



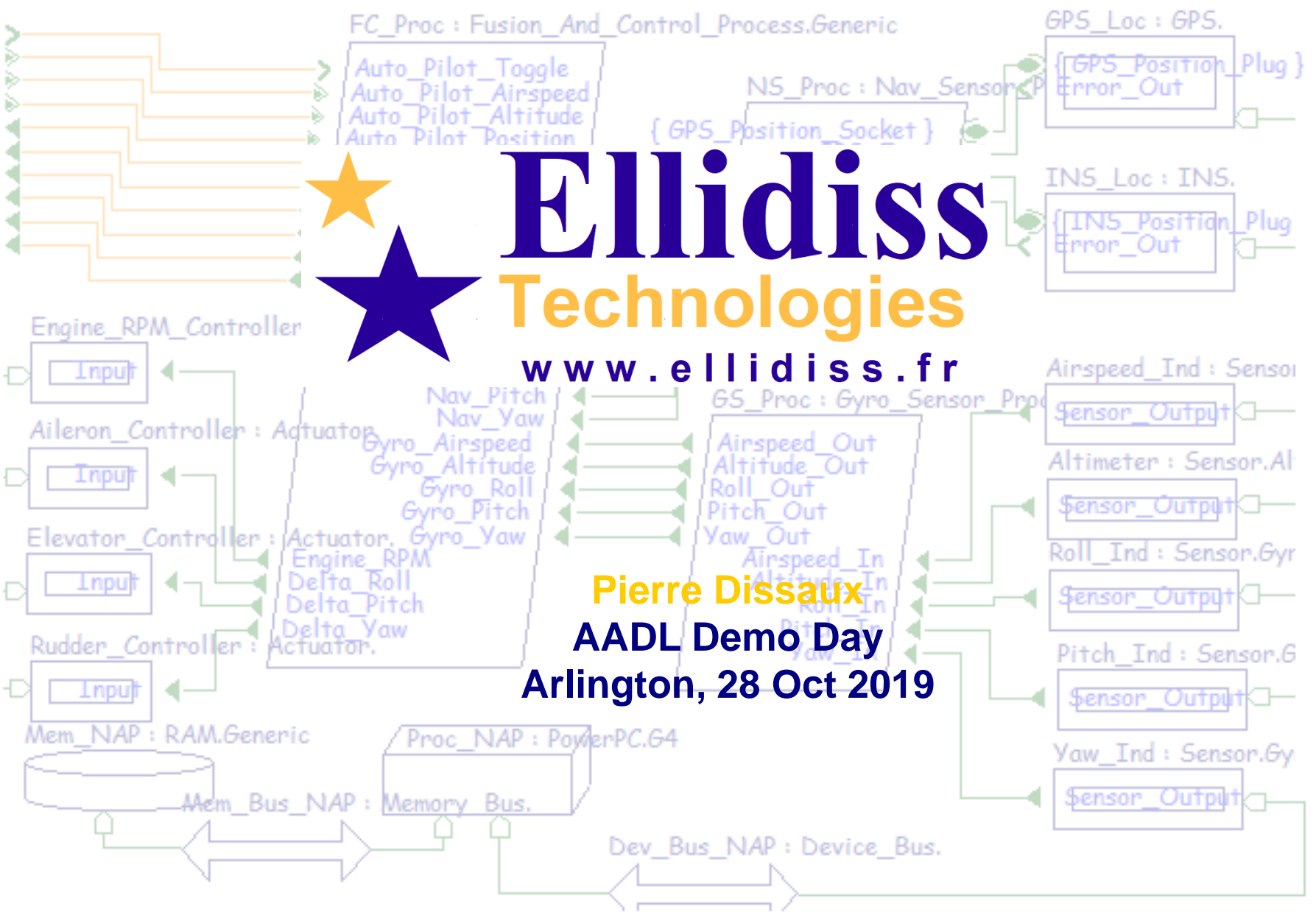
# Ellidiss Technologies

[www.ellidiss.fr](http://www.ellidiss.fr)

