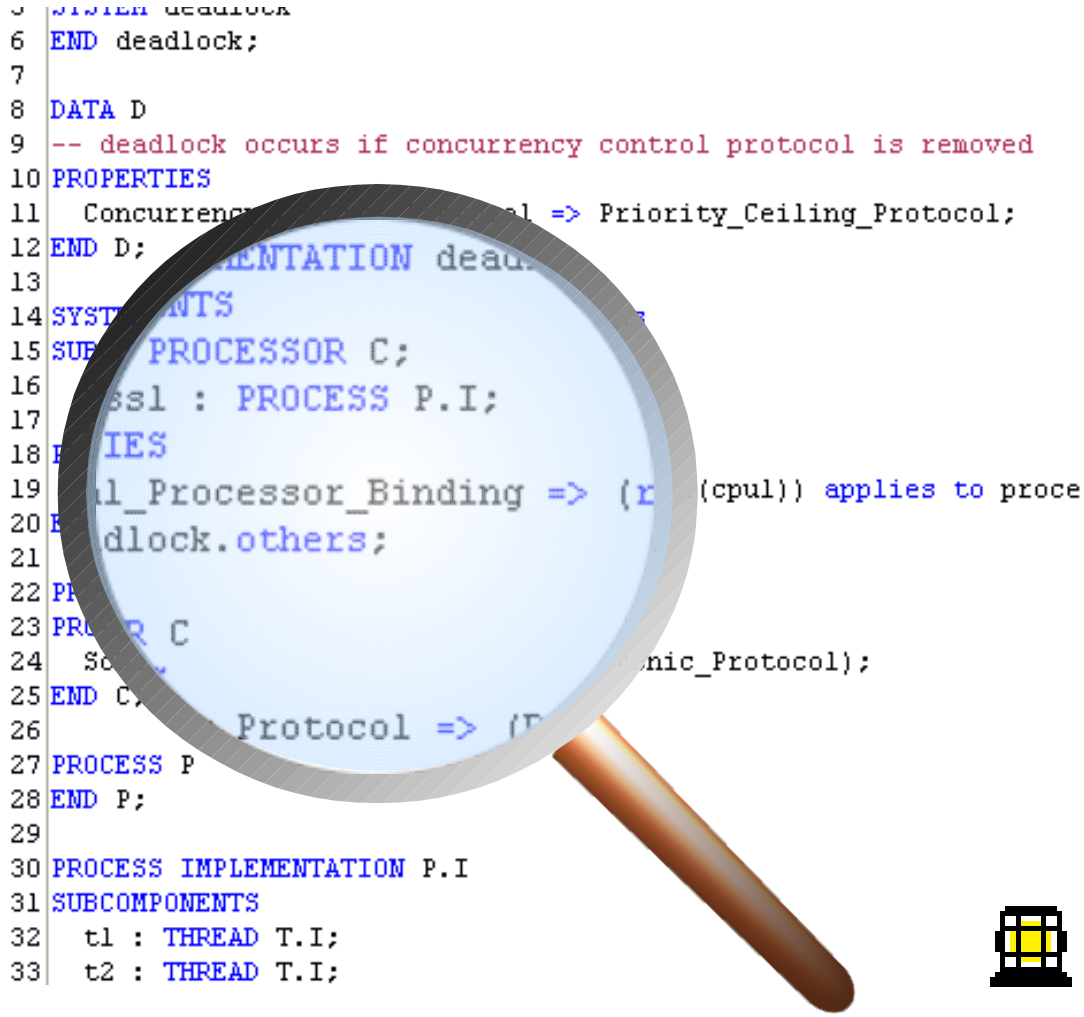
Ellidiss Technologies

AADL Inspector

Model analysis tool for the
Architecture Analysis and Design Language



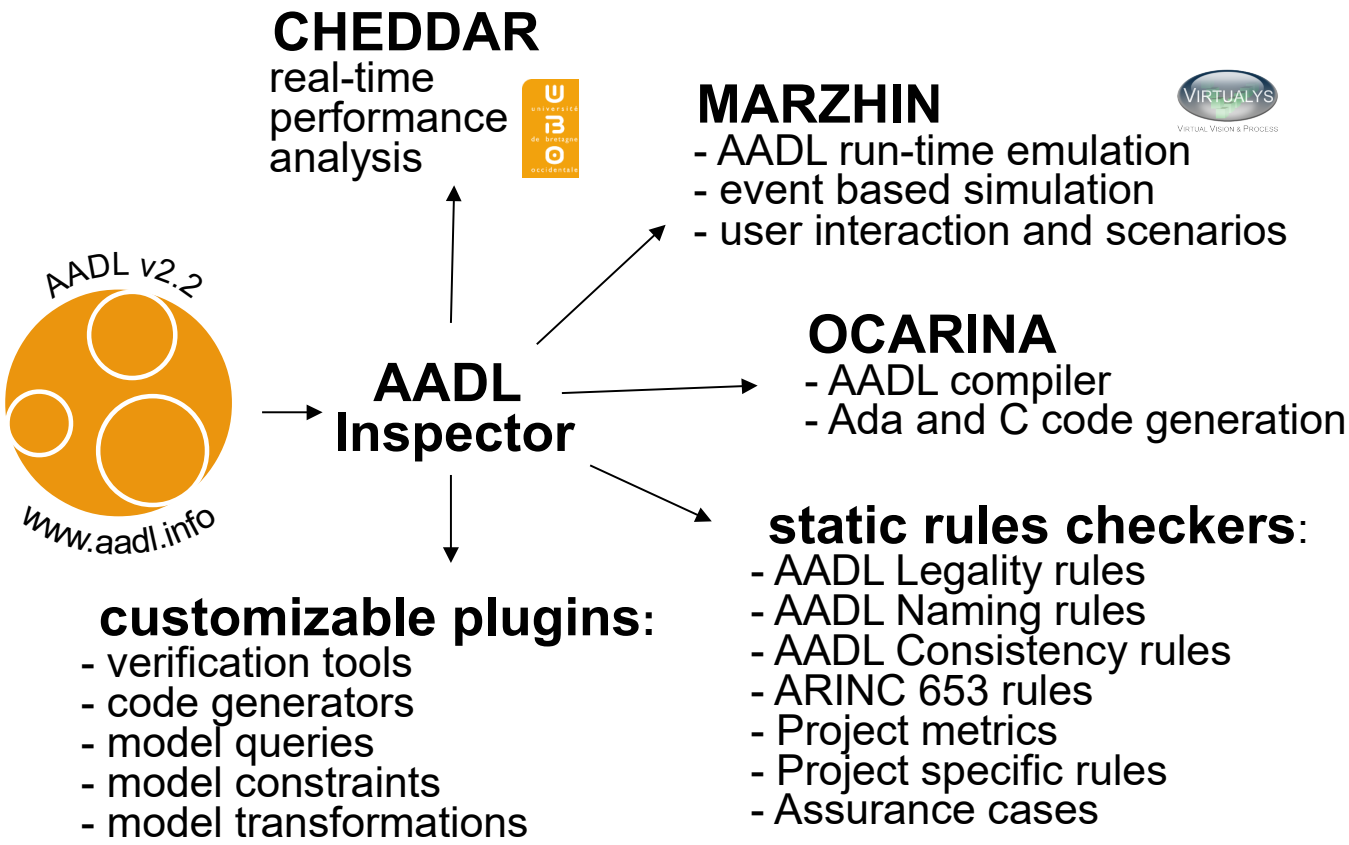
Multi-threaded software (RTOS)
Time and Space Partitioned systems (TSP)
Multi-processor distributed networks
Multi-core architectures

AADL: designing scalable performance-critical real-time systems

Modeling and early validation of real-time features is a key concern for embedded applications. The Architecture Analysis and Design Language (AADL) is a textual and graphical language that can be used to design and analyse the software and hardware architecture of performance-critical real-time systems. AADL v2.2 is the last published version by the SAE, avionics systems division, under number AS 5506C. AADL Inspector can load the multiple textual AAD files or projects that contribute to a system specification and can also analyse AADL Behavior Annex and Error Annex extensions.
Thanks to its textual syntax, AADL can be used as a first class front end modelling language for designing large scale software intensive systems or as a hidden pivot format in case of use of another model driven modelling approach (UML, SysML, Capella, EEA, DSL, ...)

Early detection of problems

One of the benefits of using the AADL for modeling critical systems is that this language has been designed in the purpose of beeing verified. AADL inspector has been specifically developped to insure that the model is correctly formed according to the AADL semantics, allowing an early detection of problems in the design process. Analysis reports are provided by the various analysis tools that are encompassed by AADL Inspector and have been developed by Ellidiss and his partners.

