Automated diagnosis of feature model configurations

Jules White, David Benavides, Douglas C. Schmidt, Pablo Trinidad, Brian Dougherty, and Antonio Ruiz Cortés

Dpto. de Lenguajes y Sistemas Informáticos
University of Seville
Av. de la Reina Mercedes S/N, 41012 Seville, Spain
{benavides, ptrinidad, aruiz}@us.es

Abstract. Software product-lines (SPLs) are software platforms that can be readily reconfigured for different project requirements. A key part of an SPL is a model that captures the rules for reconfiguring the software. SPLs commonly use feature models to capture SPL configuration rules. Each SPL configuration is represented as a selection of features from the feature model. Invalid SPL configurations can be created due to feature conflicts introduced via staged or parallel configuration or changes to the constraints in a feature model. When invalid configurations are created, a method is needed to automate the diagnosis of the errors and repair the feature selections. This paper provides two contributions to research on automated configuration of SPLs. First, it shows how configurations and feature models can be transformed into constraint satisfaction problems to automatically diagnose errors and repair invalid feature selections. Second, it presents empirical results from diagnosing configuration errors in feature models ranging in size from 100 to 5,000 features. The results of our experiments show that our CSP-based diagnostic technique can scale up to models with thousands of features. [1]

1 Summary

A feature model is a compound of features that are connected with tree–like relationships and cross-tree constraints like the ones presented in Figure 1 that presents a summary of our contribution in this paper. In the top of the Figure we show an example of a feature model that represent the possible valid configurations of a product line of software to be embedded in a family of cars. Features are represented as boxes and grey features represent a configuration in a feature model. In the configuration shown in the top of the figure there is contradiction because feature 4 and feature 6 can not be at the same time in the

[1] The original paper is: J. White, D. Benavides, D.C. Schmidt, P. Trinidad, B. Dougherty, A. Ruiz-Cortes, Automated diagnosis of feature model configurations, Journal of Systems and Software, Volume 83, Issue 7, SPLC 2008, July 2010, Pages 1094-1107 http://dx.doi.org/10.1016/j.jss.2010.02.017

model according to the current relationships. Before presenting this paper, there was no technique to correct flawed configurations as the one presented here.

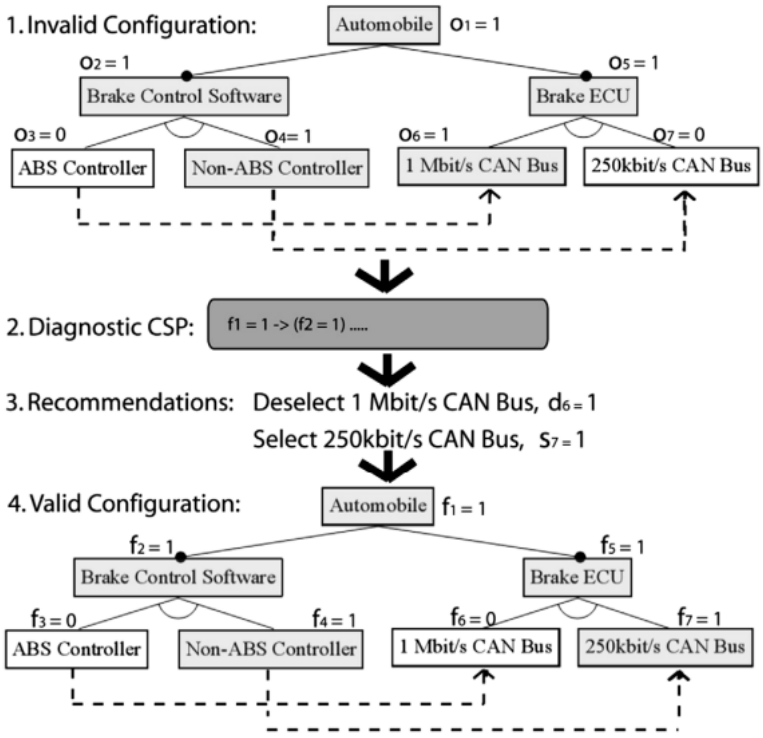

Fig. 1. Summary of the proposal

We used constraint programming techniques to diagnose and repair flawed configurations and constructed a prototype tool to perform this task that we called CURE (Configuration Understanding and REmedy). CURE translates a feature model into a constraint satisfaction problem inspired by our own existing research and provide possible changes to be done in the wrong configuration to repair it. For instance, in the Figure, the system recommends to deselect feature 6 and select feature 7. This makes the valid configuration of step 4. Doing this type of diagnosis and repairing is not easy, specially in large–scale feature models. Our implementation showed the feasibility of the approach in random situations since it can scale up to models with thousands of features. As current work, we are working in extending CURE to propose changes not only in the configuration but also in the feature model is needed. For instance, in this case, CURE could recommend removing the dependency between feature 4 and 7 which would also repair the problem but from another point of view.