Detection of Feature Interactions in Automotive Active Safety Features

Abstract

With the introduction of software into cars, many functions are now realized with reduced cost, weight and energy. The development of these software systems is done in a distributed manner independently by suppliers, following the traditional approach of the automotive industry, while the car maker takes care of the integration. However, the integration can lead to unexpected and unintended interactions among software systems, a phenomena regarded as *feature interaction.* This dissertation addresses the problem of the automatic detection of feature interactions for automotive active safety features. Active safety features control the vehicle's motion control systems independently from the driver's request, with the intention of increasing passengers' safety (*e.g.,* by applying hard braking in the case of an identified imminent collision), but their unintended interactions could instead endanger the passengers (*e.g.,* simultaneous throttle increase and sharp narrow steering, causing the vehicle to roll over). My method decomposes the problem into three parts: (I) creation of a definition of feature interactions based on the set of actuators and domain expert knowledge; (II) translation of automotive active safety features designed using a subset of MATLAB's STATEFLOW into the input language of the model checker SMV; (III) analysis using model checking at design time to detect a representation of all feature interactions based on partitioning the counterexamples into equivalence classes. The key novel characteristic of my work is exploiting domain-specific information about the feature interaction problem and the structure of the model to produce a method that finds a representation of all different feature interactions for automotive active safety features at design time.

My method is validated by a case study with the set of non-proprietary automotive feature design models I created. The method generates a set of counterexamples that represent the whole set of feature interactions in the case study. By showing only a set of representative feature interaction cases, the information is concise and useful for feature designers. Moreover, by generating these results from feature models designed in MATLAB's STATEFLOW translated into SMV models, the feature designers can trace the counterexamples generated by SMV and understand the results in terms of the STATEFLOW model. I believe that my results and techniques will have relevance to the solution of the feature interaction problem in other cyber-physical systems, and have a direct impact in assessing the safety of automotive systems.