Pierre Dissaux

AADL Demo Day

Arlington, 28 Oct 2019



## Critical Software Design tools (HOOD)



*Eurofighter Typhoon*



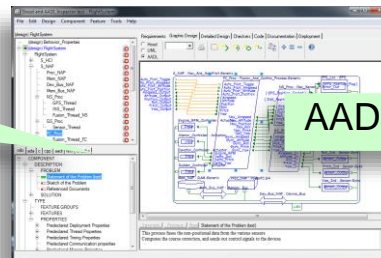
*Tiger*



*Airbus A350*

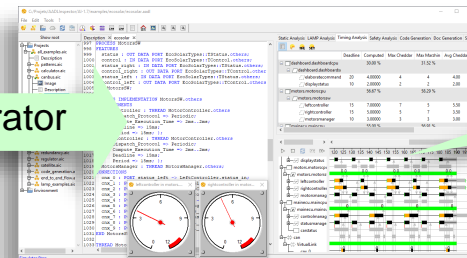
## Real-Time, Safety and Security modeling and analysis tools (AADL)

software  
design



*Stood for AADL*

AADL generator

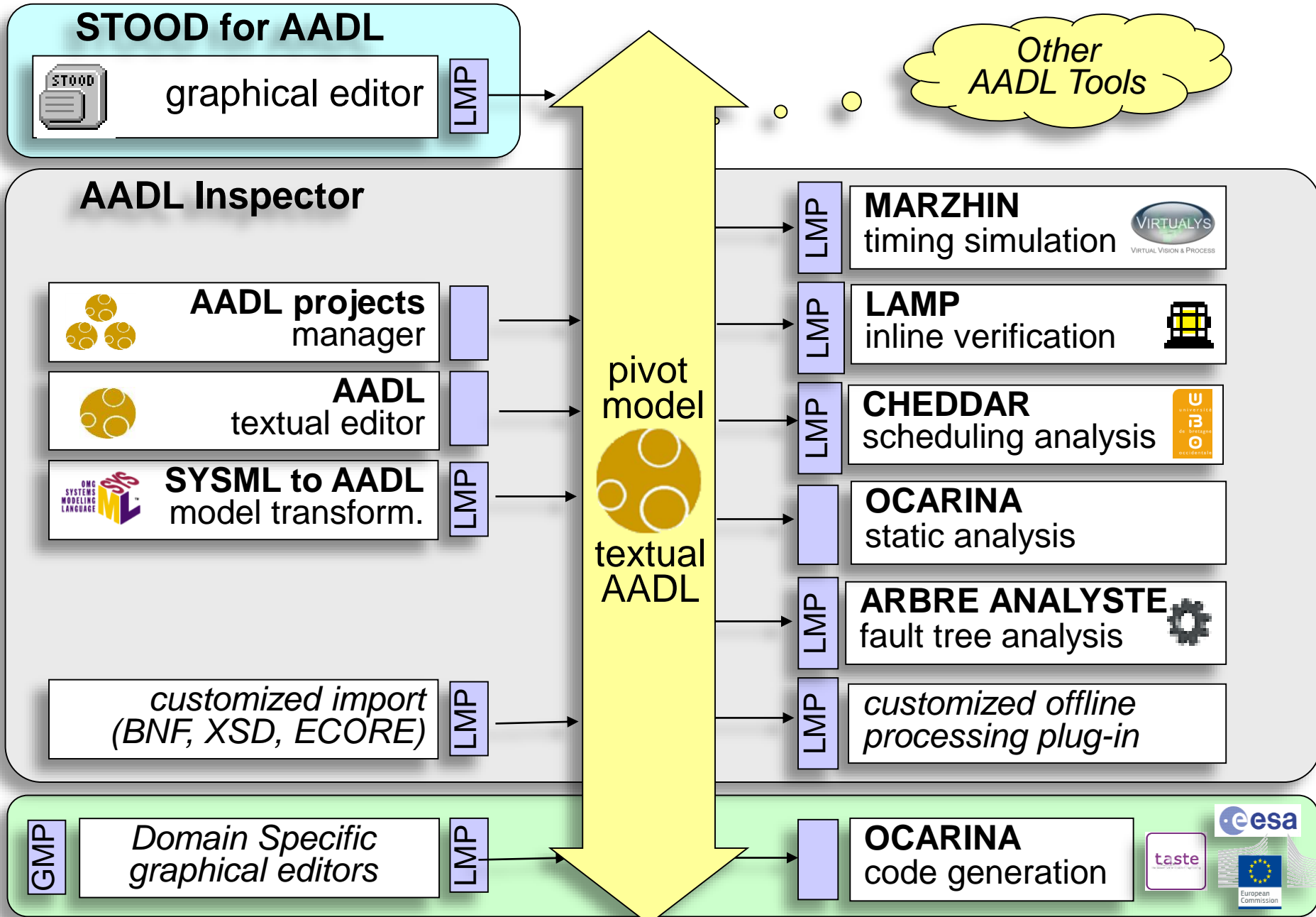


*AADL Inspector*

early  
verification

## Modeling tools and model processing technologies (GMP, LMP)

# AADL centric tool-chains



# Stand for AADL key features

requirements coverage

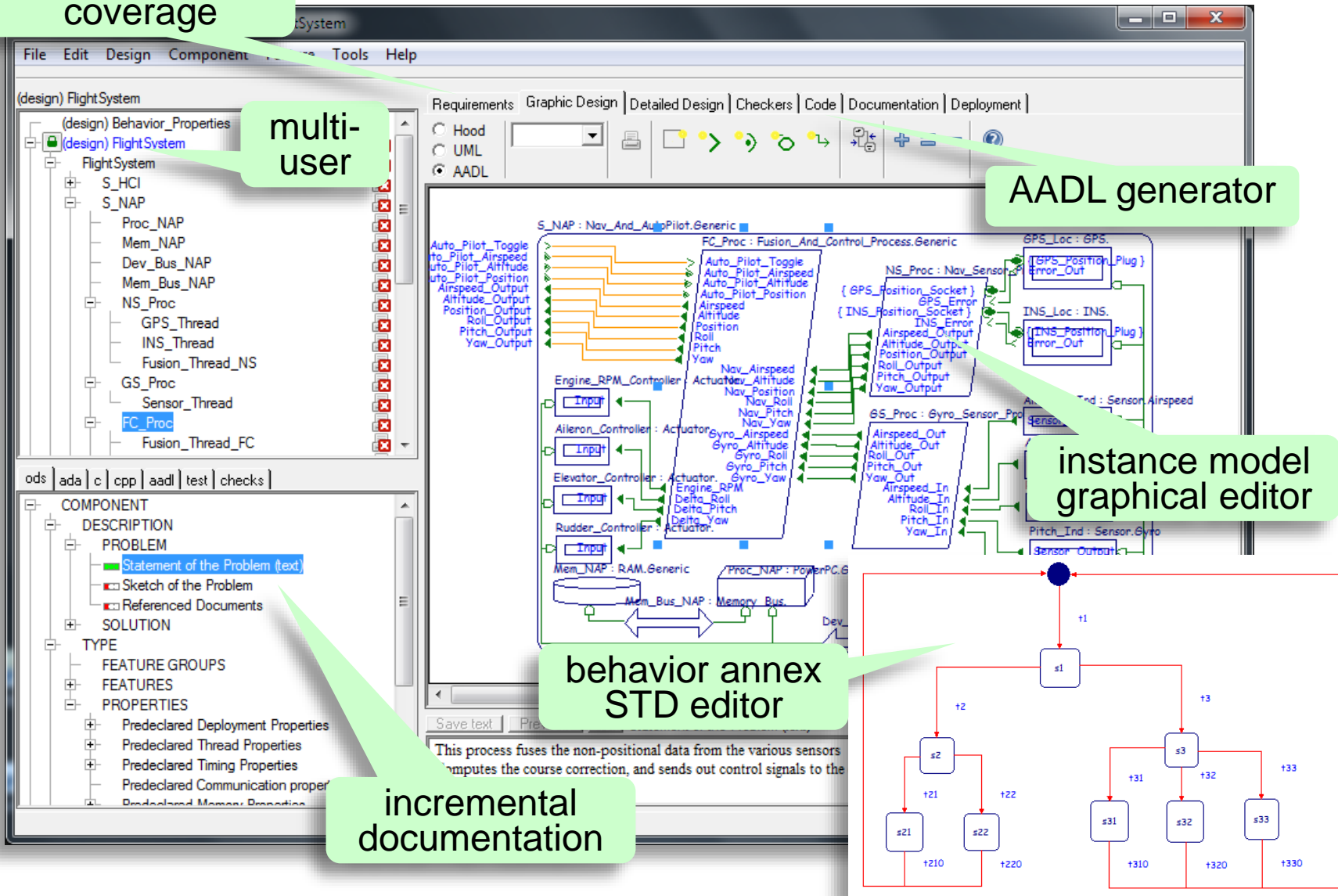
multi-user

AADL generator

instance model graphical editor

behavior annex STD editor

