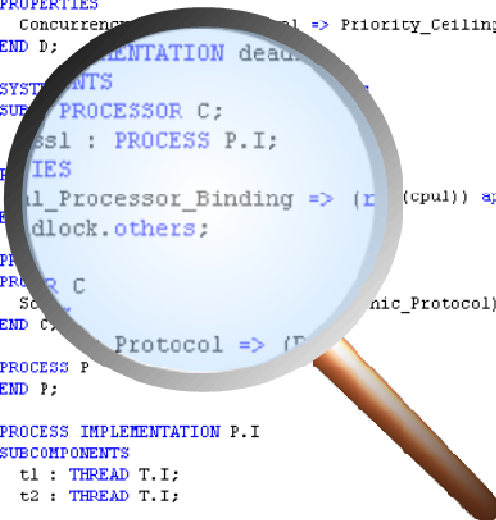


# AADL Inspector 1.7

## Quick Start Guide

```
5 PROCESS 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 SUBCOMPONENTS
15   SUBCOMPONENT PROCESSOR C;
16   Processor : PROCESS P.I;
17
18 PROPERTIES
19   Processor_Binding => (C (cpu1)) applies to processor;
20   deadlock.others;
21
22 PROCESSOR C
23 PROCESSOR C
24   Scheduler : Scheduler (Priority_Ceiling_Protocol);
25 END C;
26
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

Textual  
Editor

Analysis  
Tools

The screenshot displays a software development environment with three main panels:

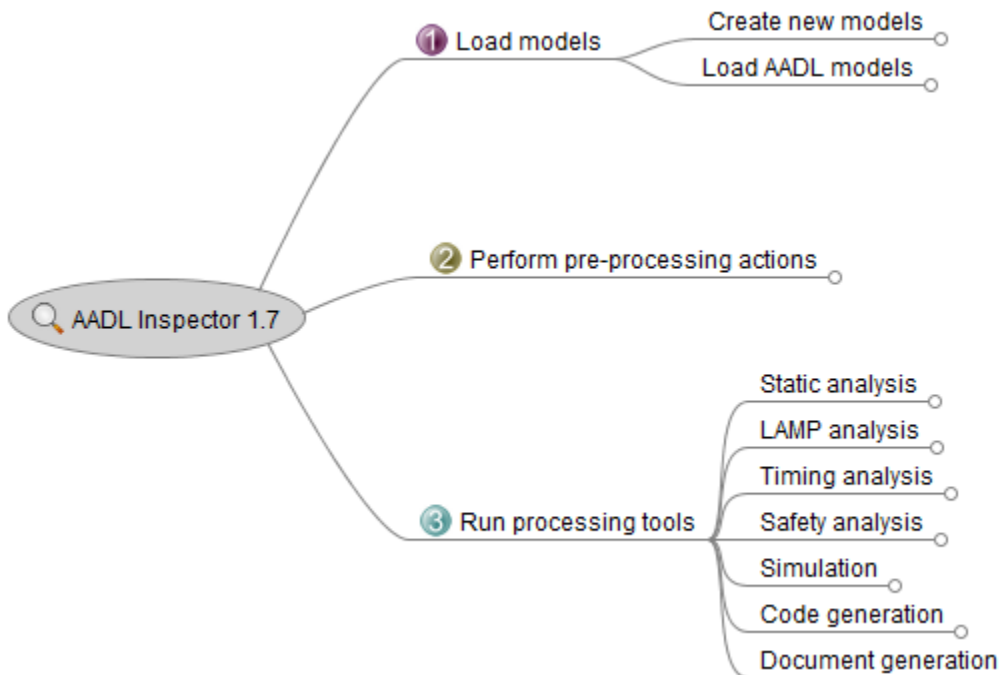
- Project Browser (Left):** Shows a tree view of projects. The 'ecosolar.aadl' file is selected under the 'ecosolar' project.
- Textual Editor (Middle):** Displays the source code for 'ecosolar.aadl'. The code includes package declarations, system definitions, and implementation details for a CAN bus system.
- Analysis Tools (Right):** Shows a table of static analysis results for various components.

Component	Deadline	Computed	Max Cheddar	Max Marzhin	Avg Chedd
dashboard.dashboard	30.00 %			32.61 %	
elaboratec	20	4.00000	4	4	4.00
