Architecture-Driven Assurance for Model-Based Systems Engineering

Submitted by Katie Dey on Sat, 03/10/2018 - 2:58pm. Contributor: Darren Cofer

BIO
Darren Cofer is a Fellow in the Rockwell Collins Advanced Technology Center. He earned his PhD in Electrical and Computer Engineering from The University of Texas at Austin. His principal area of expertise is developing and applying advanced analysis methods and tools for verification and certification of high-integrity avionics systems. His background includes work with formal methods for system and software analysis, the design of real-time embedded systems for safety-critical applications, and the development of nuclear propulsion systems in the U.S. Navy. He has served as the principal investigator on numerous sponsored research projects for NASA, AFRL, and DARPA.

ABSTRACT
Continued advances in aircraft performance and safety will be based on software-intensive systems with frightening levels of complexity. As the growth in complexity of aircraft systems continues, development costs and schedules have increased dramatically. Systems engineering tools and methods have not sufficiently adapted to the demands of today’s complex systems and as a result aircraft systems are typically late, over budget, and often have reduced capabilities compared to original goals. New model-based systems engineering (MBSE) tools that leverage advanced automation and analysis capabilities are needed to manage this complexity.

This talk describes our Architecture-Driven Assurance approach to MBSE. Our approach is based on comprehensive use of formal methods throughout the development process to ensure that vulnerabilities and design defects are eliminated from critical systems. We have developed integrated tools for architectural modeling, analysis, and synthesis to make this approach practical and effective. It is based on the following foundations: System Architecture modeled in AADL - We create analyzable models of the system architecture to support "virtual integration" throughout the development and sustainment life cycle.

Architecture model is correct - We apply a variety of tools to verify that the architecture implements system-level requirements. This includes: compositional reasoning about system behavior based on formal contracts added to the AADL model; a logic and supporting tool for generating assurance cases from the AADL model structure; model-based safety analysis to evaluate behavior in the presence of faults; and simulation and automatic test case generation for validation of the model and, eventually, the implementation.

Software components are correct - Component contracts are checked for consistency and realizability. Component contracts are exported to component development environments (e.g., Simulink) for implementation and verification.

System does what the model says - We target the formally verified kernel (seL4) to guarantee isolation between components, and to ensure that there are no information flows other than those explicitly defined in the architecture.

Software implementation corresponds to model - Our Trusted Build process automatically generates implementation code from architecture model, component specifications, and kernel/OS build system.