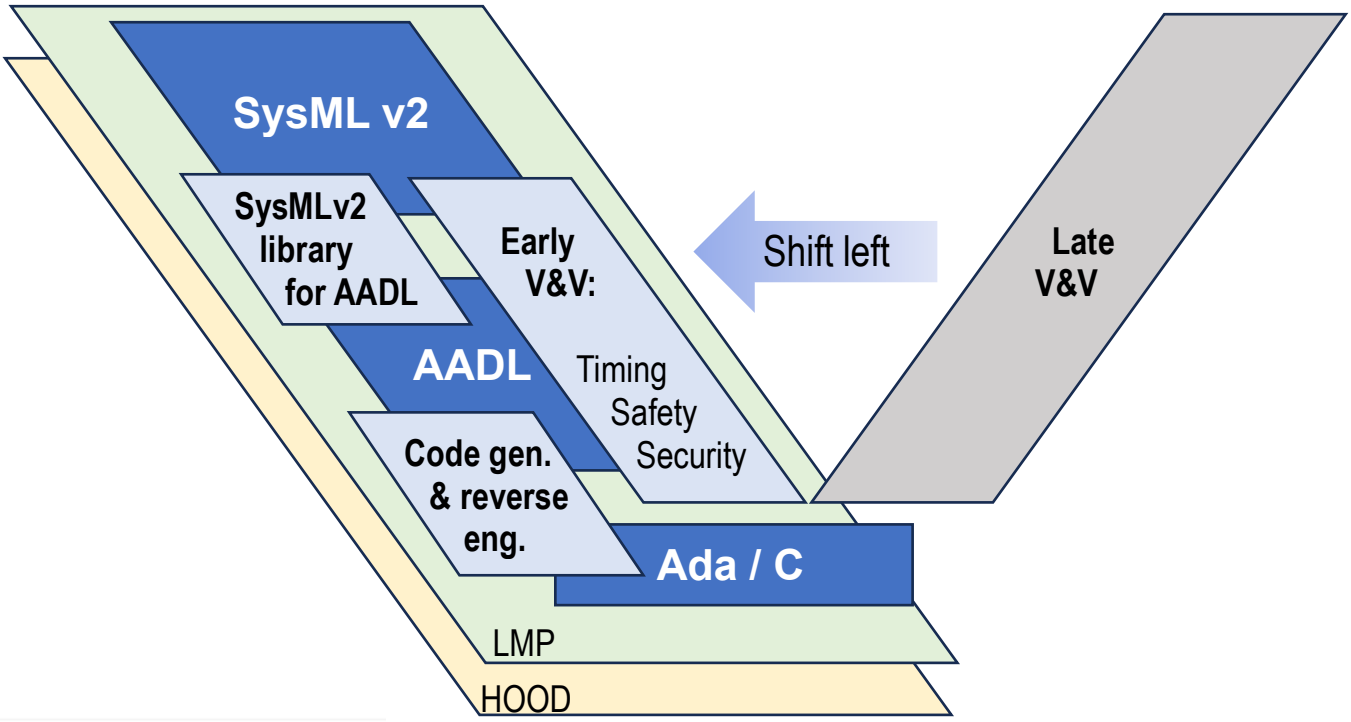
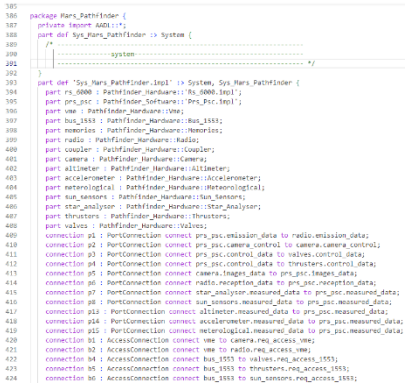
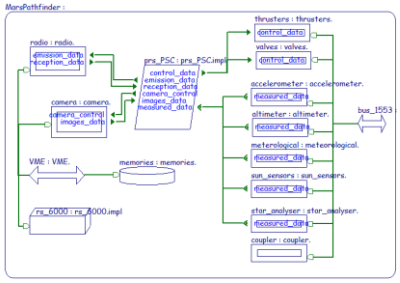
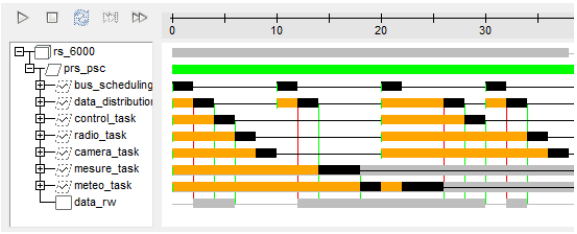
Ellidiss Technologies

