

AADL Inspector 1.8

Quick Start Guide

```
5 WITH deadlock;
6 END deadlock;
7
8 DATA D
9 -- deadlock occurs if concurrency control protocol is removed
10 PROPERTIES
11   Concurrency_Control_Protocol => Priority_Ceiling_Protocol;
12 END D;
13
14 SYSTEMS
15 SUBCOMPONENTS
16   Processor C;
17   P1 : PROCESS P.I;
18 PROPERTIES
19   P1.Processor_Binding => (P1 (cpu1)) applies to proce
20   P1.deadlock.others;
21
22 PROCESS P
23 PROCESS C
24   S1 : Semaphore (Priority_Ceiling_Protocol);
25 END C;
26   P1.Protocol => (P1
27 PROCESS P
28 END P;
29
30 PROCESS IMPLEMENTATION P.I
31 SUBCOMPONENTS
32   t1 : THREAD T.I;
33   t2 : THREAD T.I;
```



Overview

Project
Browser

AADL Textual
Editor

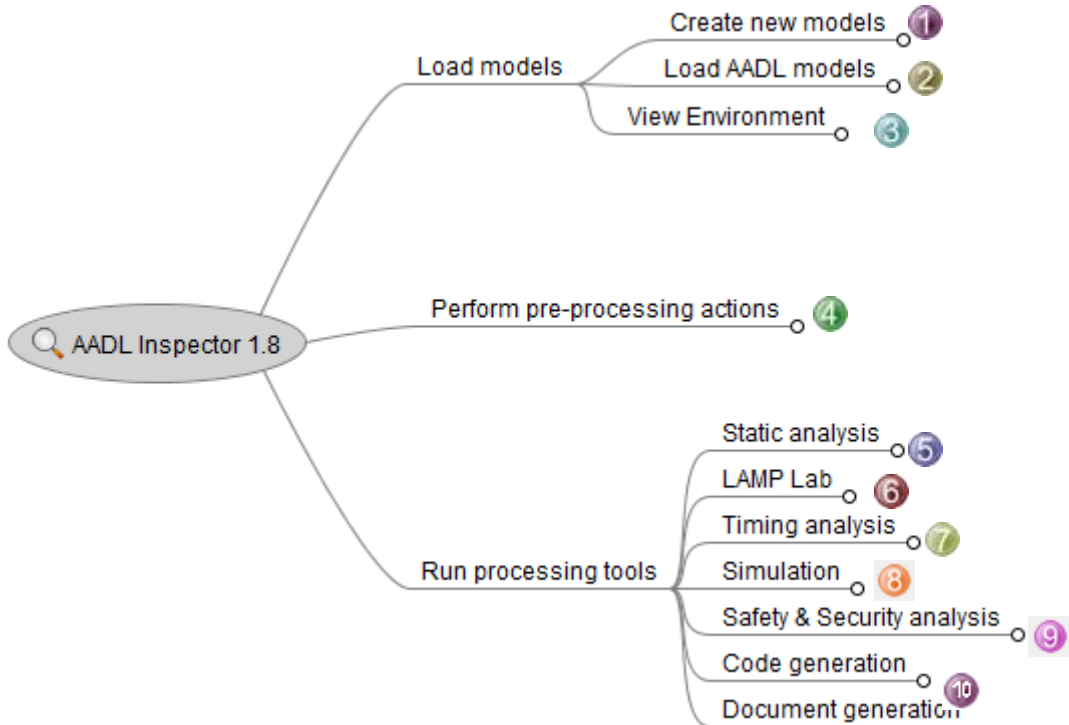
Analysis
Tools

The screenshot displays the AADL Inspector 1.8 interface. On the left is the Project Browser showing a tree view of projects and files. The central pane is the AADL Textual Editor, displaying AADL code for a control system. On the right is the Analysis Tools pane, which includes a table for static analysis results and a timeline view for timing analysis. The Simulator pane at the bottom right shows a graphical representation of the system's state, including a gauge and various component icons.

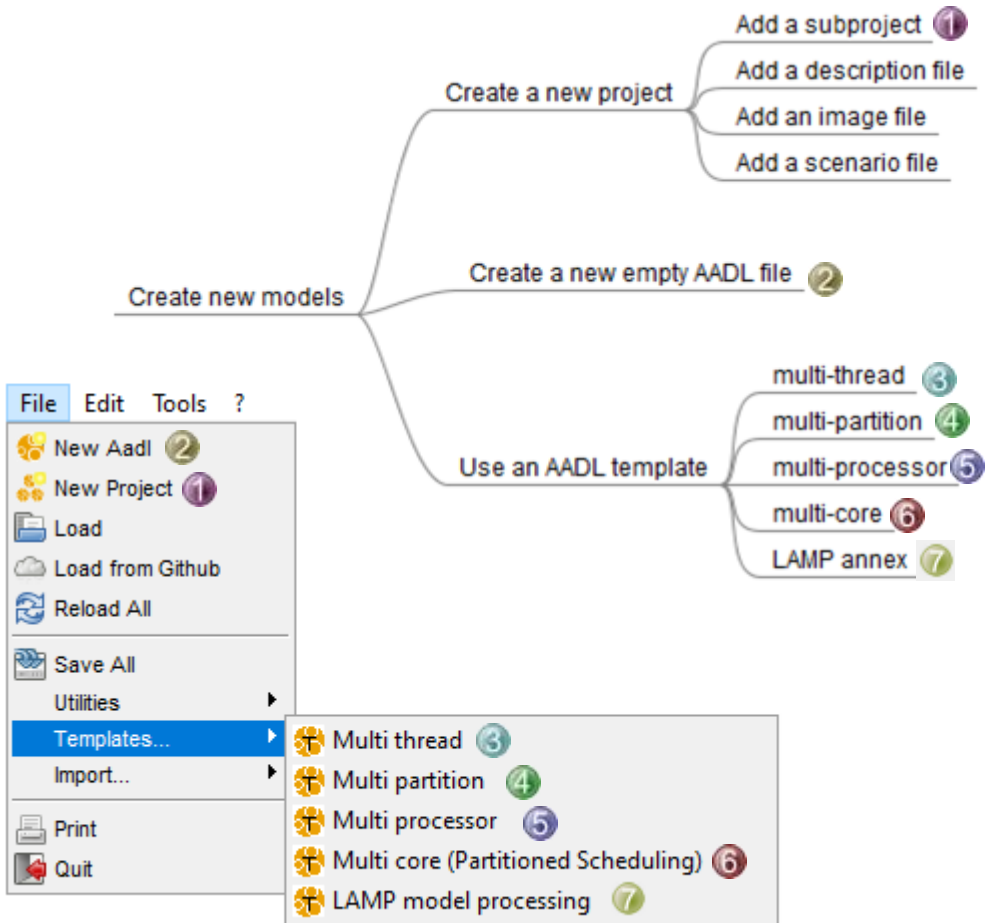
Component	Deadline	Computed	Max Cheddar	Max Marzhin	Avg Cheddar	Avg Marzhin
actuators.act_cpu		15.00%		26.79%		
actuators.act_sw						
act_driver	100	15.00000				
dashboard.dsbd_cpu		15.00%		100.00%		
dashboard.dsbd_sw						
keyboard_driver	200	20.00000	20	20	20.00	20.00
screen_driver	100	10.00000	10	10	10.00	10.00
network		19.50%		43.90%		
VirtualLink						
cnx1	2.00000		2	2.00		

