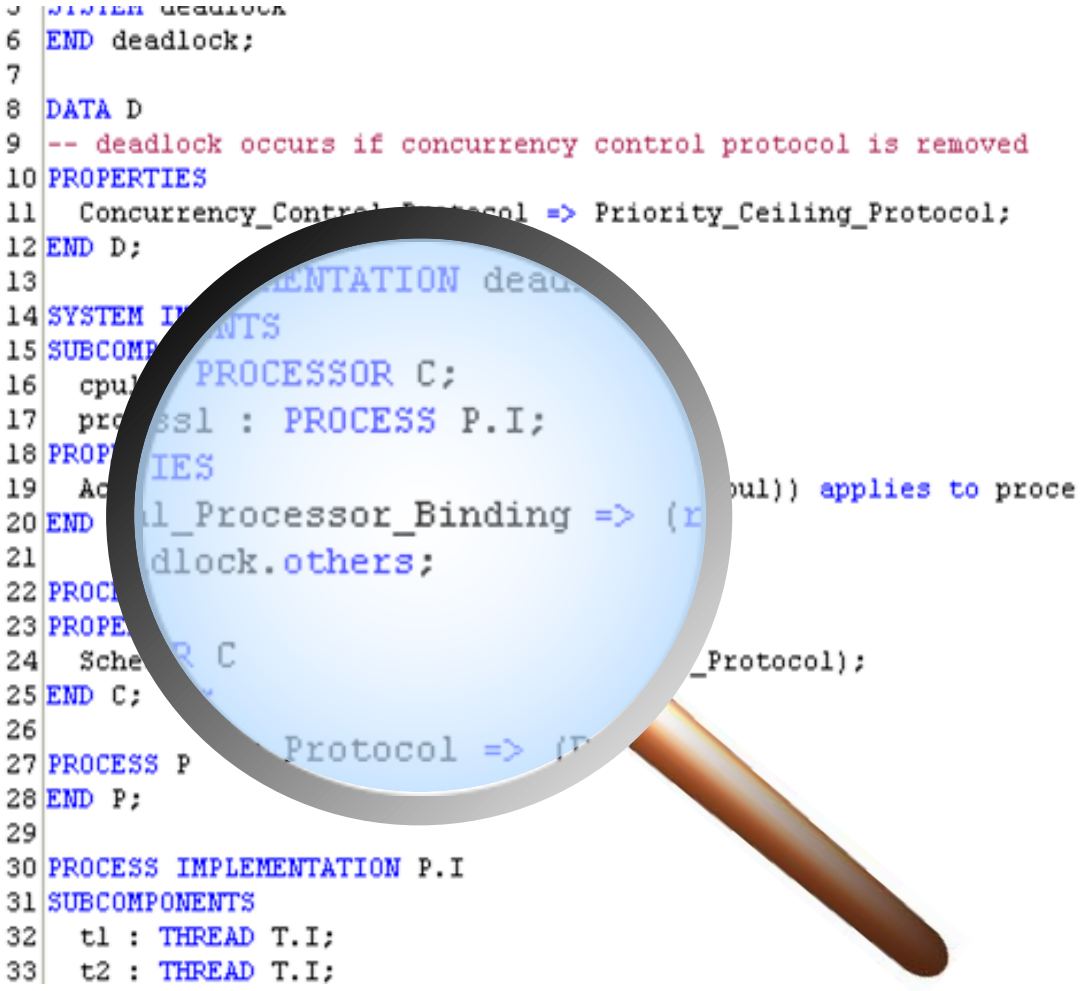
Ellidiss Technologies

AADL Inspector

Model processing framework for the
Architecture Analysis and Design Language



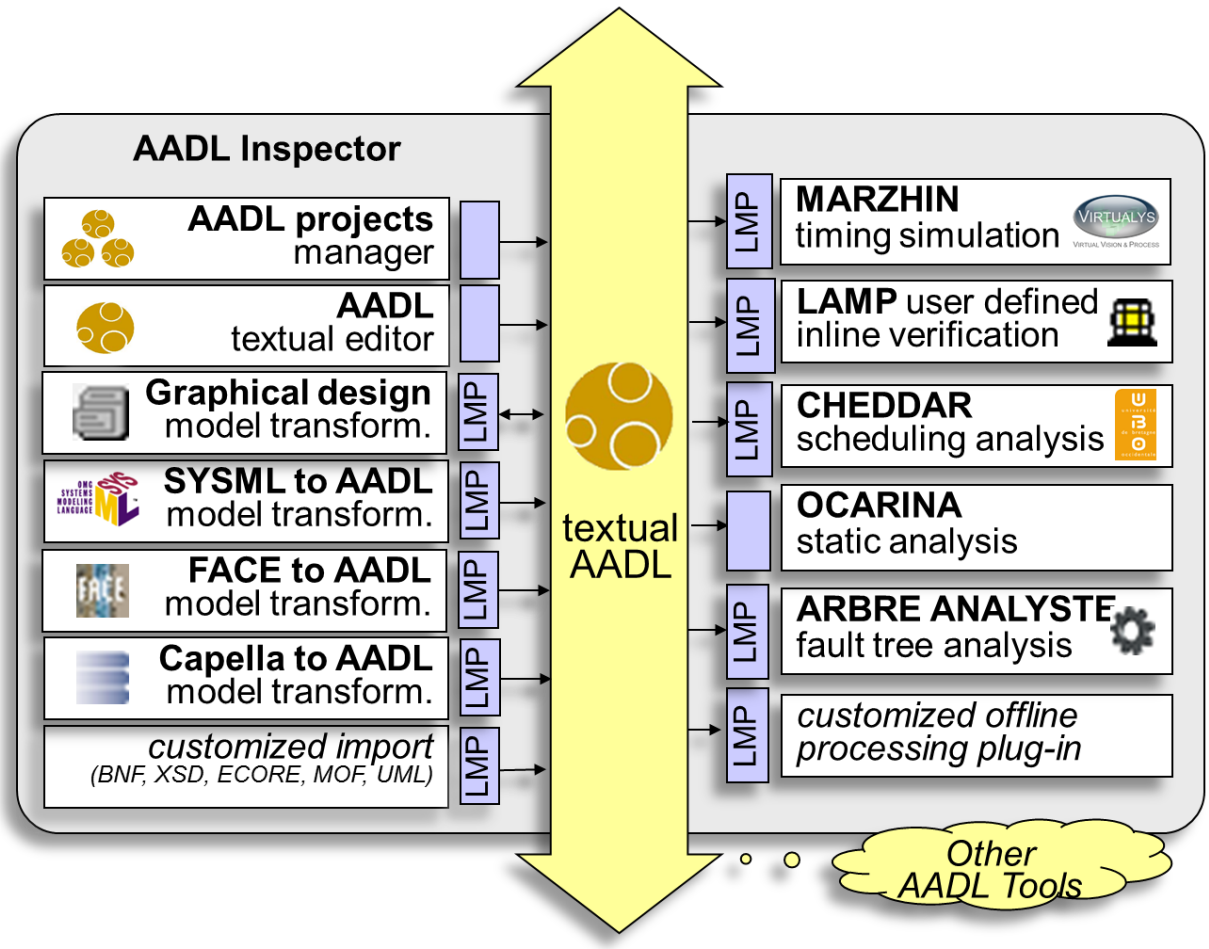
Real-Time

Safety

Security

AADL: designing scalable safe and secure real-time systems.

Modeling and early validation of software architectures is a key concern for embedded applications. The Architecture Analysis and Design Language (AADL) is a textual and graphical language dedicated to design and analysis of architectural models of applicative software and its execution platform. AADL Inspector can process AADL v2.3 (SAE AS-5506D) as well as its Behavior Annex and Error Annex extensions. The rich semantics of AADL enable the specification of advanced assurance cases sharing a common representation of the system and involving a variety of analysis domains including static properties, real-time, safety and security. Thanks to its textual syntax, AADL is scalable and can be used as a first-class front end modelling language for designing large scale software intensive systems or as a pivot format to implement software development toolchains.