displaystat	10	2.00000	2	2	2.00
motors.motorscpu	56.67 %			59.79 %	
leftcontrol	15	7.00000	7	5	5.50
rightcontn	15	5.00000	5	7	3.50
motorsma	10	3.00000	3	3	3.00
mainecu.maincpu	55.00 %			59.14 %	

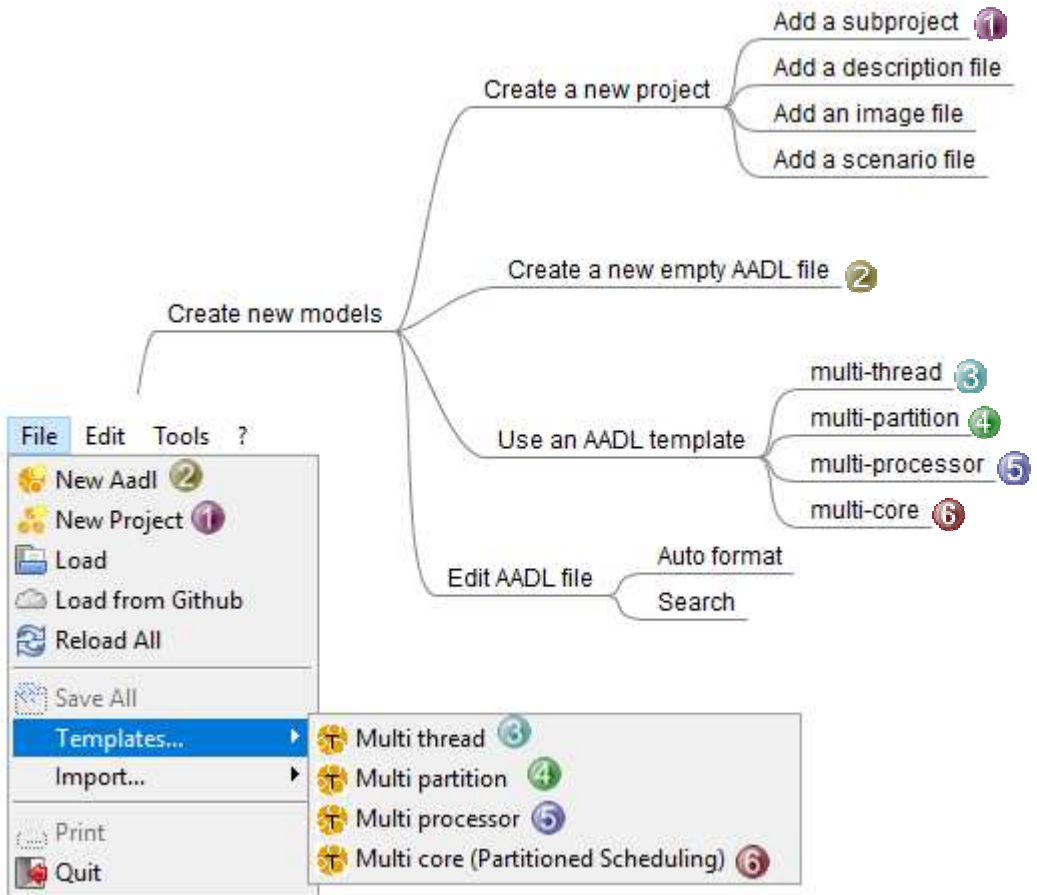
Below the table is a simulator window showing a hierarchical tree of components and a corresponding timing diagram with colored bars representing signal activity over time.

Simulator

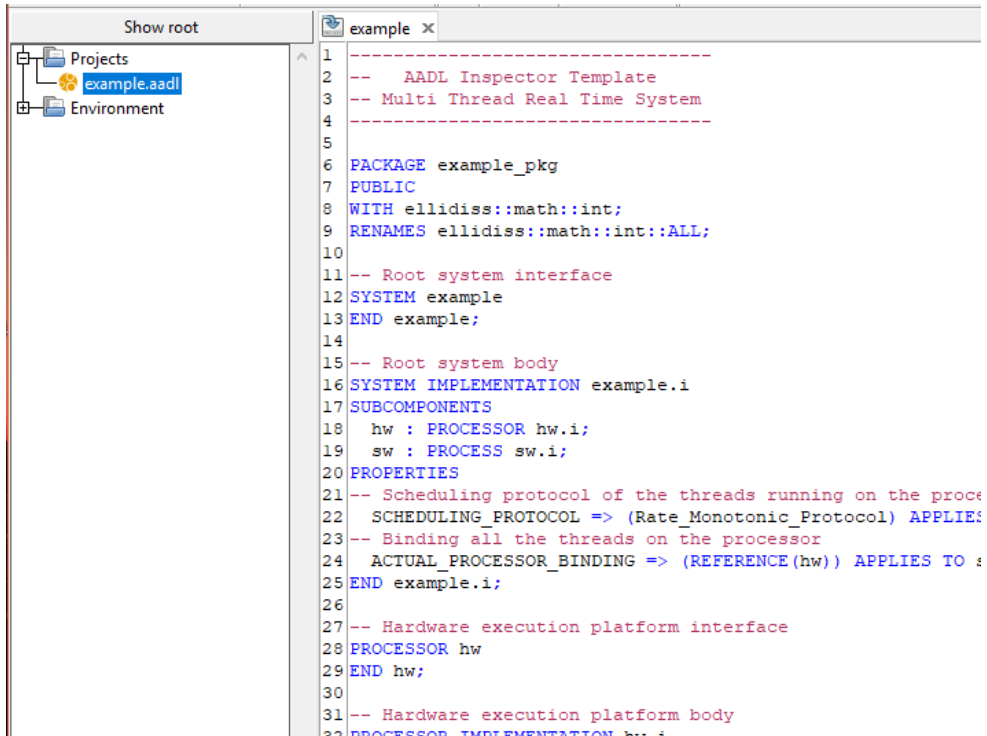
# Overview



# Create New Models



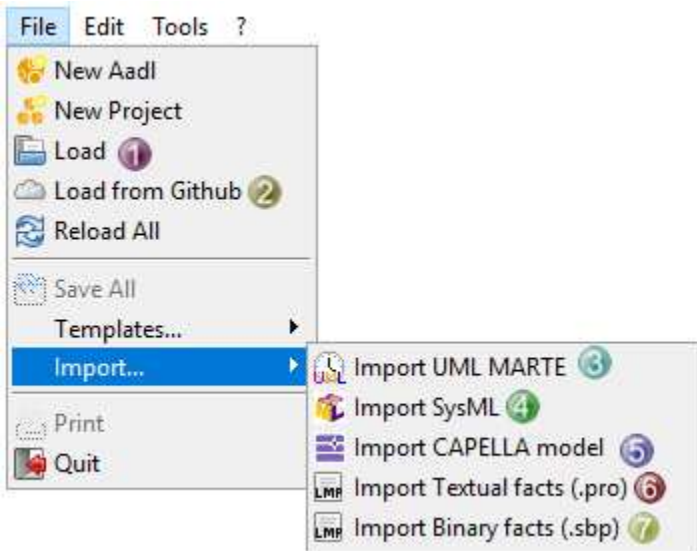
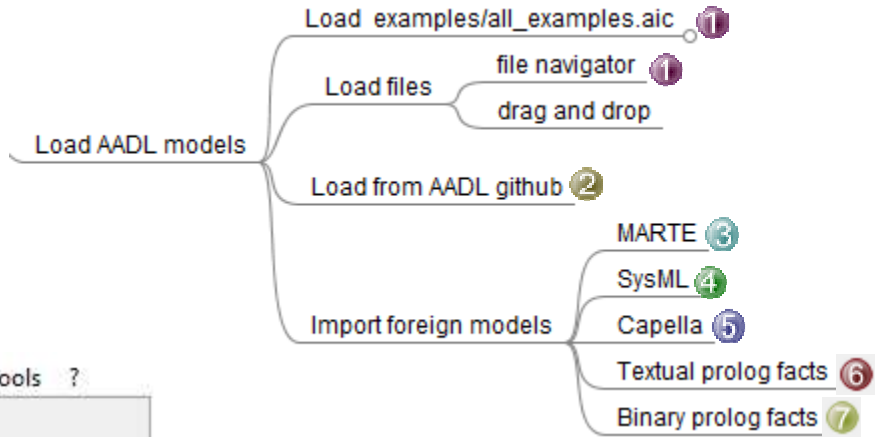
# Create New Models



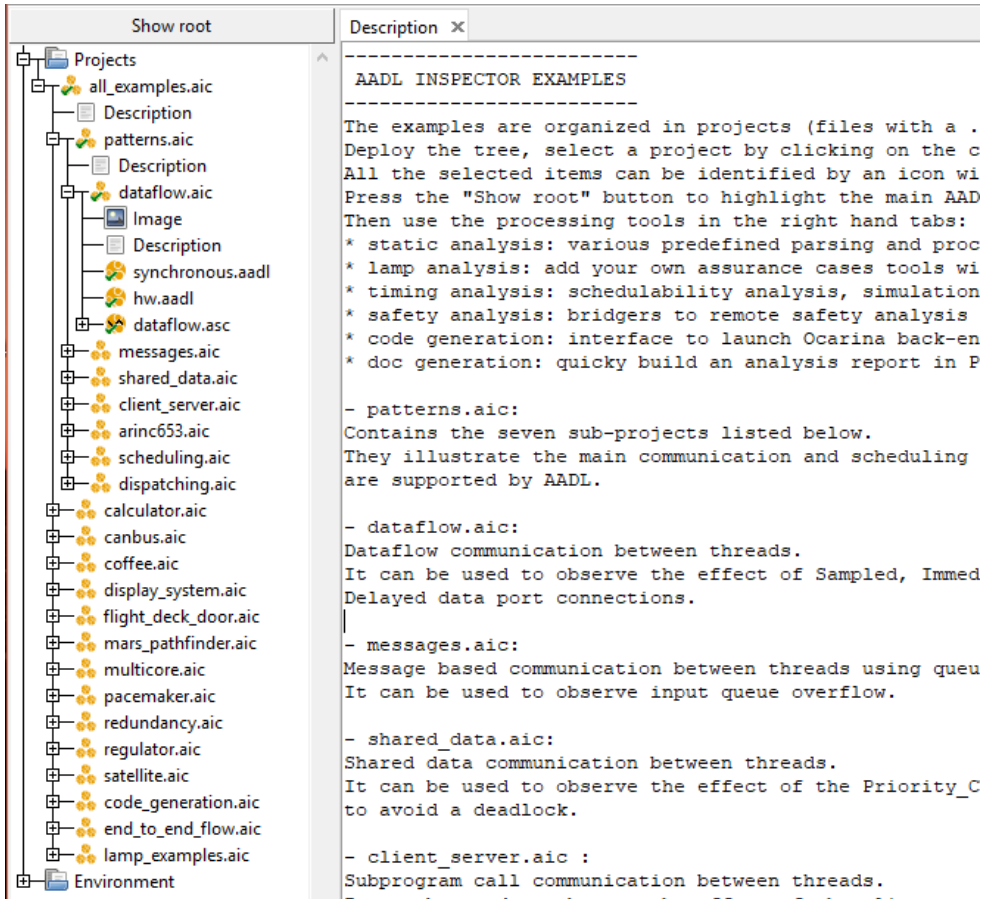
The screenshot shows a software development environment with a project tree on the left and a code editor on the right. The project tree, titled "Show root", contains a "Projects" folder with "example.aadl" selected, and an "Environment" folder. The code editor, titled "example x", displays the following AADL code:

```
1 -----
2 -- AADL Inspector Template
3 -- Multi Thread Real Time System
4 -----
5
6 PACKAGE example_pkg
7 PUBLIC
8 WITH ellidiss::math::int;
9 RENAMES ellidiss::math::int::ALL;
10