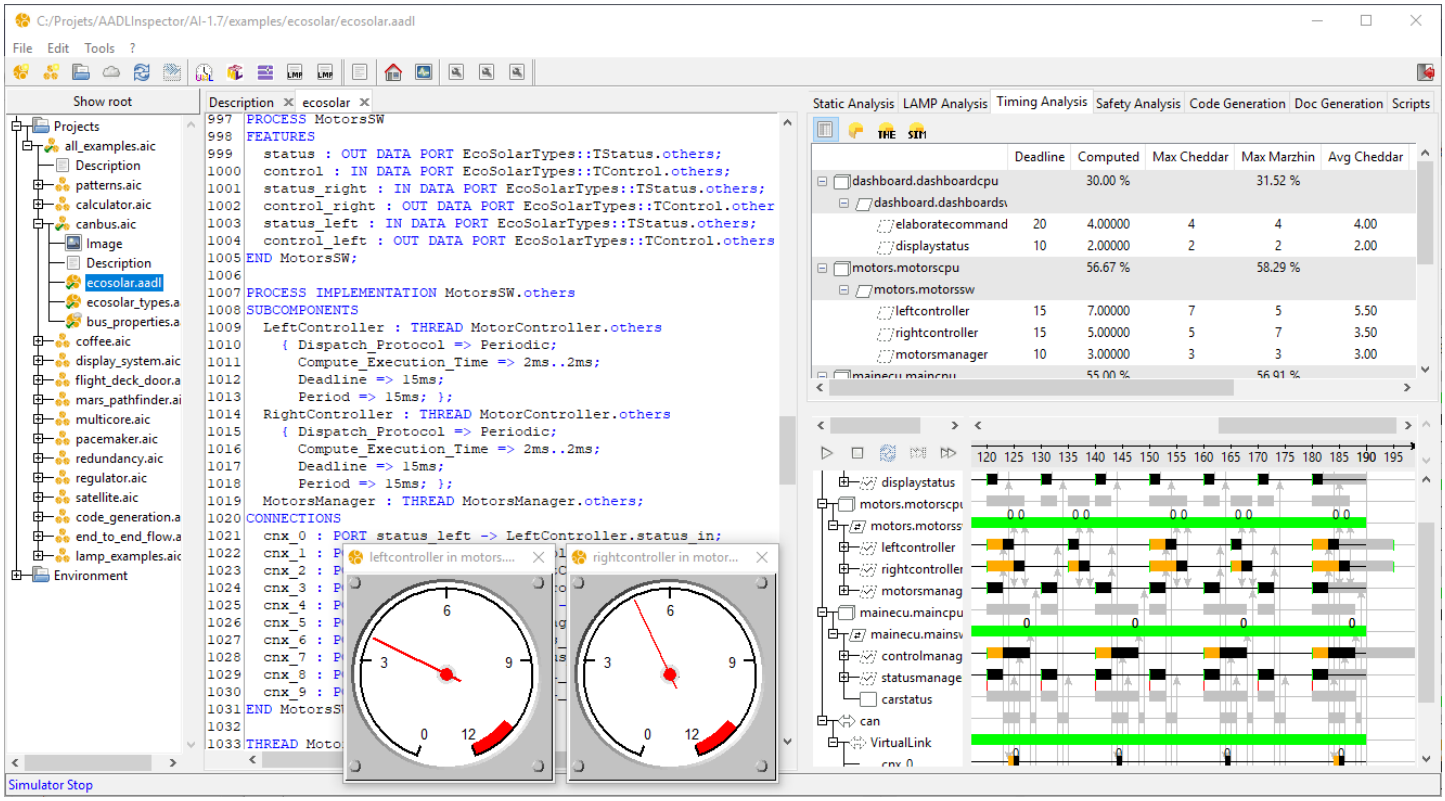


It doesn't include your model processing features yet ?

Thanks to the Logic Model Processing (LMP) technology that provides a powerful and formal way to connect remote tools, AADL Inspector can easily incorporate additional analysis or production plugins or customize the existing ones. All the appropriate support will be provided by our technical team to help you to introduce project specific rules checkers, to add a new AADL bridger to your favorite verification tool, or plug a target source code generator.

Static Analysis

AADL Inspector complies with the most recent version of the AADL standard, including its annexes (Data Modeling Annex, Behavior Annex, ARINC 653 Annex and Error Annex). For upwards compatibility, older AADL files are also accepted. Syntactic analysis, static semantics verification and model instanciation are implemented by two technologies: **Ocarina**[1] and **LMP**.



Timing Analysis

AADL Inspector also includes two Real-Time analysis tools and their respective AADL LMP bridger:

- **CHEDDAR**[2] for schedulability analysis (feasibility tests and static simulation over the hyper-period). It supports RM, DM, HPF, EDF and LLF scheduling of periodic tasks.
- **MARZHIN**[3] for event-driven dynamic simulation. It supports RM, DM, HPF and EDF scheduling of periodic and non-periodic tasks, with possible interaction with the external environment (pre-defined scenarios or user interaction).
- Response Time analysis and CPU Load estimate.