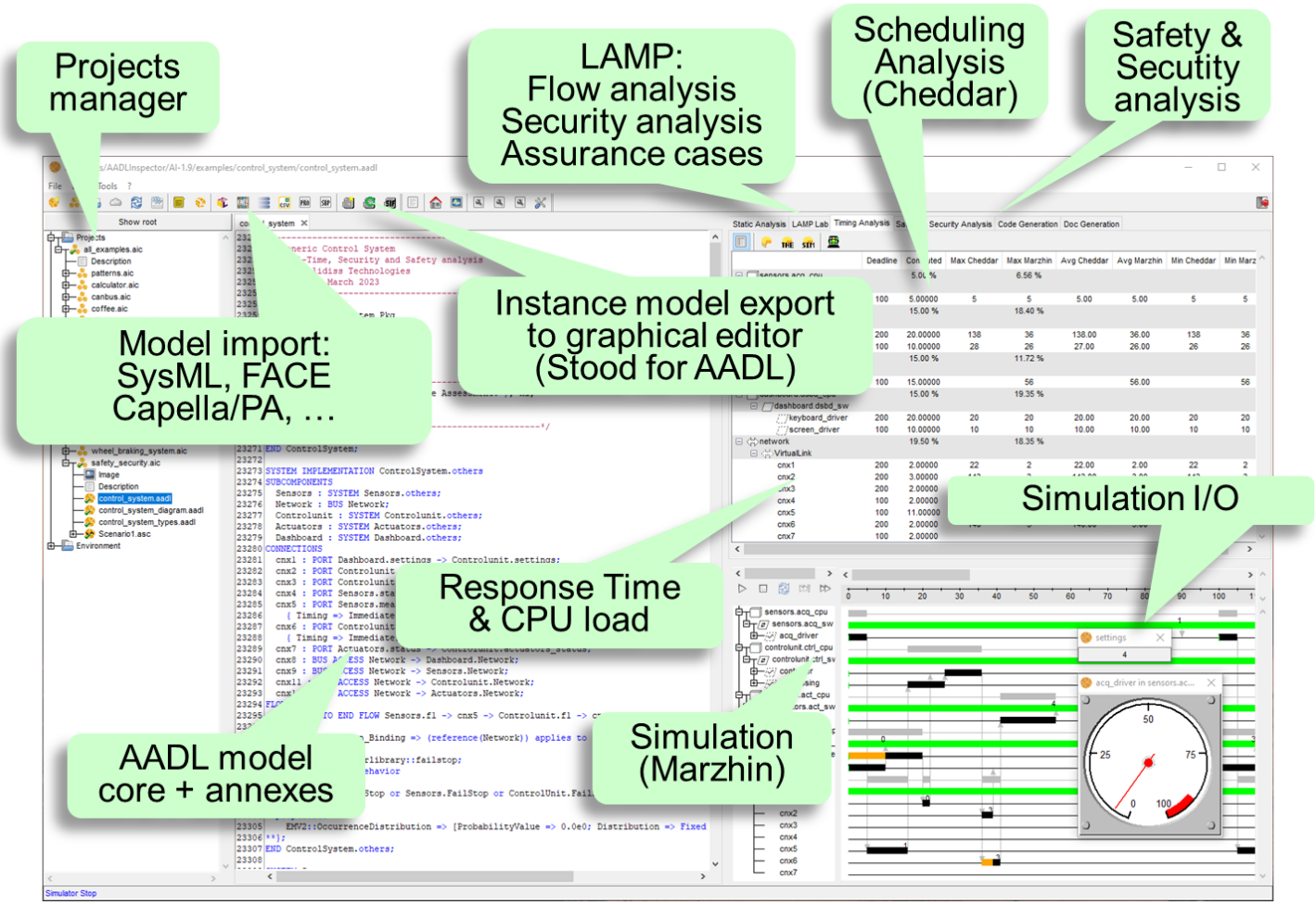
Toolchain integration:

Thanks to the AADL textual notation, AADL Inspector can easily be integrated into heterogenous development toolchains. AADL textual specifications can be produced by AADL graphical editors such as Stood for AADL, by Domain Specific Language editors or even by a basic text editor. In addition, AADL source code can be automatically generated by model transformations of system engineering modeling sources such as FACE™, SysML or Capella. A direct access to AADL model libraries that are available on GitHub is also provided. Additionally, the outcomes of the analysis tools are saved in XML, VCD (Value Change Dump) or Prolog formats for further processing.

AADL Inspector main features:

AADL Inspector is a light and standalone tool with an easy-to-use graphical user interface composed of three main parts:

- An AADL source file browser that allows for the definition of hierarchical projects and clear access to available libraries.
- A multi-files AADL text editor.
- A set of customizable tabs to activate and display the outcomes of processing tools.

