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

Oisín Sheridan
Supervisor: Rosemary Monahan

June 2025

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.

The formal verification of safety-critical software requires formal requirements. Software requirements are often written in natural-language, which then needs to be translated into a formal language for use in verification. However, as a requirements set becomes larger and more defined over the course of a project, an ever-increasing amount of work is required to ensure consistency, readability and traceability across the set. Often, dependencies and duplication of information between requirements emerges, which then requires additional work from engineers to update all of the affected requirements when changes need to be made.

We propose that these issues can be mitigated by applying refactoring to requirements. Refactoring is a software engineering technique where code is reorganized to improve its internal structure without changing its behavior; in the case of requirements, we can reduce duplication of information and improve the readability of the requirements without changing the behavior that the requirements specify for the system.

This project aims to provide a working implementation of requirements refactoring in Mu-FRET, a fork of the Formal Requirements Elicitation Tool (FRET) which allows for requirements to be written in a structured natural-language, which is then translated automatically into temporal logic. In addition, we will provide a theory of refactoring that can be generalized to other requirements languages.