Simulator

Overview



1. Create New Models



1. Create New Models example

The screenshot illustrates the process of creating a new AADL model in the AADL Inspector. The 'File' menu is open, showing the 'Templates...' option selected. A sub-menu is displayed with the following options:

- Multi thread
- Multi partition
- Multi processor
- Multi core (Partitioned Scheduling)
- LAMP model processing

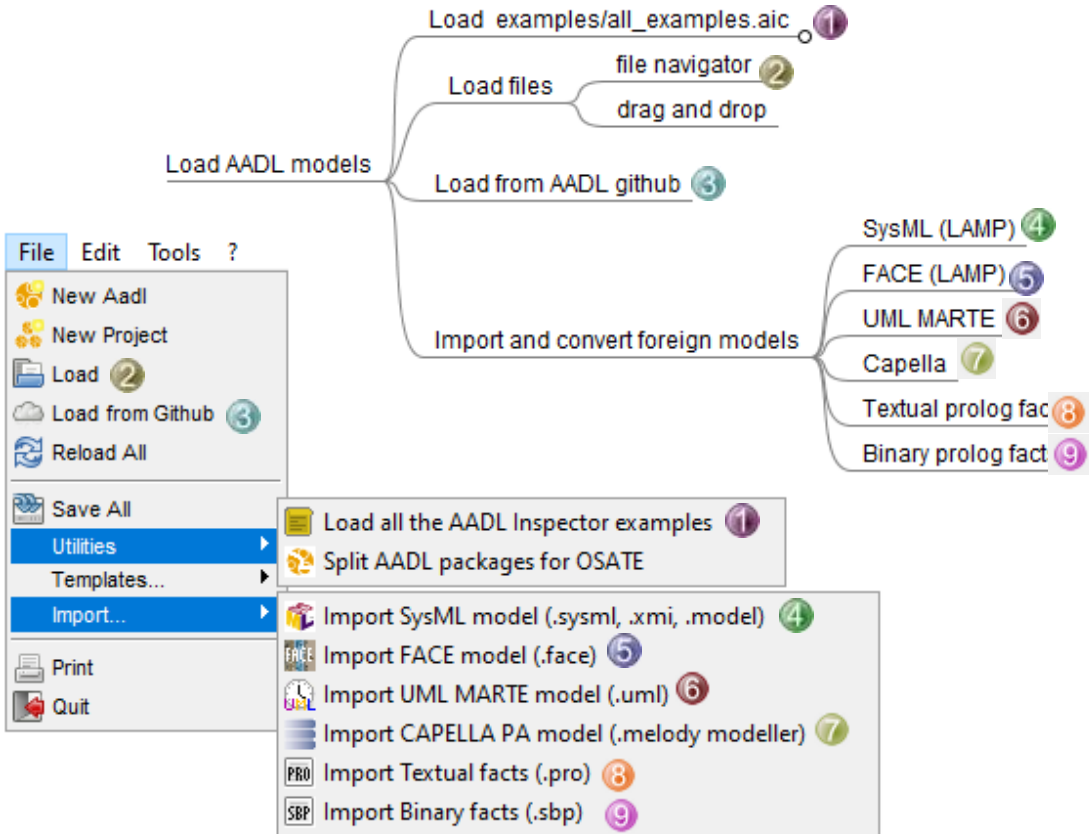
The 'Multi thread' option is highlighted. Below the menu, a dialog box titled 'Set default values' is shown with the following fields:

- Model Name: new_model
- Number of Threads: 4

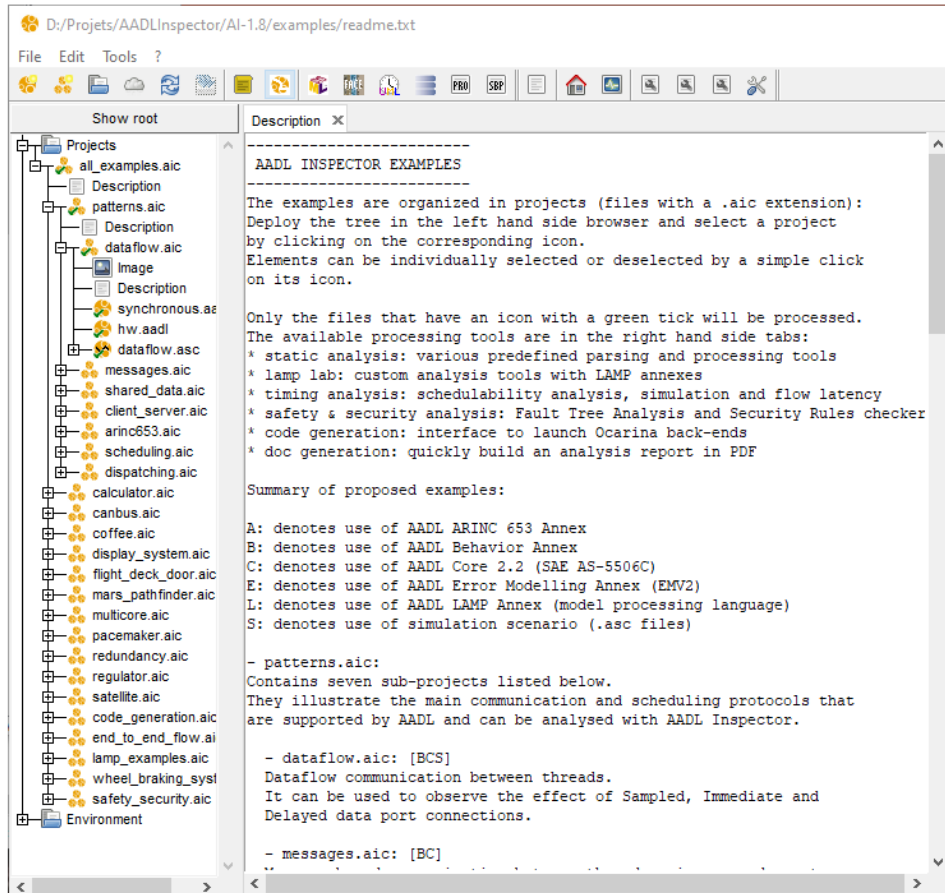
The 'Ok' button is highlighted. In the background, the AADL Inspector interface is visible, showing a project tree with 'new_model.aadl' selected and a code editor displaying the following AADL code:

```
1 -----  
2 -- AADL Inspector Template  
3 -- Multi Thread Real Time System  
4 -----  
5  
6 PACKAGE new_model_pkg  
7 PUBLIC  
8 WITH ellidiss::math::int;  
9 RENAMES ellidiss::math::int::ALL;  
10  
11 -- Root system interface  
12 SYSTEM new_model  
13 END new_model;  
14  
15 -- Root system body  
16 SYSTEM IMPLEMENTATION new_model.i  
17 SUBCOMPONENTS  
18 hw : PROCESSOR hw.i;  
19 sw : PROCESS sw.i;
```