incremental documentation



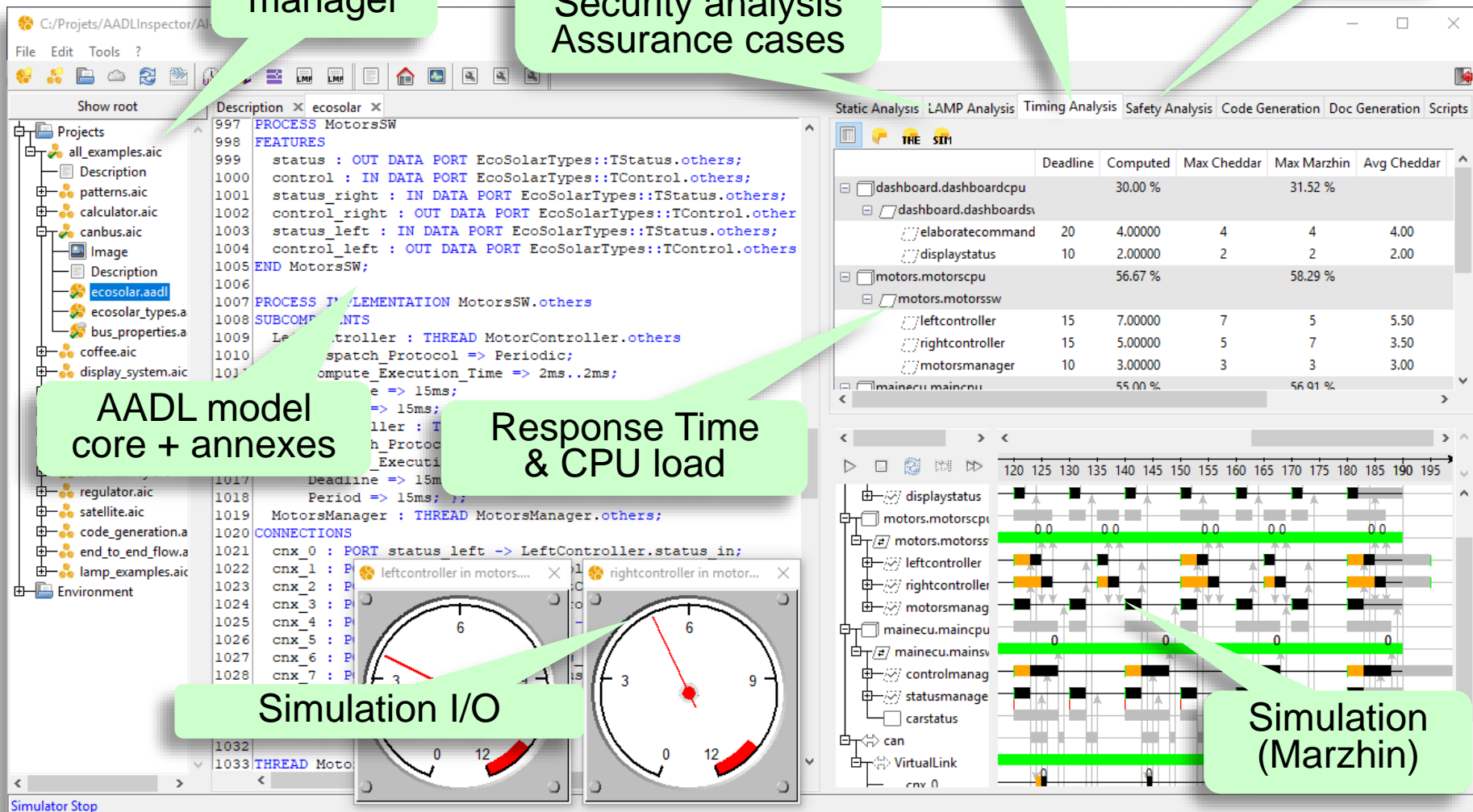
# AADL Inspector key features

Projects manager

LAMP:  
Flow analysis  
Security analysis  
Assurance cases

Scheduling analysis

Safety analysis

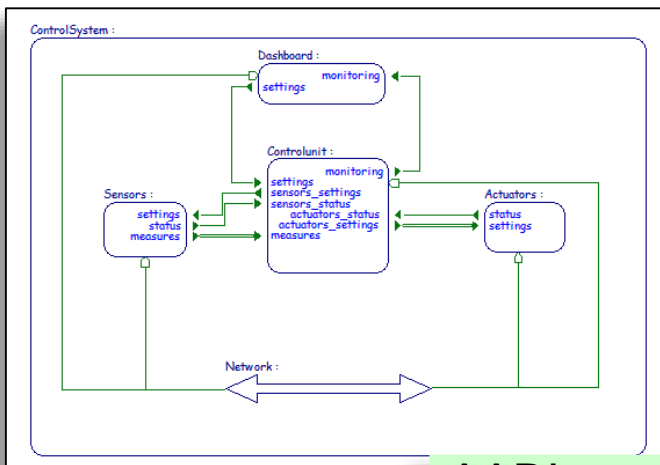


The screenshot displays the AADL Inspector interface with several key features highlighted by callouts:

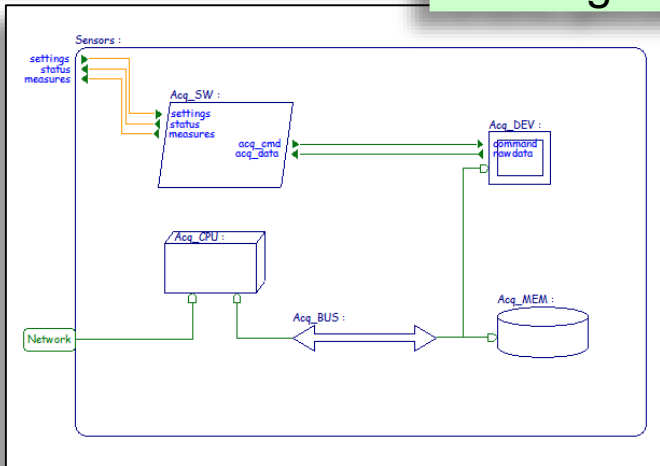
- Projects manager:** A tree view on the left shows a project structure including 'all\_examples.aic', 'patterns.aic', 'calculator.aic', 'canbus.aic', 'Image', 'ecosolar.aadl', 'ecosolar\_types.a', 'bus\_properties.a', 'coffee.aic', and 'display\_system.aic'.
- AADL model core + annexes:** The central pane shows AADL code for 'PROCESS MotorsSW' and 'PROCESS IMPLEMENTATION MotorsSW.others', including status and control ports.
- Response Time & CPU load:** A table in the 'Timing Analysis' tab provides performance metrics for various components.
- Simulation I/O:** Two circular gauges at the bottom show simulation input/output values.
- Simulation (Marzhin):** A Gantt chart at the bottom right visualizes the execution timeline of components like 'displaystatus', 'motors.motorscpu', and 'mainecu.maincpu'.

Component	Deadline	Computed	Max Cheddar	Max Marzhin	Avg Cheddar
dashboard.dashboardcpu		30.00 %		31.52 %	
elaboratecommand	20	4.00000	4	4	4.00
displaystatus	10	2.00000	2	2	2.00
motors.motorscpu		56.67 %		58.29 %	
leftcontroller	15	7.00000	7	5	5.50
rightcontroller	15	5.00000	5	7	3.50
motorsmanager	10	3.00000	3	3	3.00
mainecu.maincpu		55.00 %		56.91 %	

## AADL modeling with Stood



AADL generator



**SYSTEM IMPLEMENTATION** ControlSystem.others

**SUBCOMPONENTS**

```

Sensors:      SYSTEM Sensors.others;
Controlunit:  SYSTEM Controlunit.others;
Actuators:    SYSTEM Actuators.others;
Dashboard:    SYSTEM Dashboard.others;
Network:      BUS Network;
  
```

**CONNECTIONS**

```

cnx1:  PORT Dashboard.settings -> ...
cnx2:  PORT Controlunit.monitoring -> ...
cnx3:  PORT Controlunit.sensors_settings -> ...
cnx4:  PORT Sensors.status -> ...
cnx5:  PORT Sensors.measures -> ...
cnx6:  PORT Controlunit.actuators_settings -> ...
cnx7:  PORT Actuators.status -> ...
cnx8:  BUS ACCESS Network -> Dashboard.Nwk;
cnx9:  BUS ACCESS Network -> Sensors.Nwk;
cnx10: BUS ACCESS Network -> Actuators.Nwk;
cnx11: BUS ACCESS Network -> Controlunit.Nwk;
  
```

**FLOWS**

f1: **END TO END FLOW**

```
Sensors.f1 -> cnx5 -> Controlunit.f1 -> cnx6 -> Actuators.f1;
```

**PROPERTIES**

```

Connection_Binding => (reference(Network))
  applies to cnx1, cnx2, cnx3, cnx4, cnx5, cnx6, cnx7;

Timing => Immediate
  applies to cnx5, cnx6;
  
```

**ANNEX EMV2** {\*\*

```

use behavior errorlibrary::failstop;
composite error behavior
states
  [ Dashboard.FailStop or
    Sensors.FailStop or
    ControlUnit.FailStop or
    Actuators.FailStop or
    Network.FailStop ]-> FailStop;
end composite;
**};
END ControlSystem.others;
  
```

Real Time

Safety

Security

**PACKAGE** ControlSystemTypes  
**PUBLIC**

**DATA** T\_measures

**PROPERTIES**

LAMP::Security\_Level => 5;

**END** T\_measures;

**DATA** T\_monitoring

**PROPERTIES**

LAMP::Security\_Level => 2;

**END** T\_monitoring;

-- ...