11 -- Root system interface
12 SYSTEM example
13 END example;
14
15 -- Root system body
16 SYSTEM IMPLEMENTATION example.i
17 SUBCOMPONENTS
18   hw : PROCESSOR hw.i;
19   sw : PROCESS sw.i;
20 PROPERTIES
21 -- Scheduling protocol of the threads running on the processor
22 SCHEDULING_PROTOCOL => (Rate_Monotonic_Protocol) APPLIES TO :
23 -- Binding all the threads on the processor
24 ACTUAL_PROCESSOR_BINDING => (REFERENCE(hw)) APPLIES TO :
25 END example.i;
26
27 -- Hardware execution platform interface
28 PROCESSOR hw
29 END hw;
30
31 -- Hardware execution platform body
32 PROCESSOR IMPLEMENTATION hw.i
```

# Load Existing Models



# Load Existing Models



The screenshot displays the AADL Inspector interface. On the left, a tree view shows a hierarchy of projects under 'Projects'. The 'all\_examples.aic' project is expanded, showing sub-projects like 'patterns.aic', 'dataflow.aic', 'messages.aic', etc. On the right, the 'Description' pane shows the content for the selected project, which is 'all\_examples.aic'. The description includes a title 'AADL INSPECTOR EXAMPLES' and a detailed explanation of the examples, listing various analysis tools and their purposes.