2. Load Existing Models



2. Load Existing Models example

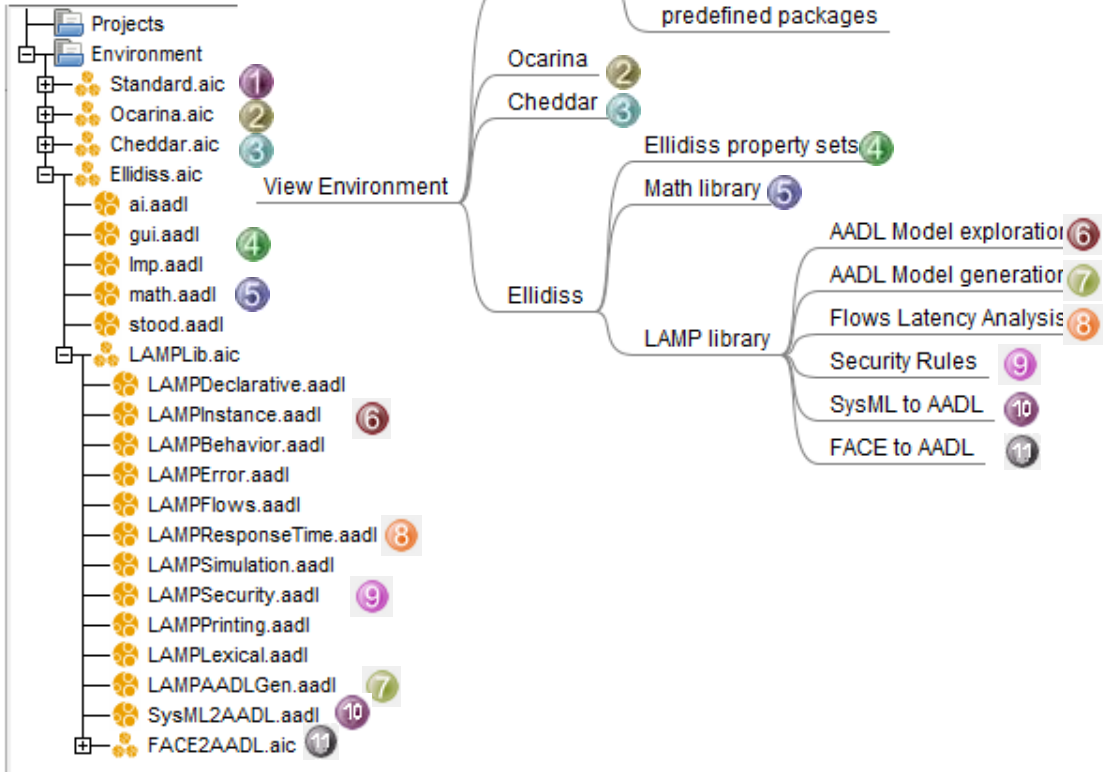


The screenshot displays the AADL Inspector application window. The title bar shows the file path: `D:/Projets/AADLInspector/AI-1.8/examples/readme.txt`. The menu bar includes `File Edit Tools ?`. The toolbar contains various icons for file operations and analysis. The main interface is split into two panes:

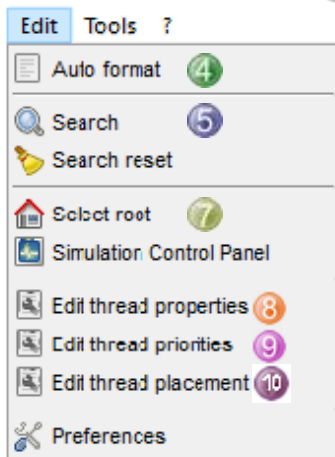
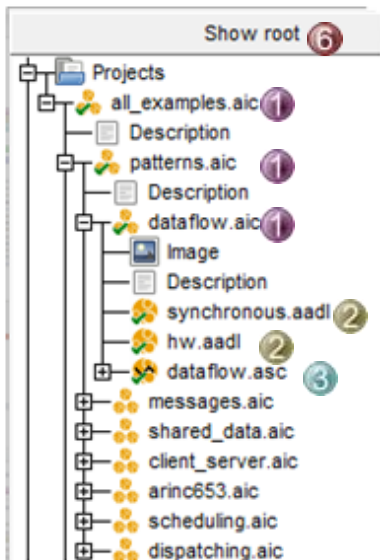
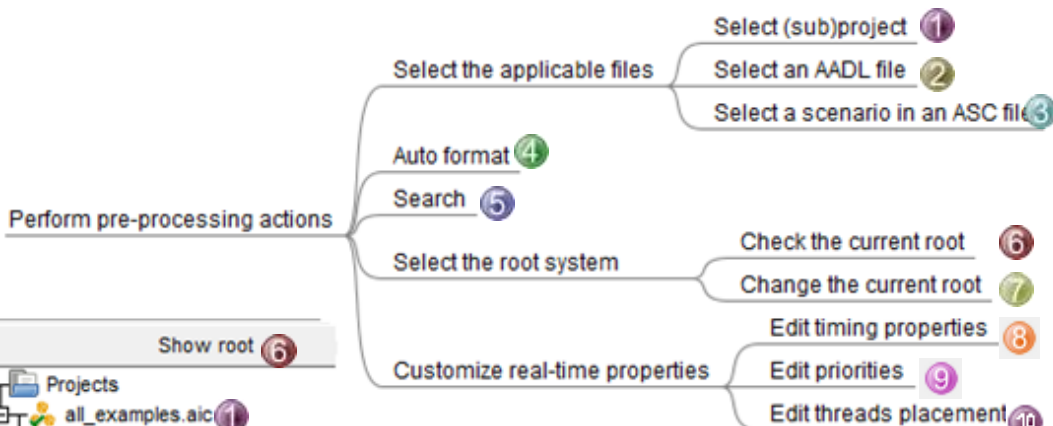
- Left Pane (Project Tree):** Shows a hierarchical view of projects under "Show root". The tree includes folders like "Projects" and "Environment", and numerous files with ".aic" or ".asc" extensions, such as `all_examples.aic`, `patterns.aic`, `dataflow.aic`, `synchronous.aic`, `hw.aadl`, `dataflow.asc`, `messages.aic`, `shared_data.aic`, `client_server.aic`, `arinc653.aic`, `scheduling.aic`, `dispatching.aic`, `calculator.aic`, `canbus.aic`, `coffee.aic`, `display_system.aic`, `flight_deck_door.aic`, `mars_pathfinder.aic`, `multicore.aic`, `pacemaker.aic`, `redundancy.aic`, `regulator.aic`, `satellite.aic`, `code_generation.aic`, `end_to_end_flow.ai`, `lamp_examples.aic`, `wheel_braking_syst`, and `safety_security.aic`.
- Right Pane (Description):** Displays the content of the selected file, `readme.txt`. The text is as follows:

```
-----  
AADL INSPECTOR EXAMPLES  
-----  
The examples are organized in projects (files with a .aic extension):  
Deploy the tree in the left hand side browser and select a project  
by clicking on the corresponding icon.  
Elements can be individually selected or deselected by a simple click  
on its icon.  
  
Only the files that have an icon with a green tick will be processed.  
The available processing tools are in the right hand side tabs:  
* static analysis: various predefined parsing and processing tools  
* lamp lab: custom analysis tools with LAMP annexes  
* timing analysis: schedulability analysis, simulation and flow latency  
* safety & security analysis: Fault Tree Analysis and Security Rules checker  
* code generation: interface to launch Ocarina back-ends  
* doc generation: quickly build an analysis report in PDF  
  
Summary of proposed examples:  
  
A: denotes use of AADL ARINC 653 Annex  
B: denotes use of AADL Behavior Annex  
C: denotes use of AADL Core 2.2 (SAE AS-5506C)  
E: denotes use of AADL Error Modelling Annex (EMV2)  
L: denotes use of AADL LAMP Annex (model processing language)  
S: denotes use of simulation scenario (.asc files)  
  
- patterns.aic:  
Contains seven sub-projects listed below.  
They illustrate the main communication and scheduling protocols that  
are supported by AADL and can be analysed with AADL Inspector.  
  
- dataflow.aic: [BCS]  
Dataflow communication between threads.  
It can be used to observe the effect of Sampled, Immediate and  
Delayed data port connections.  
  
- messages.aic: [BC]
```

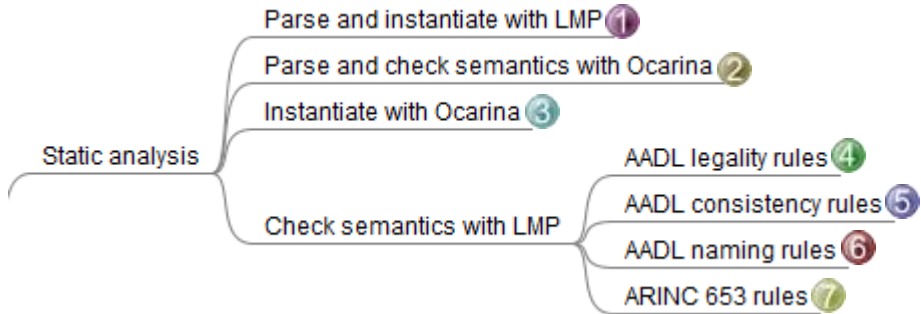
3. View environment



4. Pre-Processing



5. Static Analysis

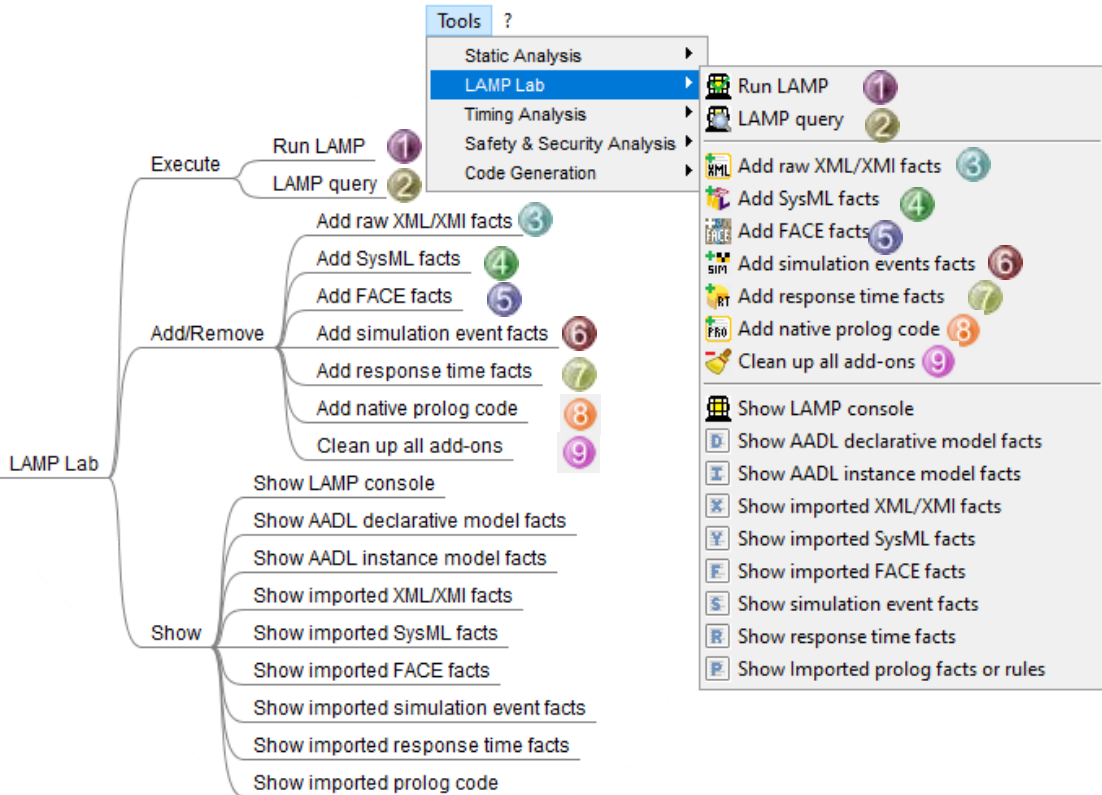


Tools	?
Static Analysis	Parse and Instantiate (LMP) 1
LAMP Lab	Parse (Ocarina) 2
Timing Analysis	Instantiate (Ocarina) 3
Safety & Security Analysis	Check Consistency Rules (LMP) 4
Code Generation	Check Legality Rules (LMP) 5
	Check Naming Rules (LMP) 6
	Check ARINC 653 Rules (LMP) 7