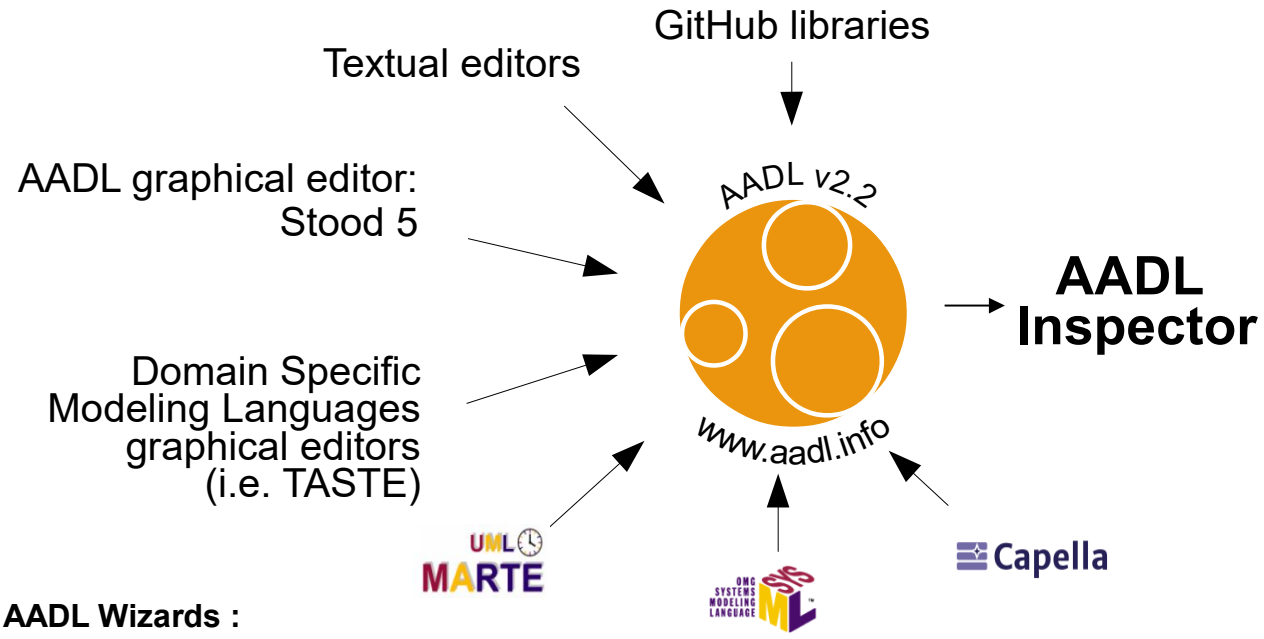
Custom Assurance Cases Tools

AADL Inspector provides the **LAMP** (Logic AADL Model Processing) engine to implement inline verification tools that are embedded within the AADL model as annexes. It uses the industry proven LMP technology. LMP model checkers have been qualified by Airbus for the DO 178 certification of A380 and A350 embedded software.

[1] Ocarina is developed by Telecom ParisTech, ISAE and ESA
[2] Cheddar is developed by the University of Brest
[3] Marzhin is developed by Ellidiss Technologies and Virtualys

Inter-operability :

Thanks to the AADL textual notation, AADL Inspector can easily be integrated into heterogenous development tool-chains. Such textual specifications can be produced by AADL graphical editors such as Stood, Domain Specific editors or even a basic text editor. In addition, AADL source code can be produced by model transformations from other modeling languages such as SysML or UML/Marte. A direct access to AADL models libraries available on GitHub is also provided. The time-lines generated by the simulators can be saved in VCD (Value Change Dump) format for further processing.



AADL Wizards :

AADL Inspector can also be used to dynamically modify your AADL project with data collected from the analysis tools, such as :
- Auto-format the AADL source text (pretty printer)
- Select the root system instance
- Set threads priorities according to the schedule policy
- Allocate threads onto available processors

Platform Requirements :

AADL Inspector runs as a standalone executable on PC Windows or Linux.
A recent Java Run-time Environment (1.8 or more) is required to run the Marzhin simulator.
Less than 25 Mb free disk space is needed

French office:
24 quai de la douane
29 200 Brest
France

+33 (0) 298 451 870

Ellidiss Technologies

www.ellidiss.fr

UK office:
Mountbatten Court
Worrall Street
Congleton
CW12 1DT
UK
+44 (0) 1260 291449