-----  
AADL INSPECTOR EXAMPLES  
-----

The examples are organized in projects (files with a .  
Deploy the tree, select a project by clicking on the c  
All the selected items can be identified by an icon wi  
Press the "Show root" button to highlight the main AAD  
Then use the processing tools in the right hand tabs:

- \* static analysis: various predefined parsing and proc
- \* lamp analysis: add your own assurance cases tools wi
- \* timing analysis: schedulability analysis, simulation
- \* safety analysis: bridgers to remote safety analysis
- \* code generation: interface to launch Ocarina back-en
- \* doc generation: quicky build an analysis report in P

- patterns.aic:  
Contains the seven sub-projects listed below.  
They illustrate the main communication and scheduling  
are supported by AADL.

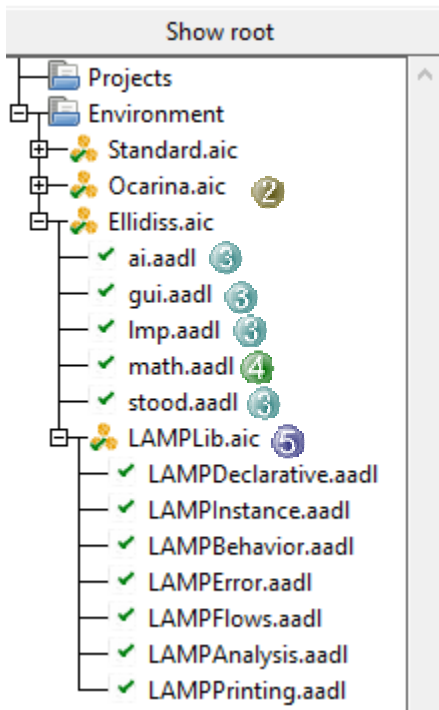
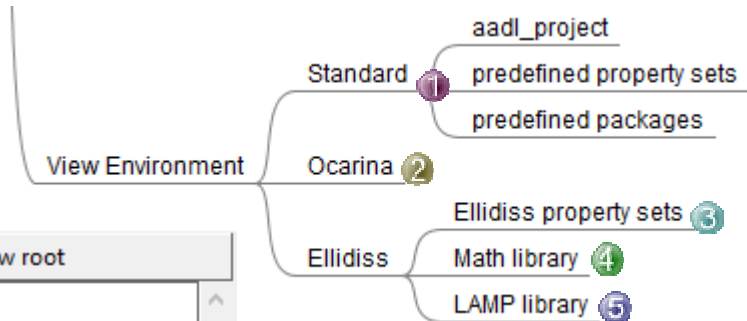
- dataflow.aic:  
Dataflow communication between threads.  
It can be used to observe the effect of Sampled, Immed  
Delayed data port connections.

- messages.aic:  
Message based communication between threads using queu  
It can be used to observe input queue overflow.

- shared\_data.aic:  
Shared data communication between threads.  
It can be used to observe the effect of the Priority\_C  
to avoid a deadlock.

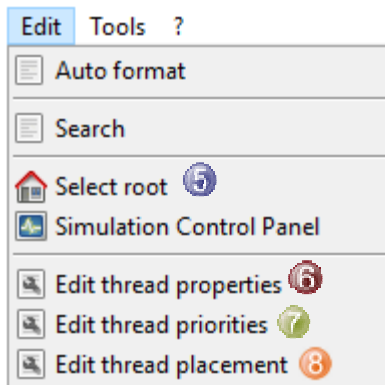
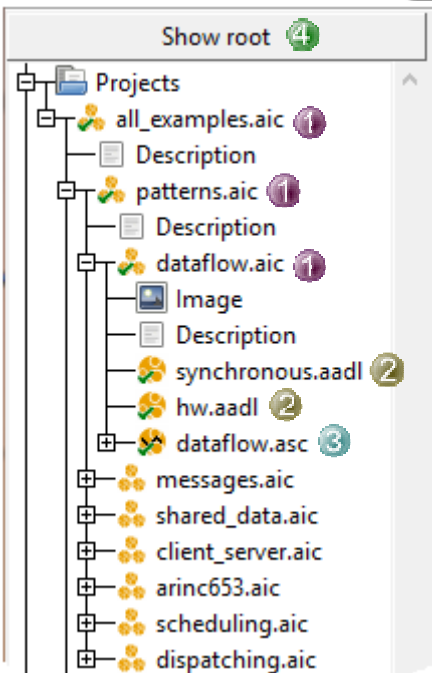
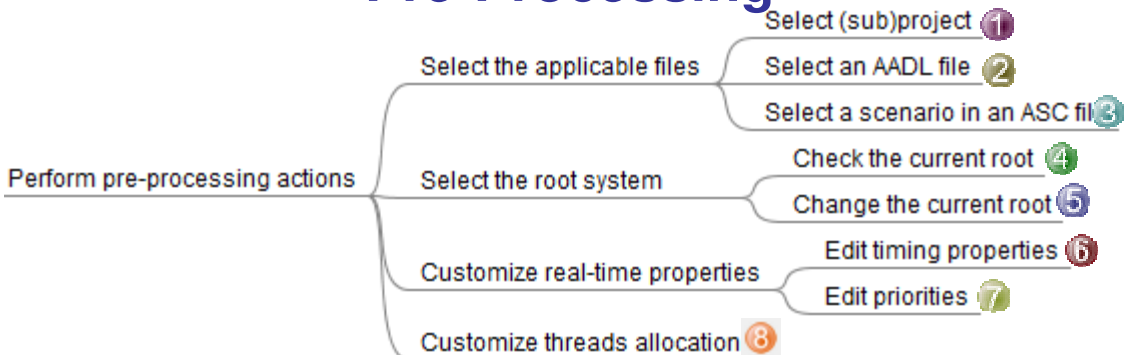
- client\_server.aic :  
Subprogram call communication between threads.

