

Tool supported error detection and explanations on feature models

Jesus García-Galán
Dept. Computer Languages
and Systems
University of Seville
Avda. Reina Mercedes s/n,
41012, Seville, Spain
jegalán@us.es

Pablo Trinidad
Dept. Computer Languages
and Systems
University of Seville
Avda. Reina Mercedes s/n,
41012, Seville, Spain
ptrinidad@us.es

José A. Galindo
Dept. Computer Languages
and Systems
University of Seville
Avda. Reina Mercedes s/n,
41012, Seville, Spain
jagalindo@us.es

Antonio Ruiz-Cortés
Dept. Computer Languages
and Systems
University of Seville
Avda. Reina Mercedes s/n,
41012, Seville, Spain
aruiz@us.es

Automated analysis of feature models (FM) is a field of interest in recent years. Many operations over FMs have been proposed and developed, and many researchers and industrial companies have adopted FMs as a way to express variability. This last makes more necessary having support to detect, explain and fix errors on FMs. The notation of FMs makes very easy to express variability, but makes hard detecting errors and find their cause manually. And these errors may cause the model does not express the variability what we want of it. Therefore, we need support to detect errors and find their causes.

The contribution of this paper is a method to detect errors in FMs, based on the concept of *observation*. We also present implementations of this approach and of an approach to explain errors, in *FaMa Framework* [1] tool.

To detect FM errors, firstly we have to identify the different error types and what it means each of them. Void FM error means that the FM does not represent any product, dead feature error means that a feature of the FM does not appear in any product, false optional error means that an optional feature appears in every product that its parent feature also appears, and wrong cardinality error means that one or more values of a set relationship cardinality are not reachable.

We can check for these errors in a intuitive way. For instance, to detect if a FM has dead features, we can calculate every product and check if each feature appears in, at least, one product. But further, we propose a method based on observations, it means, FM configurations associated with a specific element (feature or cardinality). Each type of error has its type of observation associated too. With an algorithm, we calculate the set of observations of a FM. Then, for each observation, we check if FM has at least one product. If not, we have found an error. For instance, dead feature observation sets its feature as selected. If the FM with a dead feature observation is not valid, it means the feature we are checking is dead.

When we have found the errors, explanations tell us what is the cause of each error. An explanation is a set of rela-

tionships that originates one or more errors. Changing or removing these relationships we can fix a error. However, explanations by themselves do not provide information about how to change the relationship. For instance, if an explanation about a dead feature is a mandatory relationship, we can turn it into a optional relationship, but the explanation does not tell us directly.

We have implemented observations and explanations approaches in FaMa Framework, a tool for the automated analysis of FMs. The observations approach implemented is the previously mentioned, while the explanations approach implemented is the one described by *Trinidad et al.* [3] [4]. With these approaches, we have detected errors in SPLOT FM repository [2], and we have obtained explanations for them also.

1. ACKNOWLEDGEMENTS

This work has been partially supported by the European Commission (FEDER) and Spanish Government under CI-CYT project SETI (TIN2009-07366), and by the Andalusian Government under ISABEL project (TIC-2533) and THEOS project (TIC-5906).

2. REFERENCES

- [1] Fama framework. www.isa.us.es/fama.
- [2] Software product line online tool. <http://www.splot-research.org/>.
- [3] P. Trinidad, D. Benavides, A. Durán, A. R. Cortés, and M. Toro. Automated error analysis for the agilization of feature modeling. *Journal of Systems and Software*, 81(6):883–896, 2008.
- [4] P. Trinidad and A. Ruiz-Cortés. Abductive reasoning and automated analysis of feature models: How are they connected? In *Proceeding of the Third International Workshop on Variability Modelling of Software-intensive Systems (VAMOS)*, 2009.