Sebastian Oster: **Feature Model-based Software Product Line Testing**

Promotion: Technische Universität Darmstadt, Fakultät Elektrotechnik und Informationstechnik
Erstgutachter: Prof. Dr. Andy Schürr, Technische Universität Darmstadt
Zweitgutachter: Prof. Dr. Ursula Goltz, Technische Universität Braunschweig
Datum der Prüfung: 16. Dezember 2011
Veröffentlichung: http://tuprints.ulb.tu-darmstadt.de/2881/

Kurzfassung:

Software Product Line (SPL) engineering is a popular approach for the systematic reuse of software artifacts across a very large number of similar products. SPLs are gaining widespread acceptance and various domains already apply SPL engineering successfully to address the well-known needs of the Software Engineering community, such as increasing quality, saving costs for development and maintenance, and decreasing time-to-market. The central aspect of systematic reuse is the concept of variability often leading to an enormous number of possible products In the automotive sector, we are increasingly encountering a situation where a single electronic control unit (ECU) may be instantiated in at least 10,000 different ways, and the software running on a network of more than 50 ECUs in a single car may exist in millions of different configurations. As a result, we are confronted with a world where any instance of a certain brand of car possesses a unique configuration of the embedded software of all its ECUs. At a first glance, SPL engineering seems to result in a major benefit for the automotive sector as well as for other industrial domains, allowing the combination of mass production and product customization. The reverse side of the coin is the challenge to assure the quality of each derivable product of the SPL e.g. via testing activities. Testing all products of an SPL individually is generally not feasible. This thesis contributes an approach to significantly reduce the test effort for SPL testing. Faults are likely to be revealed at execution points where features exchange information with other features or influence one another. Therefore, a criterion for test adequacy is to cover as many interactions among different features as possible, thus, increasing the probability of finding bugs based on feature interaction. We present a novel approach to generate a representative set of products of the SPL required for comprehensive coverage of feature interactions. The features of the feature model are combined in products using a combinatorial strategy assuring a certain degree of feature interaction coverage. For this purpose we introduce a graph transformation-based algorithm to translate the feature model into a binary constraint satisfaction problem and also an algorithm combining constraint solving techniques with a feature combination strategy to generate the representative set of products. A mapping between the feature model and a reusable test model allows for generating test cases for each product automatically. We implemented our approach as a tool chain and applied it to three different industrial SPLs for evaluation purposes. The results suggest that with our approach higher coverage of feature interactions is achieved at a fraction of cost and time when compared with the state-of-the-art approach of testing all derivable products.