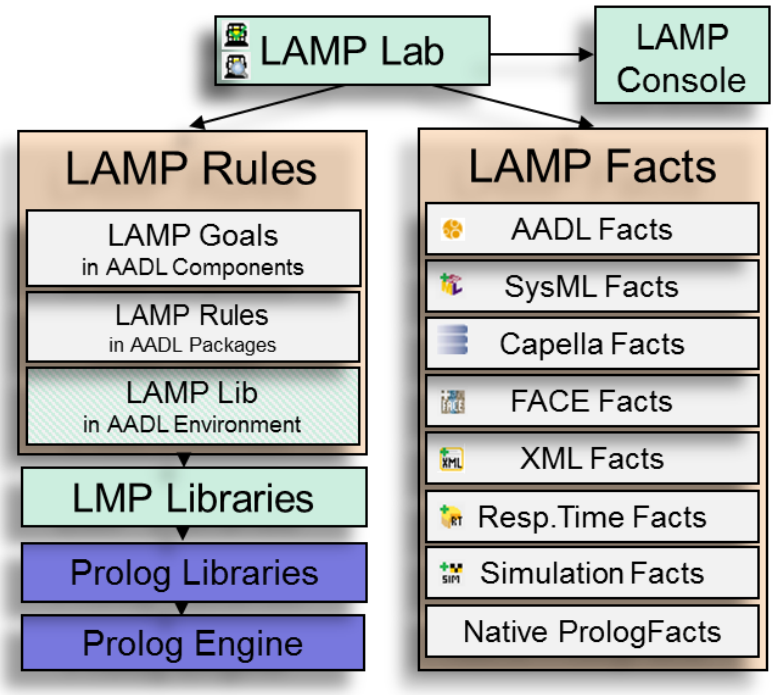


Processing tools included in AADL Inspector 1.8

- Import/Export tools:
  - SysML to AADL (*LAMP*)
  - FACE to AADL (*LAMP*)
  - Capella PA to AADL (*LAMP*)
  - AADL to HOOD (LAMP)
- Static analysis:
  - AADL syntactic analysis (*LMP & Ocarina*)
  - AADL legality rules (*LMP & Ocarina*)
  - AADL instantiation (*LMP & Ocarina*)
- Timing analysis:
  - Scheduling analysis (*Cheddar*)
  - Static simulation over the hyper-period (*Cheddar*)
  - Event-based simulation (*Marzhin*)
  - Scheduling Aware Flow Latency Analysis (*LAMP*)
- Safety analysis
  - Fault tree analysis (*Arbre Analyste*)
- Security analysis
  - Security rules checker (*LAMP*)

Customized processing rules:

The LAMP Laboratory included in AADL Inspector 1.9 is a powerful tool that enables the specification of customized processing rules and the implementation of complex assurance cases. LAMP stands for Logical AADL Model Processing and consists in giving access to standard Prolog programs inside AADL annex subclauses associated with AADL packages and components. Ellidiss Technologies provides a set of tools to parse AADL and XML/XMI based languages as well as Prolog libraries containing model accessors and utilities to help the user to develop his own exploration, constraints, transformation, and architectural reasoning rules. Use of an existing standard formal and declarative language such as Prolog brings almost unlimited model processing possibilities.



SysML to AADL:

Most of existing SysML to AADL transformation tools are based on the definition of a dedicated UML profile to represent AADL constructs. This approach seems natural; however, it forces the system engineer to “think” AADL and requires SysML tool specific customizations.

On the contrary, our solution takes plain SysML XMI input files and applies LAMP transformation rules to build an AADL model. Source code of the transformation template is provided and can be customized to fit corporate or project specific SysML to AADL mapping.

A similar approach can be applied to any other model transformation.

Requirements:

AADL Inspector runs as a standalone executable on PC Windows and Linux.
A recent Java Run-time Environment (1.8 or more) is required to run the Marzhin simulator.
End-user licenses can be issued for commercial, academic or evaluation purposes.
Less than 80 Mb free disk space is needed to install the product.

Ellidiss Technologies

24 quai de la douane
29200 Brest
Brittany, France
+33 (0)298 451 870

Acknowledgements:

Cheddar is developed by the University of Brest
Ocarina is developed by Telecom ParisTech, ISAE and ESA
Marzhin is developed by Ellidiss Technologies and Virtualys
Arbre Analyste is developed by Emmanuel Clément
FACE is a trademark of the Open Group

web site:
www.ellidiss.com
contact:
aadl@ellidiss.com