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

Oisín Sheridan
Supervisor: Rosemary Monahan

Abstract

Guaranteeing the trustworthiness of autonomous safety-critical software presents a significant challenge for developers. Industry often relies on manual testing and simulation to verify such systems, but these techniques are time-consuming, expensive, and error-prone. These problems can be solved through 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.

During the VALU3S project, we developed a methodology for formal verification that utilizes a number of tools and techniques to verify safety-critical systems. The core of this methodology is the Formal Requirements Elicitation Tool (FRET), an open-source tool developed by NASA designed for formalizing natural-language requirements. While formalizing the requirements for VALU3S, we found that applying refactoring techniques to the requirements in FRET could make them more readable and reduce repetition across the set, making the requirements easier to work with. Thus, the aim of this research became to explore the theory of refactoring requirements and implement it in FRET in a way that could be used in a verification methodology like the one that we developed.

We identified four refactorings from existing literature that we felt could be applied to FRETish requirements. The goal of this research is to implement these refactorings into our fork of FRET, called MuFRET, and develop a rigorous theory that allows them to be applied in as many cases as reasonably possible. The refactorings should preserve the underlying meaning of the requirements while making them easier to work with, and integrate with the existing functionality within FRET for formal translation, thus improving the verification and validation (V&V) process overall. In doing so, we will develop an improved approach to V&V that allows both traceability so that detected software errors can be traced back to their source, and scalability to systems of greater size and complexity. At present, two of the four refactorings have been implemented in MuFRET, with further improvements planned along with the two additional refactorings.