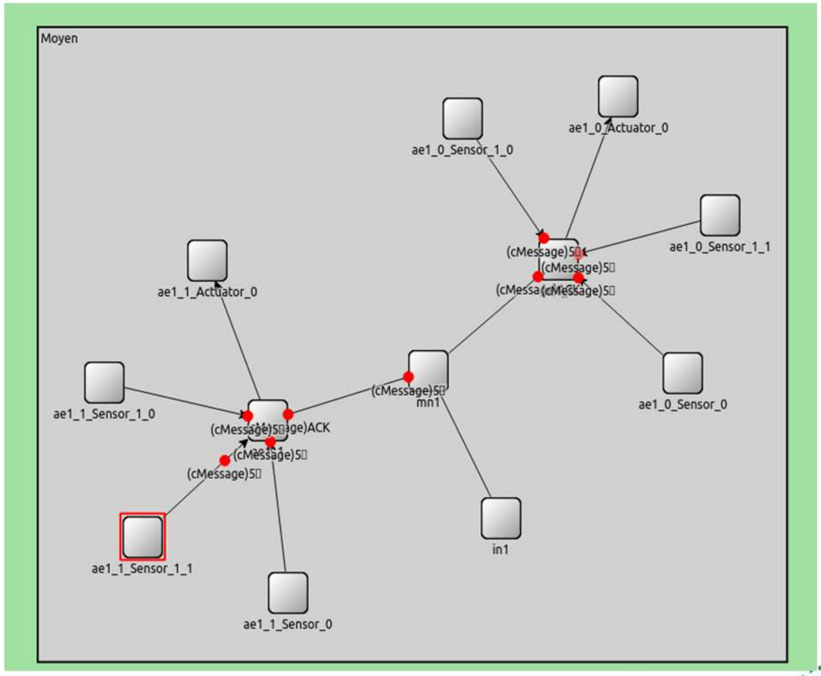
Automatic generation

Generation of Ned and Ini and files for OMNeT++

```
Package exemple {

    Network Petit {
        standard Mobius

        ApplicationEntity ae1 {
            Sensor [2] {
                unit "%"
                value 5
                production Sporadic {
                    period 1us..10ms
                }
            }
        }

        CommonServiceEntity_IN cin {
            location "INRIA"
        }

        Connection {
            protocol curl
            from ae ae1
            to cse cin
        }
    }

}
```





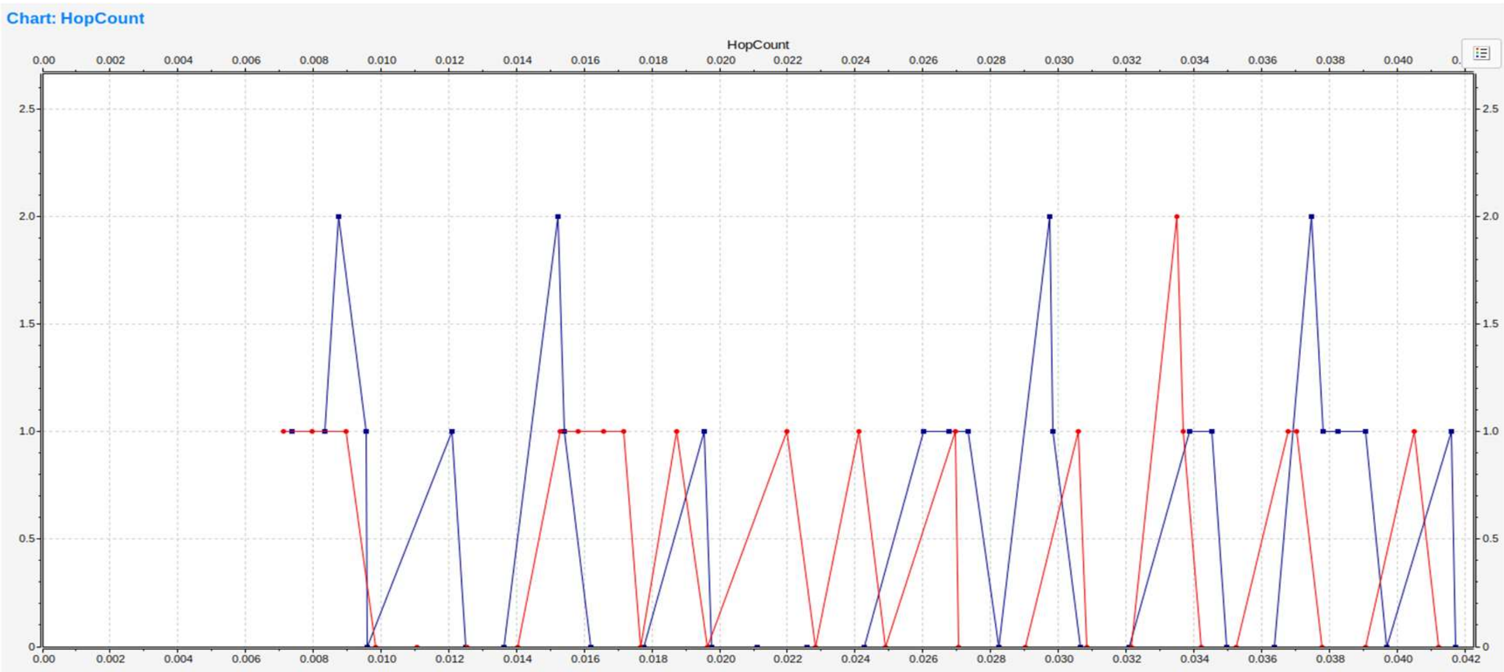


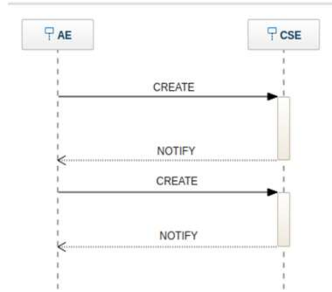
RÉPUBLIQUE FRANÇAISE

UNIVERSITÉ CÔTE D'AZUR

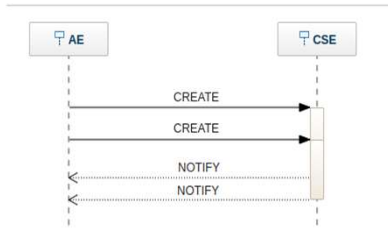
Simulation in OMNeT++ (2)

- Execution traces and scenario validation





✓



✗

RÉPUBLIQUE FRANÇAISE

UNIVERSITÉ CÔTE D'AZUR

Agenda

- Critical embedded systems, IoT, Standardization,
- Special focus on requirements types
- Methodological needs
- DSL + multiform clocks + Formal models -> parts of the solution & V&V means
- **Summary & Roadmap**

Summary

- Requirement egineering for CPS & IoT systems
- Traceability aspects *Req<->Solution<-> V&V*
- High level validation/verification of requirements
- DSL + multiform/logical clocks + formal models parts of the solution
- Example on oneM2M IoT system evaluation

Next Steps

- Agility for a high-level validation of requirements at every stage of the Development cycle
- Integrated feedback from validation to requirements
- Maintain the focus on multiform requirements ++

Thank you for your attention

Questions ?