Feature Model-based

Software Product Line Testing

Oster, Sebastian. Feature Model-based Software Product Line Testing. Diss. Technische Universität, 2012.

4.2.2 Combinatorial Interaction Testing

Another possibility is to generate a subset of configurations, using combinatorial design [McG01]. The commonalities and variability within an SPL are frequently represented by features that can be interpreted as parameters in the SPL engineering process. One of the best-known applications of combinatorial testing is the pairwise testing approach. This method is based on the assumption that the majority of faults originate from a single parameter value or are caused by the interaction of two values [SM98]. This idea can be transferred to features. Combinatorial testing can be used to cover a certain T-wise feature interaction. McGregor initially introduced combinatorial testing to SPLs in [McG01]. However, he neither describes how combinatorial testing may be applied to SPLs nor how variability models, such as feature models or OVMs, can be mapped onto an appropriate representation to apply existing combinatorial testing algorithms. Gustafsson introduced an algorithm generating a subset of products so that every single feature is covered at least once [Gus07]. The disadvantages of such an approach realizing 1-wise combination are obvious. There is no guarantee that a feature functions as expected on its own nor does this approach take feature interactions into account. Cohen et al. use the OVM approach to model the variable and common parts of the SPL, which are mapped onto a relational model. This relational model serves as a semantical basis for defining coverage criteria for the SPL under test [CDS06]. Furthermore, Cohen et al. describe the development of combinatorial interaction testing (CIT) that achieves a desired level of coverage. In [CDS07], the authors use the CIT approach to systematically select products that should be tested. The approach described in [PSK+10] is similar to the one of Cohen et al.. The significant difference is that Perrouin et al. utilize SAT-solvers and do not use a relational model. Nevertheless, the output of the presented algorithm is not deterministic, due to random components that we will explain in the evaluation chapter of this thesis. The number of found products may vary strongly, due to this random component. These aforementioned methods provide feature oriented approaches for generating a representative set of configurations for an SPL. However, none of the approaches actually provides a test case generation process for the generated set of configurations. Furthermore, the application of these methods is rather complicated and not supported by a tool chain combining variability modeling and combinatorial configuration selection. In this thesis, this will be considered as well.