# View environment

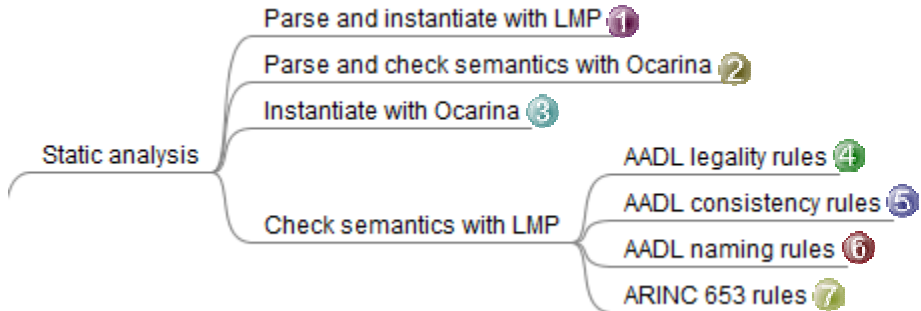




# Pre-Processing



# Static Analysis



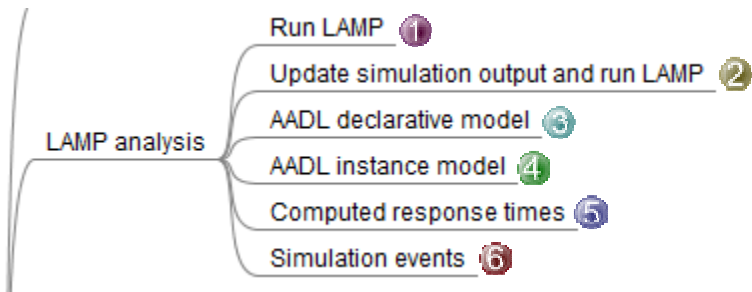
Tools ?

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

# Static Analysis

```
Static Analysis | LAMP Analysis | Timing Analysis | Safety Analysis | Code Generatio
ME OP OI CC LC MC 653
-----
aadlrev2.13 (c)Ellidiss Technologies 27Aug2018
AADL-2.2 + BA-2.0
-----
the reference time unit is: ms
-----
*** INSTANCE MODEL ***
-----
Root System Instance: example_pkg::example.i
-----
16 (system)..... root
18 (processor)... root.hw (RM)
19 (process)..... root.sw
45 (thread)..... th1 (PERIODIC)
46 (thread)..... th2 (PERIODIC)
47 (thread)..... th3 (PERIODIC)
48 (thread)..... th4 (PERIODIC)
-----
```

# LAMP Analysis



Tools ?

Static Analysis ▶

**LAMP Analysis ▶**

Timing Analysis ▶

Safety Analysis ▶

Code Generation ▶

Scripts ▶

- Run LAMP ①
- Update timing simulation data and run LAMP ②
- AADL Declarative Model ③
- AADL Instance Model ④
- Computed Response Times ⑤
- Simulation Events ⑥

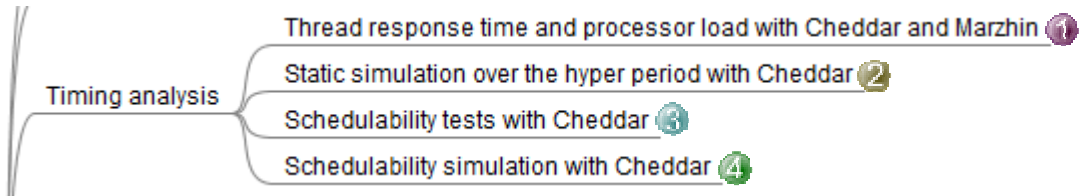
# LAMP Analysis



Static Analysis	LAMP Analysis	Timing Analysis	Safety Analysis	Code Generation

```
*** LAMP: AADL model predicates loaded.
*** LAMP: no response time predicates loaded.
*** LAMP: no simulation events predicates loaded.
*** LAMP: library rules loaded.
*** LAMP: goal rules loaded.
*** LAMP: execution started.
hello!
the reference time unit is: us
no simulation trace available
~~~~~
  Checking LAMPExample1_Pkg::s.i
~~~~~
root.a : PROCESS LAMPExample1_Pkg::a.i
root.a.t1 : THREAD LAMPExample1_Pkg::t
root.a.t2 : THREAD LAMPExample1_Pkg::t
root.a.t3 : THREAD LAMPExample1_Pkg::t
root.c : PROCESSOR LAMPExample1_Pkg::c
root.a.i : PORT LAMPExample1_Pkg::d
```

# Timing Analysis



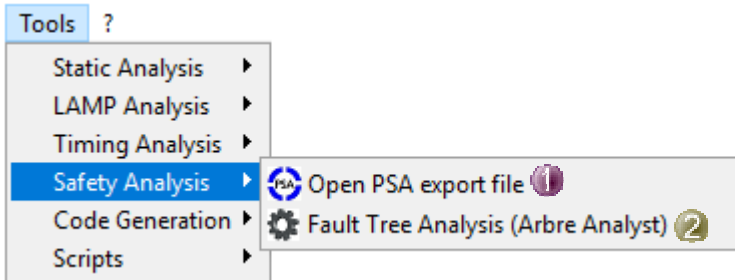
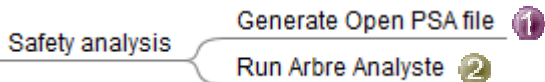
Tools ?

