Formal Methods at Scale

Darren Cofer and Matt Wilding
Collins Aerospace

Large aerospace/defense systems are not created in a monolithic fashion, but are instead compositions of many smaller modules that can be more easily comprehended by developers. Design choices are expressed using concepts from an engineering toolbox containing understood components. Engineers use informal methods - visio diagrams, whiteboard diagrams, spreadsheets, etc - to develop and communicate system concepts and their design choices and rationales. These choices are documented as requirements, architectures, or safety/security analyses to meet certification requirements. As development progresses to an implemented system, much of the conceptual development information is expressed in documents that are difficult for automated tools to consume and quickly becomes stale.

Replacement of informal methods with formal methods is needed to avoid the high cost of verification and certification activities, as well as the extremely high cost (both financial and human) of design defects that escape into service. But radical change to how systems are developed must support the engineering process for how large systems are actually created - by engineers using time-tested engineering concepts. Model-based system engineering (MBSE) tools and languages offer a common system design approach that can be used to support engineers employing these conventional design idioms. The challenge is to extend MBSE methods with formal languages and automation that make the development process efficient, rigorous, and repeatable.

MBSE can be used throughout the lifecycle to drive safety and security analysis, system development, verification, infrastructure code generation, and certification evidence. Much that is currently built by engineers using informal system descriptions could be synthesized directly from formal system models, enhancing reliability and avoiding the quick artifact obsolescence inherent in current approaches.

Research should focus on adapting MBSE methods to various domains, and building formal synthesis and analysis infrastructure to maximize MBSE benefits in high-assurance domains. Enabled capabilities would include performing safety analysis by calculating fault model cut sets, synthesizing correct-by-construction security-enforcing components such as filters and monitors, generating certification artifacts such as test suites, deriving platform communication configurations, and performing automated analysis of security and safety flaws. The use of MBSE with associated automation will lead to standardized approaches for integrating system design with system safety and cybersecurity.