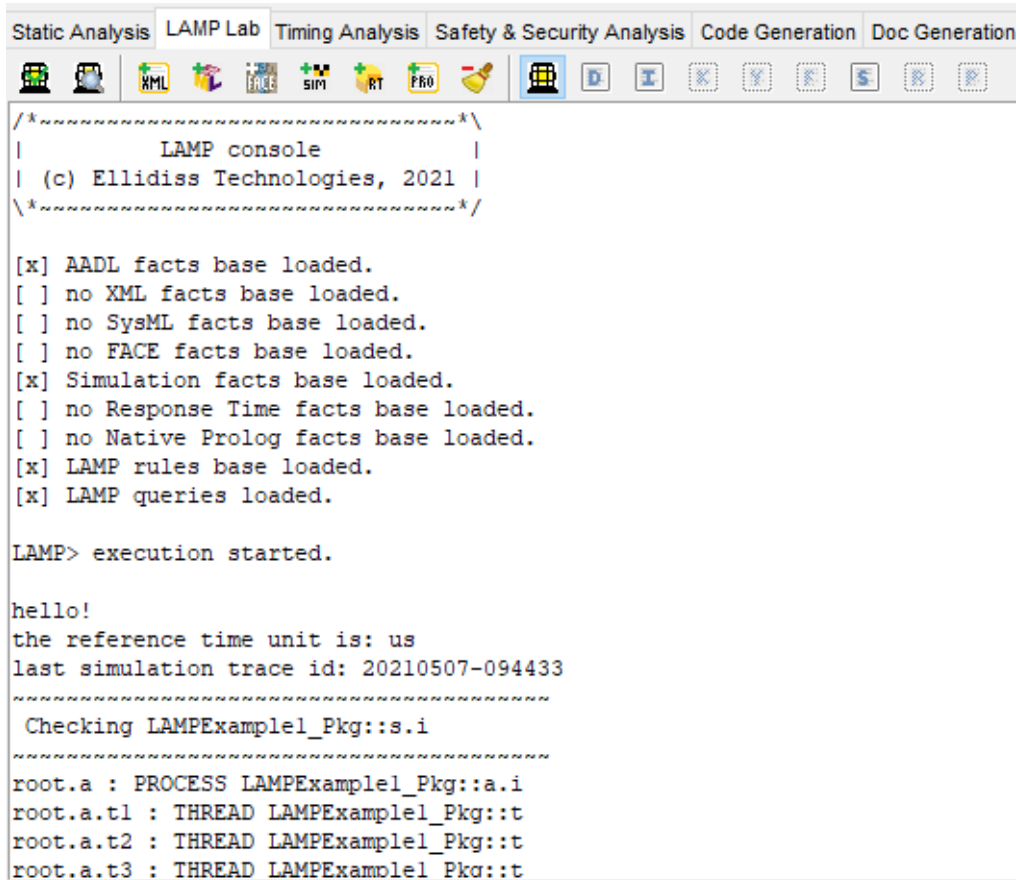
5. Static Analysis example

```
Static Analysis | LAMP Lab | Timing Analysis | Safety & Security Analysis | Co...
ME OP OI CC LC MC BS
-----
aadlrev2.14 (c)Ellidiss Technologies 16Apr2021
AADL-2.2 + BA-2.0
-----
the reference time unit is: ms
-----
*** INSTANCE MODEL ***
-----
Root System Instance: dataflow_Pkg::dataflow.others
-----
17 (system)..... root
92 (processor)... root.my_platform.cpu (RM)
19 (process)..... root.my_process
34 (thread)..... T1 (PERIODIC)
72 (subprogram)..... _square(25735)
39 (thread)..... T2 (PERIODIC)
72 (subprogram)..... _square(25735)
44 (thread)..... T3 (PERIODIC)
72 (subprogram)..... _square(25735)
50 (thread)..... T4 (PERIODIC)
72 (subprogram)..... _square(25735)
-----
```

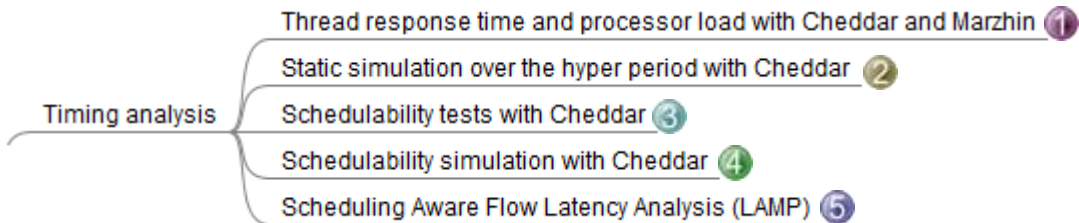
6. LAMP Lab



6. LAMP Lab example



7. Timing Analysis



Tools ?


