Refactoring in Requirements Engineering: Exploring a methodology for formal verification of safety-critical systems

Oisín Sheridan
Supervisor: Rosemary Monahan

Abstract

Despite being of utmost importance, guaranteeing the trustworthiness of safety-critical software presents a significant challenge for developers. One solution is the use of formal methods - mathematically-based techniques which can verify correctness through proof of the system's properties and exhaustive checks over its state space. These methods are widely applicable to many industrial applications, and are seeing an uptake in many high-profile companies.

As part of the VALU3S EU project, we have developed a methodology for formal verification which consists of three phases and incorporates two parallel verification workflows. In one workflow, the existing model of the system is directly verified against the requirements it is expected to meet; in the other, the existing model and requirements are used to manually construct a new model of the system's behavior in a formal language. The use of multiple tools and techniques is beneficial as different formal methods are better suited to verify different aspects of the system under consideration. We have explored both verification strategies on our case study of the software controller for an engine of a civilian aircraft.

The core of this methodology is the Formal Requirements Elicitation Tool (FRET), an open-source tool developed by NASA which allows the user to specify requirements in a structured language called FRETISH, which can then be automatically translated to formal languages such as Linear Temporal Logic. Our methodology is driven by the formalization of requirements, and thus we are also exploring how improvements in the requirements engineering process can improve the verification process as a whole. We have identified the benefits for our existing case study of incorporating refactoring techniques into FRET, and we are in the process of developing our own fork, called Mu-FRET, and investigating the impacts of refactoring on the rest of the methodology.