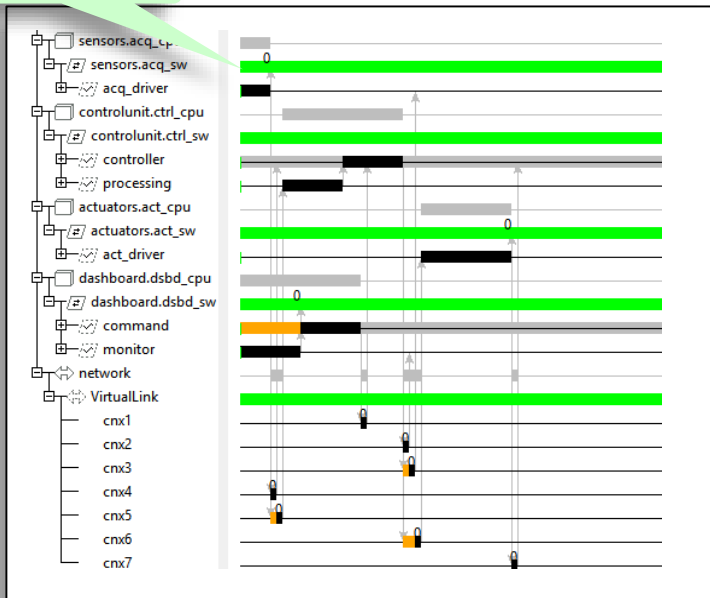
**END** ControlSystemTypes;



# Example 2/4

## Real-Time analysis with Marzhin and Cheddar

Simulation



Response Time analysis

	Deadline	Computed	Max Cheddar	Max Marzhin	Avg Cheddar	Avg Marzhin	Min Cheddar	Min Marzhin
sensors.acq_cpu		5.00 %		7.32 %				
sensors.acq_sw								
acq_driver	100	5.00000	5	5	5.00	5.00	5	5
controlunit.ctrl_cpu		15.00 %		18.43 %				
controlunit.ctrl_sw								
controller	200	20.00000		20		20.00		20
processing	100	10.00000		10		10.00		10
actuators.act_cpu		15.00 %		34.09 %				
actuators.act_sw								
act_driver	100	15.00000		15		15.00		15
dashboard.dsb_cpu		15.00 %		22.73 %				
dashboard.dsb_sw								
keyboard_c	200	20.00000	20	20	20.00	20.00	20	20
screen_driv	100	10.00000	10	10	10.00	10.00	10	10
network		5.00 %		5.43 %				
VirtualLink								
cnx1		1.00000		1		1.00		1
cnx2		1.00000		1		1.00		1
cnx3		1.00000		3		3.00		3
cnx4		1.00000		1		1.00		1

Edit real time properties

Name	Dispatch_Protocol	Period	Compute_Execution_Time	Deadline
sensors.acq_sw.acq_driver	periodic	100 ms	5 ms..5 ms	100 ms
controlunit.ctrl_sw.controller	periodic	200 ms	10 ms..10 ms	200 ms
controlunit.ctrl_sw.processing	periodic	100 ms	10 ms..10 ms	100 ms
actuators.act_sw.act_driver	periodic	100 ms	15 ms..15 ms	100 ms
dashboard.dsb_sw.keyboard_driver	periodic	200 ms	10 ms..10 ms	200 ms
dashboard.dsb_sw.screen_driver	periodic	100 ms	10 ms..10 ms	100 ms

Apply Cancel  Extend current model

Real-Time properties update

```

Static Analysis LAMP Analysis Timing Analysis Safety Analysis Code Generation Doc Generation Scripts

*** LAMP: AADL model predicates loaded.
*** LAMP: response time predicates loaded.
*** LAMP: simulation events predicates loaded.
*** LAMP: library rules loaded.
*** LAMP: goal rules loaded.
*** LAMP: execution started.

FLOW LATENCY ANALYSIS
- Max Response Time of Thread root.sensors.acq_sw.acq_driver: 5
- Max Response Time of Connection root.cnx5: 2 ms
- Max Response Time of Thread root.controlunit.ctrl_sw.processing: 10 ms
- Max Response Time of Thread root.controlunit.ctrl_sw.controller: 10 ms
- Max Response Time of Connection root.cnx6: 2 ms
- Max Response Time of Thread root.actuators.act_sw.act_driver: 15 ms
- Max Response Time of Connection root.cnx4: 1 ms
- Maximum Latency of End to End Flow root.fl: 44 ms
    
```

