An Architectural Framework for Virtual Prototyping of Next Generation Combat Vehicles

Chris Paredis (PI), Yue “Sophie” Wang (co-PI), Maziar Mahani, Ryan Colletti, Huzefa Dossaji, Alessandra “Ali” St. Germain, Denise Rizzo (GVSC), Ahsan Qamar (Ford)

Motivation

- Model-based engineering allows for faster, broader, and more thorough exploration of design alternatives, but the time and effort needed to create a virtual prototype is still considerable.
- Can we capture domain knowledge in a reference model so that this cost can be reduced?

Fundamental Research Questions

- To what extent can vehicle architectures be composed from model components that are configured based on a variability model?
- Is there sufficient parallelism between the vehicle architecture and the corresponding simulation models so that a variability modeling approach can be used to configure both simultaneously?

Objectives

- Explore the use of variant modeling in SysML to facilitate the generation of vehicle architecture models across a broad range of different architectures (from wheeled electric cars to tracked diesel-powered tanks).
- Generate simulation models and perform model checking for single variant architectures obtained by slicing a 150% model.

Approach

- Model vehicle architectures in SysML, describing the physical structure of the system, the Simulink-based control algorithms, and the ROS-based autonomy stack.
- Create a 150% reference model using a SysML variant modeling plug-in.
- Select desired features in a feature model to define a variant.
- Obtain a single variant SysML model by slicing the 150% model.
- The single variant SysML model describes the architecture for the structure, behavior, and the corresponding analysis models
- The analysis models are configured from the SysML architecture

Results

- Proof-of-concept reference architecture and feature model
- Best practice structural decomposition organization and patterns for NGCV design space
  - Novel 150% and encapsulation variant modeling patterns
  - SysML profile and variable reference architecture for ROS-based autonomous software
  - Functional decomposition of ECU behavior
    - Many-to-one mapping to simulation modules
    - Module interface management through variant model

Future Work

- Automated state machine model checking
- Automated composition of Simulink model
- Demonstration through comprehensive virtual prototype example