Assessment and Evolution of Safety-Critical Cyber-Physical Product Families

Invited Talk Abstract

Leon Moonen

Simula Research Laboratory
Norway
leon.moonen@computer.org

The research presented in this talk is part of an ongoing industrial collaboration with Kongsberg Maritime (KM), one of the largest suppliers of maritime systems worldwide. The division that we work with specialises in computerised systems for safety monitoring and automatic corrective actions on unacceptable hazardous situations. The overall goal of the collaboration is to provide our partner with software analysis tooling that provides source based evidence to support cost-effective software certification of evolving systems. In particular, we study a family of complex safety-critical embedded software systems that connect software control components to physical sensors and mechanical actuators.

A frequently advocated approach to manage the development of such complex software systems is to compose them from reusable components, instead of starting from scratch. Components may be implemented in different programming languages and are tied together using configuration files, or glue code, defining instantiation, initialisation and interconnections. Although correctly engineering the composition and configuration of components is crucial for the overall behaviour, there is surprisingly little support for incorporating this information in the static verification and validation of these systems. Analysing the properties of programs within closed code boundaries has been studied for some decades and is well-established.

Moreover, sharing components between software products introduces dependencies that complicate maintenance and evolution: changes made in a component to address an issue in one product may have undesirable effects on other products in which the same component is used. Therefore, developers not only need to understand how a proposed change will impact the component and product at hand; they also need to understand how it affects the whole product family, including systems that are already deployed. Given that these systems contain thousands of components, it is no surprise that it is hard to reason about the impact on a single product, let alone on a complete product family. Conventional impact analysis techniques do not suffice for large-scale software-intensive systems and highly populated product families, and engineers need better support to conduct these tasks.

In the talk, we will discuss the techniques we developed to support analysis *across* the components of a heterogeneous component-based system. We build upon OMG's Knowledge Discovery Metamodel to reverse engineer fine-grained

homogeneous models for systems composed of heterogeneous artifacts. Next, we track the information flow in these models using slicing, and apply several transformations that enable us to visualise the information flow at various levels of abstraction, trading off between scope and detail and aimed to serve both safety domain experts as well as developers. These techniques are implemented in a prototype tool-set that has been successfully used to answer software certification questions of our industrial partner. In addition, we discuss our ongoing research to build recommendation technology that supports engineers with the evolution of families of safety-critical, software-intensive systems. This technology builds on extensions of the previously discussed techniques to systematically reverse engineer abstract representations of software products to complete software product families, new algorithms to conduct scalable and precise change impact analysis (CIA) on such representations, and recommendation technology that uses the CIA results and constraint programming to find an evolution strategy that minimises re- certification efforts.