Static Analysis ▶


LAMP Lab ▶


Timing Analysis ▶


Safety & Security Analysis ▶


Code Generation ▶

 Processor Load & Thread Response Time Analysis ①

 Simulation Timelines (Cheddar) ②

 THE Theoretical Tests (Cheddar) ③

 SIM Simulation Tests (Cheddar) ④

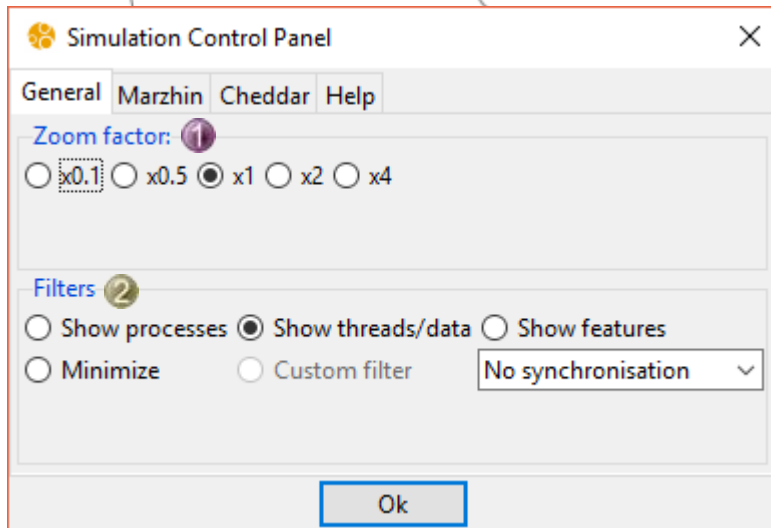
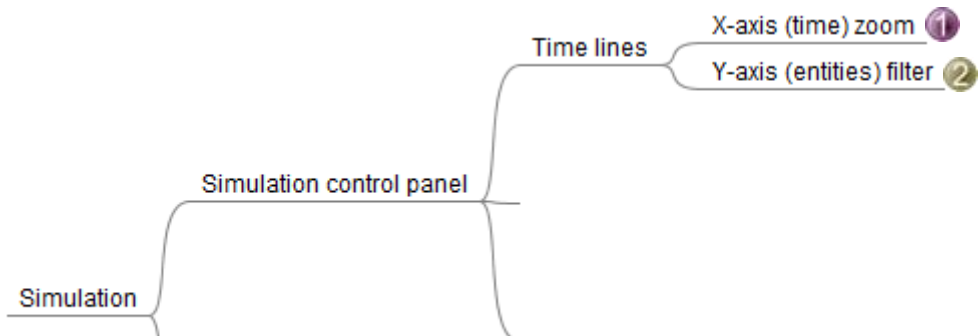
 Scheduling Aware Flows Latency Analysis (SAFLA) with LAMP ⑤

<http://beru.univ-brest.fr/cheddar/>

7. Timing Analysis example

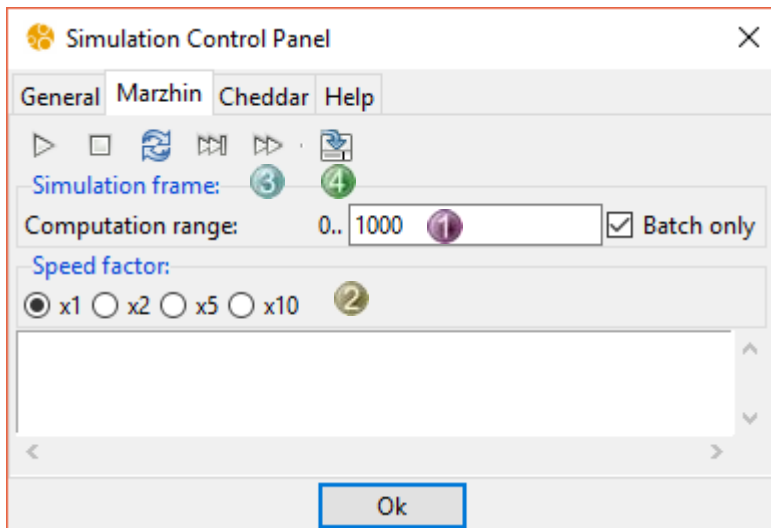
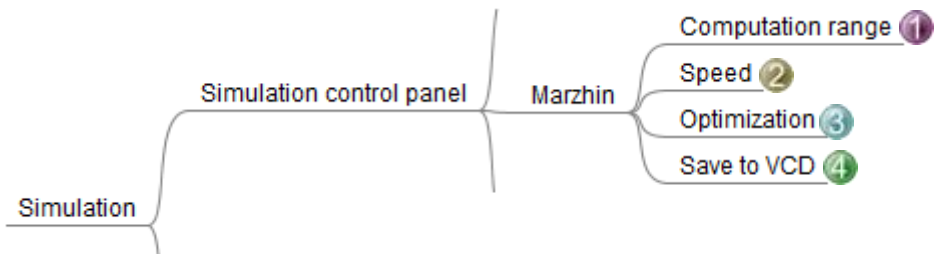
		Deadline	Computed	Max Cheddar	Max Marzhin	Avg Cheddar	Avg Marzhin	Min Cheddar	Min Marzhin
[-]	dashboard.dashboardcpu		30.00 %		31.40 %				
[-]	[-] dashboard.dashboardsw								
	[-] [-] elaboratecommand	20	4.00000	4	4	4.00	4.00	4	4
	[-] [-] displaystatus	10	2.00000	2	2	2.00	2.00	2	2
[-]	motors.motorscpu		56.67 %		58.96 %				
[-]	[-] motors.motorssw								
	[-] [-] leftcontroller	15	7.00000	7	7	5.50	5.50	4	4
	[-] [-] rightcontroller	15	5.00000	5	5	3.50	3.50	2	2
	[-] [-] motorsmanager	10	3.00000	3	3	3.00	3.00	3	3
[-]	mainecu.maincpu		55.00 %		57.23 %				
[-]	[-] mainecu.mainsw								
	[-] [-] controlmanager	20	8.00000	8	8	8.00	8.00	8	8
	[-] [-] statusmanager	10	3.00000	3	3	3.00	3.00	3	3
[-]	can		80.00 %		79.77 %				
[-]	[-] VirtualLink								
	[-] [-] cnx_0		2.00000	11	9	11.00	6.22	11	4
	[-] [-] cnx_3		3.00000	9	10	9.00	4.65	9	3
	[-] [-] cnx_4		3.00000	6	10	6.00	5.29	6	3
	[-] [-] cnx_6		2.00000	13	5	13.00	3.11	13	2

8. Simulation timelines preferences

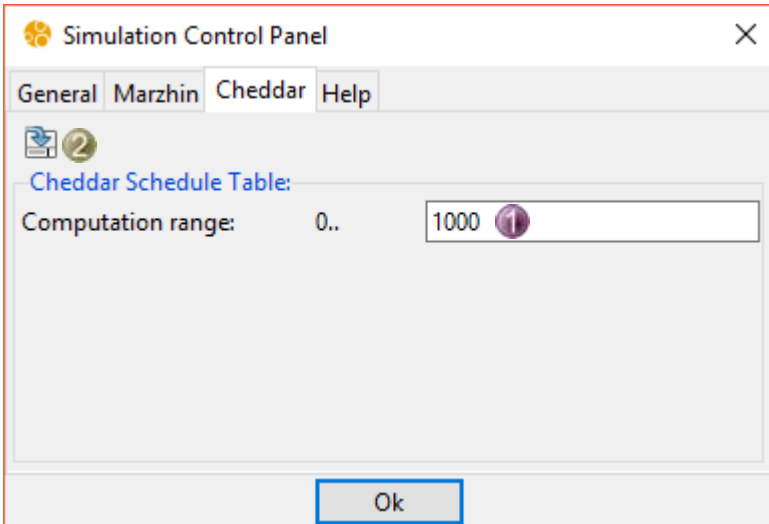
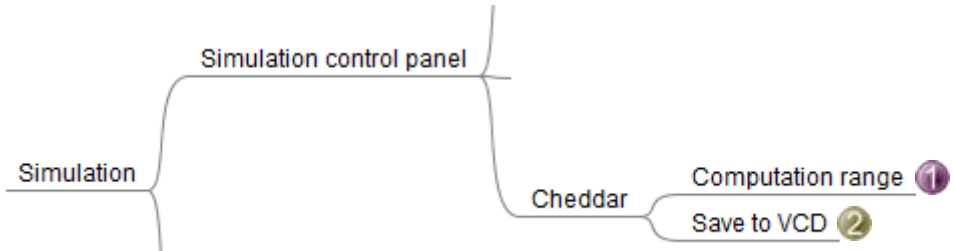