- Static Analysis ▶
- LAMP Analysis ▶
- Timing Analysis ▶**
  - Processor Load & Thread Response Time Analysis ①
  - Simulation Timelines (Cheddar) ②
  - Theoretical Tests (Cheddar) ③
  - Simulation Tests (Cheddar) ④
- Safety Analysis ▶
- Code Generation ▶
- Scripts ▶

# Timing Analysis

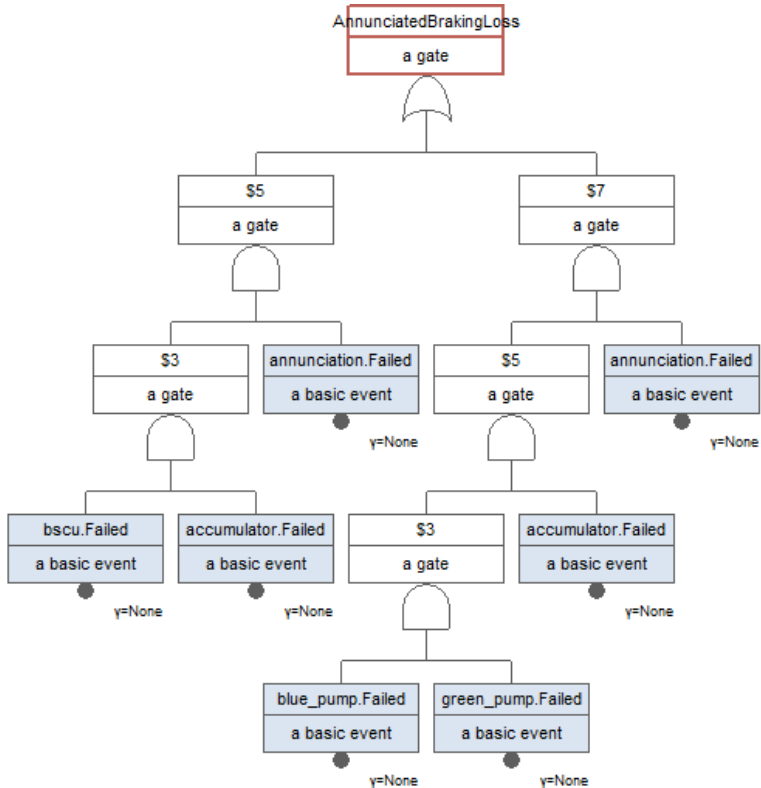
Static Analysis		LAMP Analysis		Timing Analysis		Safety Analysis		Code Generatic	
		THE		SIM					
		Deadline	Computed	Max Cheddar	Max Marzhin				
[-]	my_platform.cpu		65.00 %		67.44 %				