Scheduling Aware end to end Flow Latency Analysis with LAMP

# Example 3/4

Open PSA generator

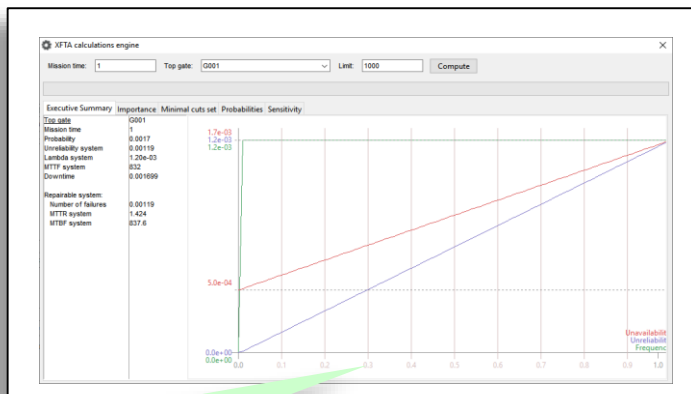
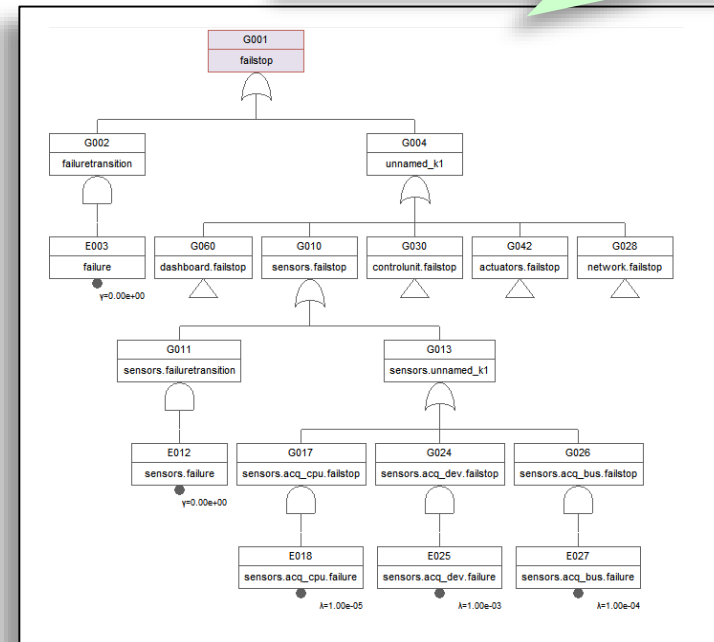
```

Static Analysis LAMP Analysis Timing Analysis Safety Analysis Code Generation Doc Ge
PSA
<?xml version="1.0" encoding="UTF-8"?>
<open-psa author="Arbre Analyste" version="1.12">
  <label>from AADL model</label>
  <attributes>
    <attribute name="company" value="Ellidiss"/>
    <attribute name="author" value="AADL Inspector 1.7"/>
    <attribute name="creation-date" value="0"/>
    <attribute name="modification-date" value="0"/>
    <attribute name="version" value="1"/>
    <attribute name="performances" value="Q,F,T"/>
    <attribute name="page-id" value="1"/>
    <attribute name="page-l-name" value="A - failstop"/>
    <attribute name="page-l-description" value=""/>
    <attribute name="page-l-group" value="0"/>
    <attribute name="compute-id" value="1"/>
    <attribute name="compute-l-name" value="A - failstop"/>
    <attribute name="compute-l-gate" value="failstop"/>
    <attribute name="compute-l-time" value="1"/>
  </attributes>

```

## Safety analysis with Arbre Analyste (\*)

Fault Tree Analysis



MTBF computation

(\*) <https://www.arbre-analyste.fr/en.html#>



# Example 4/4

Security model

## Security analysis with LAMP

Security policy

- *Sec\_R1*: All components involved in a same end to end Flow must be at the same security level.
- *Sec\_R2*: The security level of a component is the higher security level value associated with its Data ports.
- *Sec\_R3*: When two components are connected via a shared Bus, they must comply with the No-Read-Up and No-Write-Down rules.

Security assessment (LAMP)

```

Static Analysis  LAMP Analysis  Timing Analysis  Safety Analysis  Code Generation  Doc Generation  Scripts
[Icons]
SECURITY ANALYSIS
/!\ Security rule R1 error : end to end flow root.fl
    has several several security levels: 3 5 2

/!\ Security rule R2 information : component root.sensors
    is at security level: 5
/!\ Security rule R2 information : component root.sensors.acq_sw
    is at security level: 5
/!\ Security rule R2 information : component root.sensors.acq_sw.acq_drive
    is at security level: 5
/!\ Security rule R2 information : component root.sensors.acq_dev
    is at security level: 3
/!\ Security rule R2 information : component root.controlunit
    is at security level: 5
/!\ Security rule R2 information : component root.controlunit.ctrl_sw
    is at security level: 5
/!\ Security rule R2 information : component root.controlunit.ctrl_sw.ctrl
    is at security level: 5
  
```

Security rules implementation (LAMP)

```

PROPERTY SET LAMP IS
-- ...
Security_Level : AADLINTEGER APPLIES TO
  (Data, Data Access, Port, Parameter);
-- ...
END LAMP;
  