8. Simulation

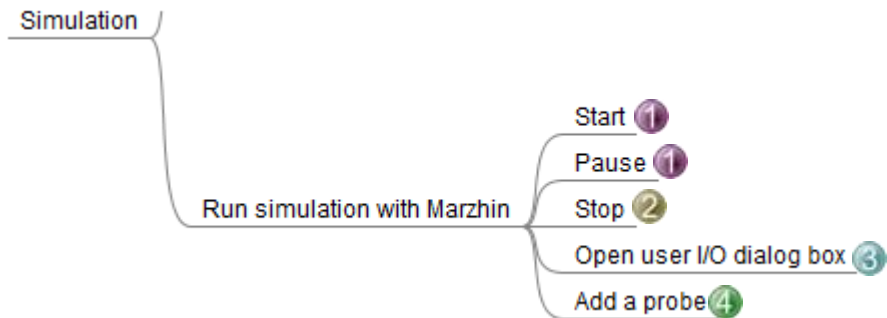
Marzhin preferences



8. Simulation Cheddar preferences



8. Simulation dashboard



1 2 3 4

my_platform.cpu

- my_process
 - input
 - output
 - t1
 - t2
 - t3
 - t4

root.my_platform.cpu.my_process in/out ports

In data port: input | 5

Out data port: output | 256

Close Send all

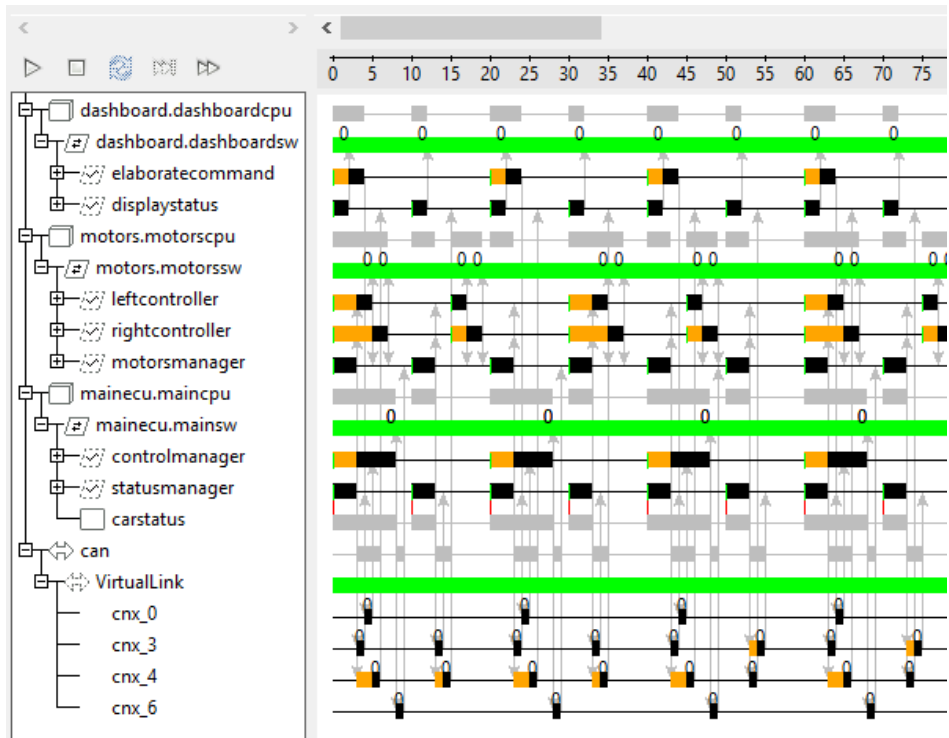
output

256

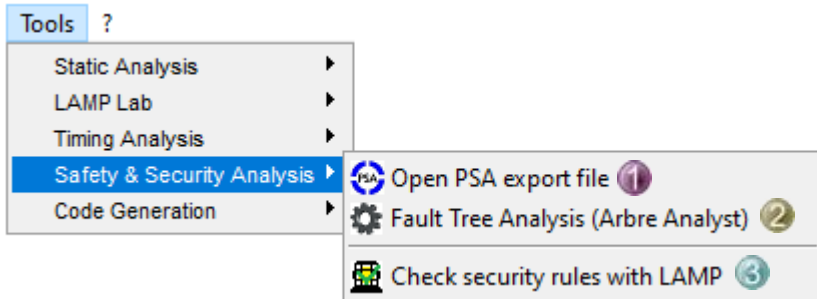
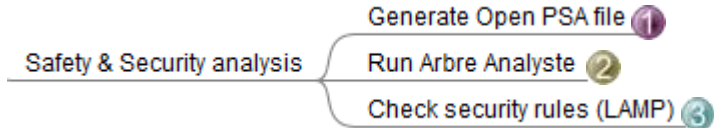
t1 in my_process on my...