[-]	[-]								
	t1	20	12.00000	6	6				
	t2	20	9.00000	9	8				
	t3	20	6.00000	12	12				
	t4	15	3.00000	3	3				

# Safety Analysis



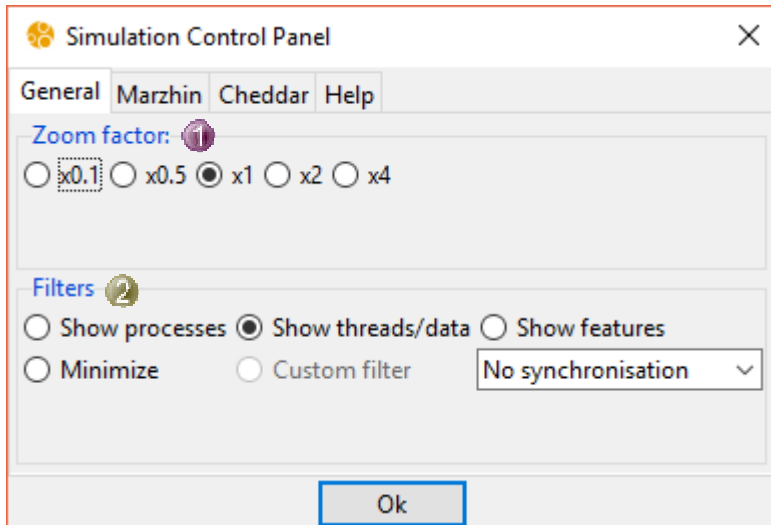
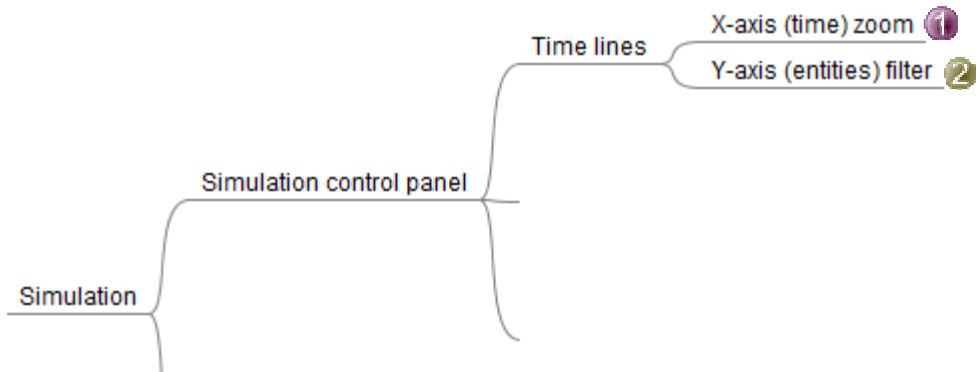


# Safety Analysis

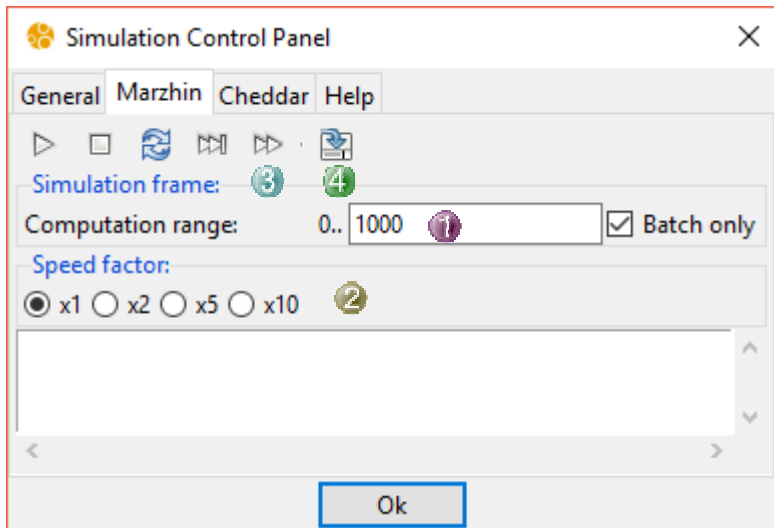
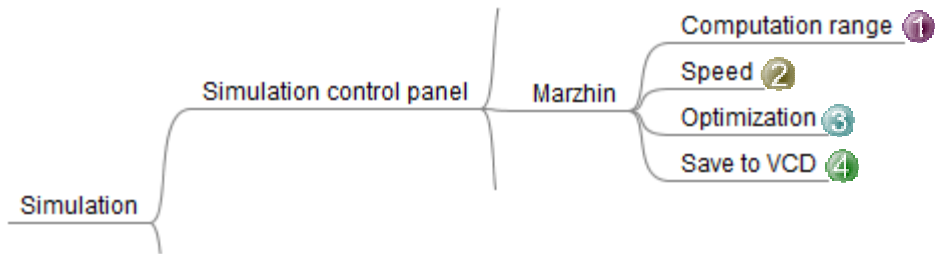


<https://www.arbre-analyste.fr/en.html>

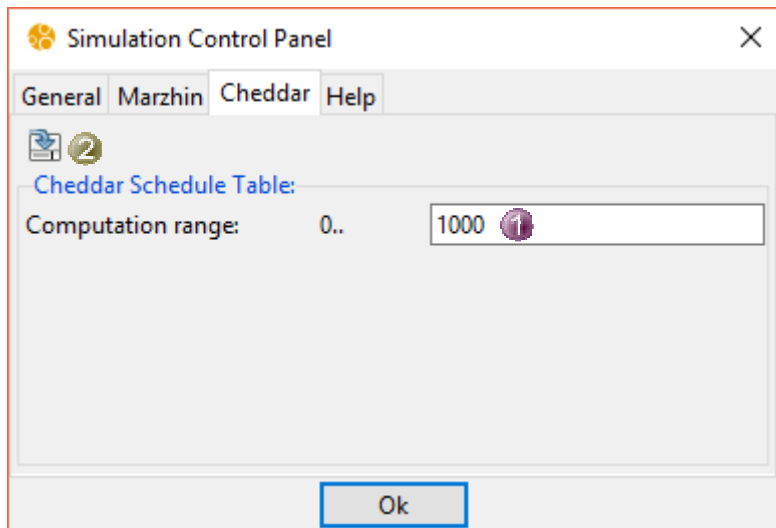
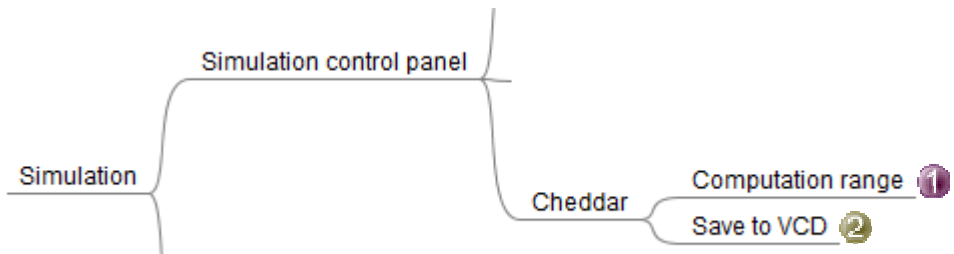
# Simulation



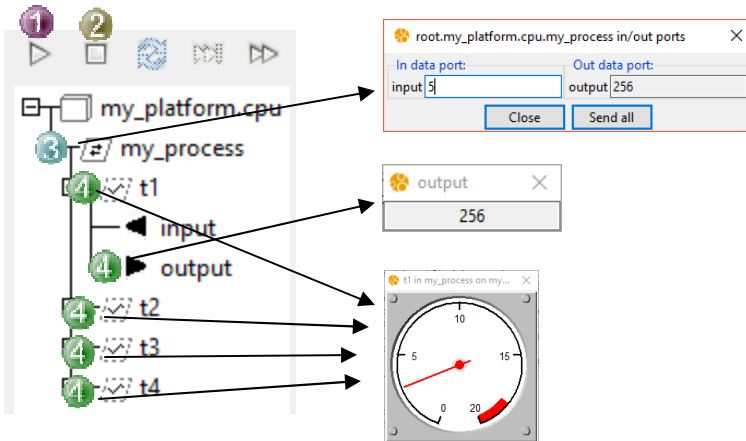