```

```

PACKAGE ControlSystemAnalysis
PUBLIC

ANNEX LAMP {**
/* rule Sec_R1 */
checkFlowSecurity :-
  getRoot(R), getClassifier(R,P,T,I),
  getAncestorRec(P,T,I,Q,U,J),
  isFlowImplementation('END TO END',Q,U,J,E),
  concat('root.',E,F),
  getEndToEndFlow('root',E,M),
  getFlowSecurityLevels(M,[],L,0,N), N > 1,
  printMessageSec_R1(F,L).
checkFlowSecurity :- nl.

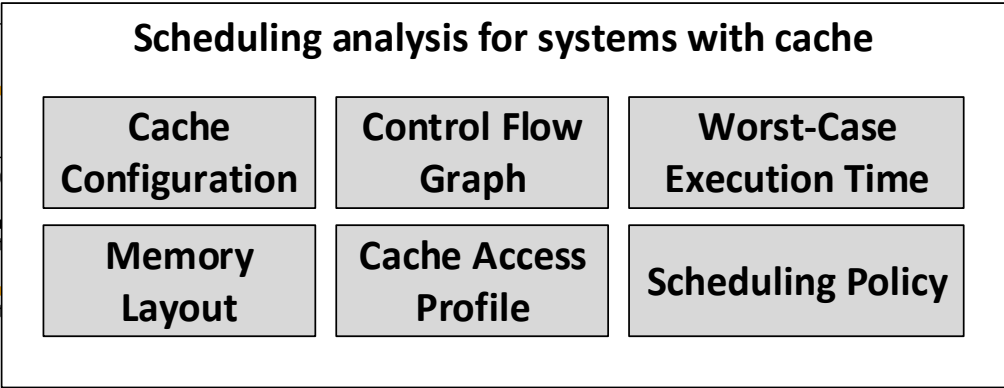
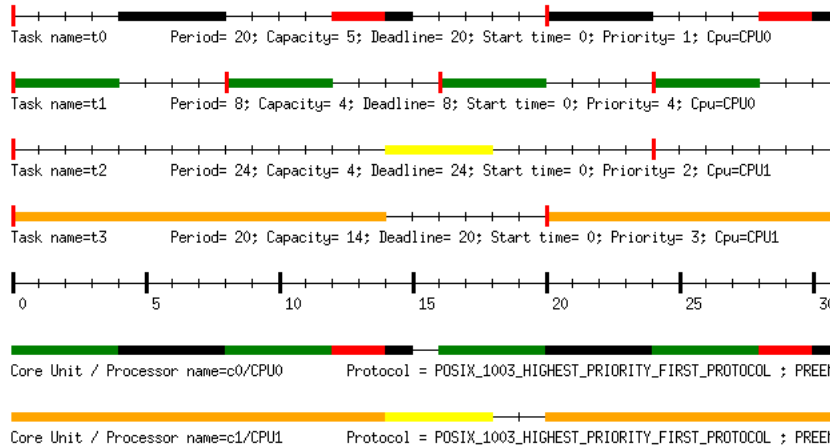
/* rule Sec_R2 */
checkMaxSecurityLevel :-
  getMaxSecurityLevel(X,L),
  printMessageSec_R2(X,L).
checkMaxSecurityLevel :- nl.

/* rule Sec_R3 */
checkNoWriteDown :-
  isAADLBusBinding(_,C,_),
  isAADLConnection(_,P,T,I,_,_,_,C,_,_,_,_),
  getConnectionEnds(P,T,I,C,Xs,Xd),
  getMaxSecurityLevel(Xs,Ls),
  getMaxSecurityLevel(Xd,Ld),
  Ls > Ld,
  printMessageSec_R3(C,Ls,Ld).
checkNoWriteDown :- nl.

-- ...
END ControlSystemAnalysis;
  
```



# Cache-Aware Scheduling Analysis



## Scheduling simulation with cache:

- L1 uniprocessor instruction caches
- Sustainable CPRD model (Cache Preemption Related Delay)
- And known feasibility interval (proved):  $[0, LCM(P_i)]$

## Cache-Aware Priority Assignment Algorithm:

- Audsley oriented algorithm

*F. Singhoff  
S. Rubini  
L. Lemarchand  
H. Nam Tran*

# Our offer

## Products

- **Stood** for AADL: instance model graphical editor for AADL
- **AADL Inspector**: analysis and simulation
- LMP Dev-kit: model processing development framework

## Technology

- LMP: model processing toolbox (prolog)
- **LAMP**: model processing language for AADL
- GMP: DSL graphical editor framework
- Research collaboration with University of Brest/Lab-STICC (Cheddar)

## Services

- Tools sales, support and long-term maintenance
- HOOD & AADL consulting
- Graphical front ends development
- Model processing tools (rules checkers, generators)
- Model transformations
- Tool-chains integration
- R&D partnerships