0 5 10 15 20

8. Simulation example

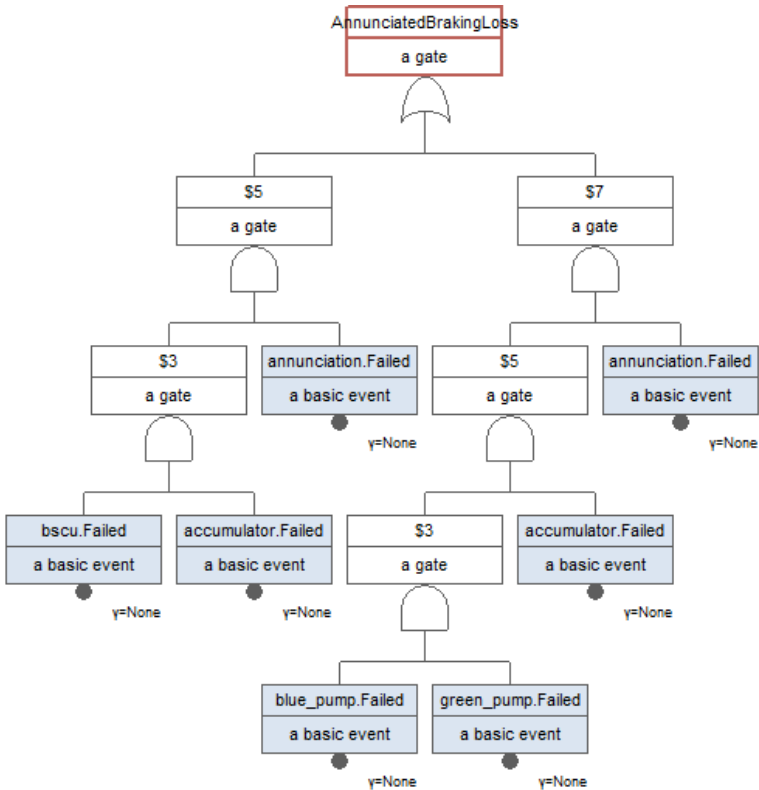


9. Safety & Security Analysis



9. Safety & Security Analysis

Fault Tree Analysis



<https://www.arbre-analyse.fr/en.html>

9. Safety & Security Analysis

Security Rules

```
Static Analysis | LAMP Lab | Timing Analysis | Safety & Security Analysis | Code Generation | Doc Generation
[SA] [G] [LAMP]

/*~~~~~*/
|      LAMP console      |
| (c) Ellidiss Technologies, 2021 |
/*~~~~~*/

[x] AADL facts base loaded.
[ ] no XML facts base loaded.
[ ] no SysML facts base loaded.
[ ] no FACE facts base loaded.
[x] Simulation facts base loaded.
[ ] no Response Time facts base loaded.
[ ] no Native Prolog facts base loaded.
[x] LAMPLib rules base loaded.

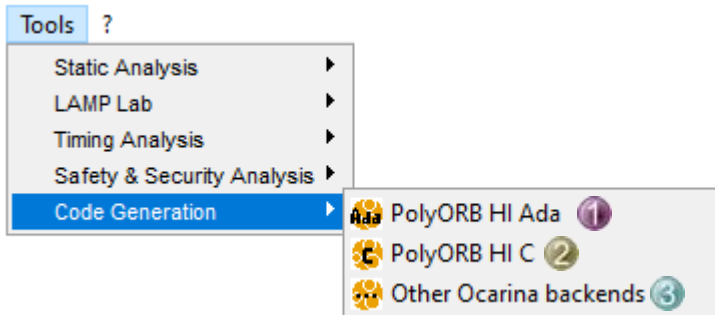
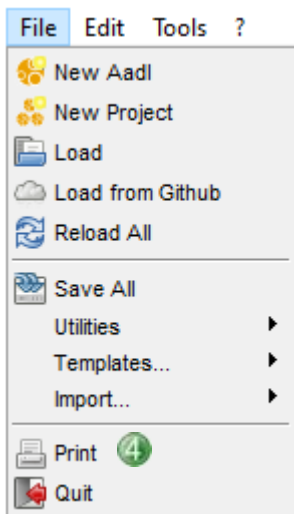
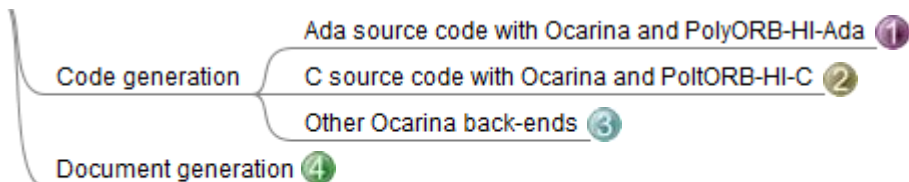
LAMP> execution started.
Information: Security rules can be customized in file:
  environment/Ellidiss/LAMPLib/LAMPSecurity.aadl

SECURITY ANALYSIS

/!\ Security rule R1 (ERROR) : end to end flow root.fl
    has several several security levels: 3 5 2

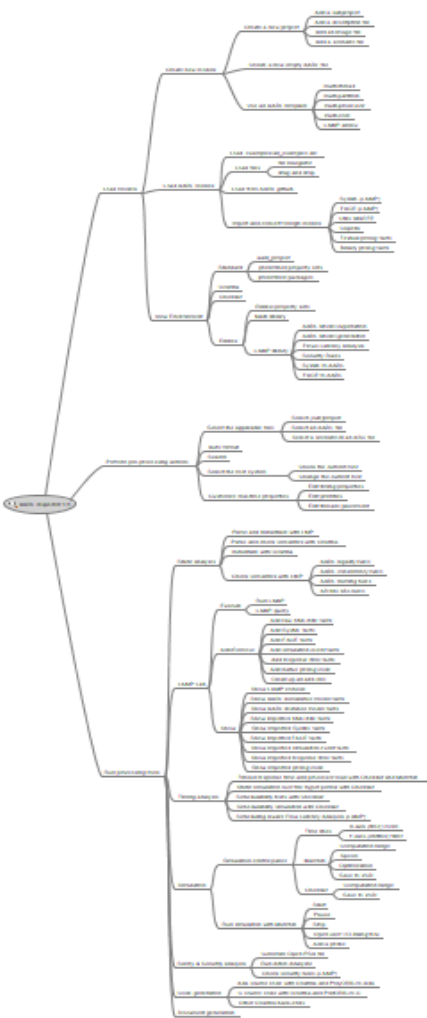
/!\ Security rule R2 (INFORMATION) : component root.sensors
    is at security level: 5
/!\ Security rule R2 (INFORMATION) : component root.sensors.acq_sw
    is at security level: 5
/!\ Security rule R2 (INFORMATION) : component root.sensors.acq_sw.acq_driver
    is at security level: 5
```

10. Code & Document Generation



<http://www.openaadl.org/ocarina.html>

More Information



?	
Help	ⓘ AADL Inspector 1.8 Quick Start
About	ⓘ AI User Manual
License info	ⓘ Cheddar
Open install dir	ⓘ Consistency Rules
Open config dir	ⓘ Legality Rules
Open tmp dir	ⓘ Marzhin
Open doc dir	ⓘ Metrics
Open code dir	ⓘ Naming Rules
	ⓘ ocarina
	ⓘ polyorb-hi-ada_ug
	ⓘ polyorb-hi-c_ug

<http://www.ellidiss.com>

aadl@ellidiss.com