Model Driven Engineering Solutions for Software Intensive Real-Time Safety Critical Systems









Textual Editors :
VS-Code extensions
for AADL and SysMLv2

Graphical Editors :
Stood
for HOOD and AADL

Timing Analysis
and Simulation :
AADL Inspector

Ellidiss Technologies
24 quai de la douane, 29200 Brest, Brittany, France
www.ellidiss.com
info@ellidiss.com

SysMLv2: the new emerging OMG standard

Version 2 of the System Modeling Language (SysML) is a general-purpose modeling language to support Model-Based Systems Engineering (MBSE) processes. It can be used to build models representing various facets of a system. Compared to its predecessor, this new version mainly differs by the availability of a textual notation in addition to the graphical one, and a language extension mechanism supported by model libraries instead of meta-model add-ons.

As a member of the OMG System Modeling Community, Ellidiss contributes to the elaboration of an extension library to describe AADL real-time safety critical systems with SysMLv2. The goal is to provide a smooth path between system and software engineering activities and to give SysMLv2. users access to existing AADL model processing tools. The company progressively enhance his existing tools to support SysMLv2.

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

The Architecture Analysis and Design Language (AADL) is textual and graphical modeling language aiming at designing and analyzing real-time software with their execution platforms. The rich semantics of AADL enable the development of early verification and code generation features.

Ellidiss is an active member of the SAE AS-2C standardization committee that specifies the AADL language. The company has developed an AADL graphical editor (Stood for AADL) and an AADL model processing framework (AADL Inspector) encompassing static, real-time, safety and security analysis tools, as well as a graphical timing simulator.

HOOD: mastering the complexity of large software projects

The Hierarchical Object Oriented Design (HOOD) method was initially created to provide a Model Driven solution on top of the Ada programming language for large scale projects. It has later been applied to the most important avionics and space projects in Europe such as the Airbus civilian aircrafts and the Eurofighter Typhoon fighter. HOOD relies on a rigorous top-down decomposition approach guided by a “low-coupling” principle. This leads to rigorous, efficient and maintainable architectural designs and production of consistent code and documentation.

Ellidiss provides long-term support for two HOOD tools that are widely used in European aerospace industry (CP-Hood and Stood for HOOD). The company has also the ability to apply the HOOD principles and methodology to the most recent modelling and programming languages.

LMP: symbolic AI solution for model processing

Logic Model Processing (LMP) applies the principles of Logic Programming to Model Driven Engineering. In a few words, each model is expressed by a list of Prolog facts and each processing function is implemented by a list of Prolog rules. Feeding a Prolog inference engine with relevant facts and rules will produce the awaited result (e.g., model exploration, analysis, verification, transformation). This generic approach can be applied to heterogeneous model sources and provides a sound way to develop efficient model processing tools.

Ellidiss has set up an extensive toolbox to support the LMP approach and uses it as a corner stone for his modeling and analysis tools. This toolbox contains many modeling and programming language parsers and printers as well as a large library of processing rules, from low level utilities to advanced exploration or transformation functions. The LMP toolbox currently includes supports for: AADL, Ada, C, Capella, CSV, FACE, HOOD, SDL, SysMLv1, SysMLv2, XMI, and XML.