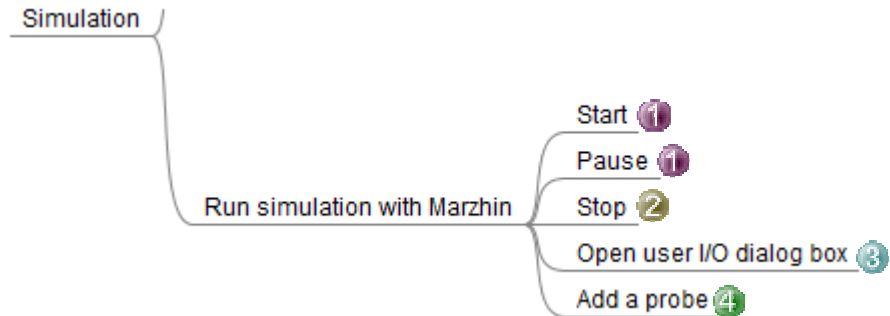
# Simulation



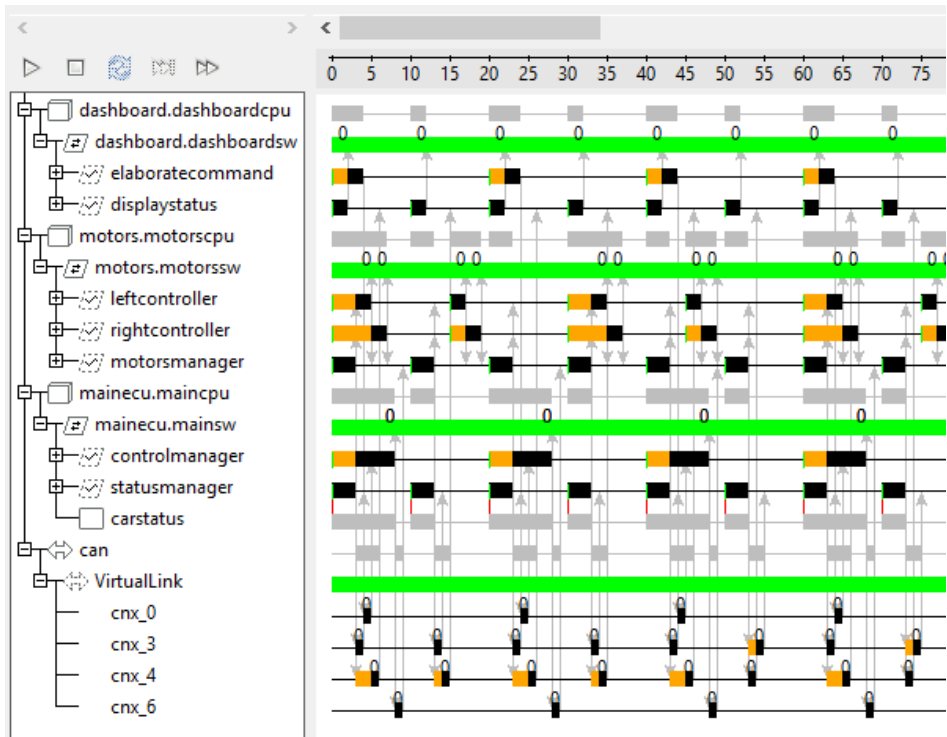
# Simulation



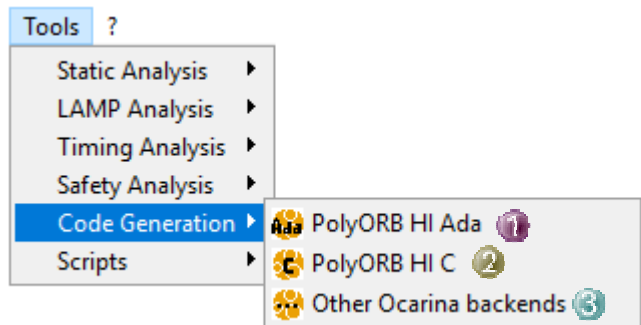
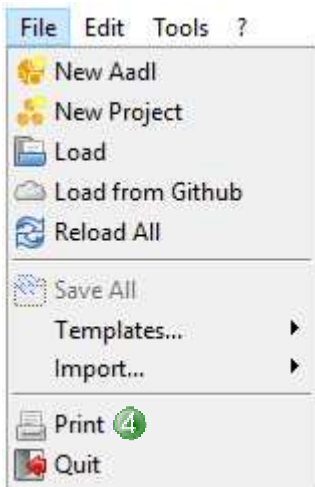
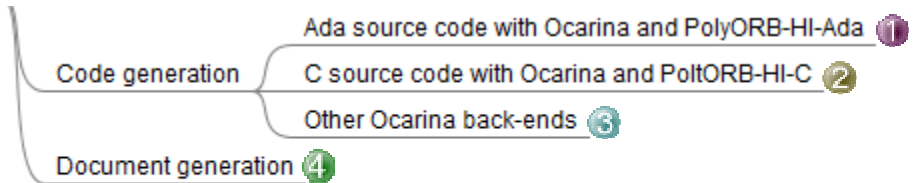
# Simulation



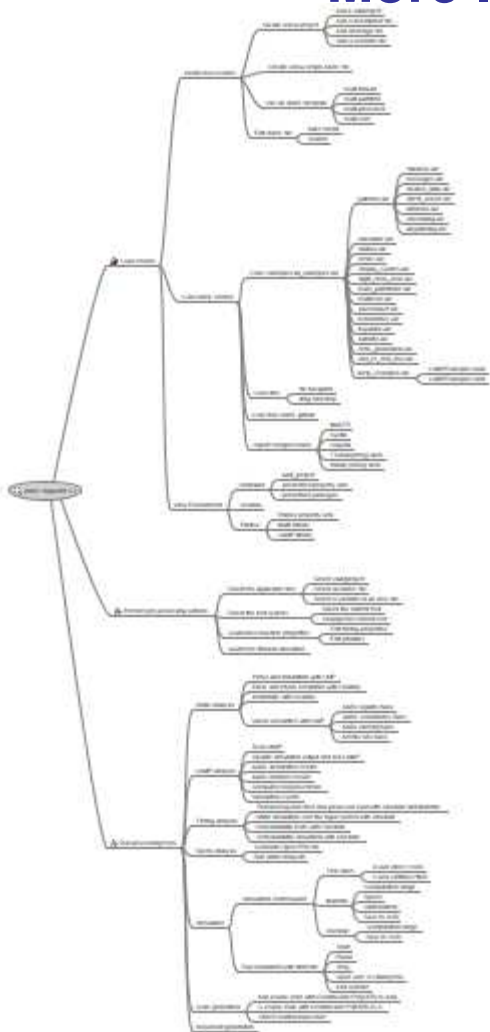
# Simulation



# Code & Document Generation



# More Information



?	
Help	🔍 AADL Inspector 1.7 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://ellidiss.fr>

aadl@ellidiss.fr