

Abstract

A software product line (SPL) is a family of closely related software systems which capitalizes on the reusability and variability of the software products. An SPL can be modeled using a feature model, a tree-like structure from which all the configurations of the SPL can be derived. Large projects such as an SPL require long-term planning, and plans for SPLs may also be defined in terms of feature models, called feature model evolution plans (FMEP). An FMEP gives information about what a feature model looks like at each stage of the plan.

As business requirements often change, FMEPs should support intermediate change. Such changes may cause paradoxes in an FMEP, e.g. a node left without a parent, making the plan impossible to realise. The complex nature of FMEPs makes detecting paradoxes by hand impractical. Current tools exist to validate FMEPs, but require analysis of the entire plan even when a modification affects only small parts of it. For larger FMEPs, this is inefficient. Thus, there is a need for a method which detects such paradoxes in a more efficient way.

In this thesis, we present a new representation for FMEPs, called a temporal feature model (TFM). This representation enables local validation, by which we mean validating only the parts of the plan that are affected by the change. We further define operations for updating a TFM, and methods for detecting paradoxes resulting from an operation. Moreover, we give a proof of correctness for the method and an implementation as proof of concept.

Using these methods, it is possible to create an efficient soundness checker for modification of FMEPs. This may be used as basis for an SPL planning tool and will contribute to productivity.