FAMA: towards the automated analysis of feature models*

Pablo Trinidad, David Benavides, Sergio Segura, Antonio Ruiz Cortés
Departamento de Lenguajes y Sistemas Informáticos
University of Seville

1 Introduction

Feature models (FM) are used to model the set of products within a Software Product Line (SPL) and guide their development. Because of the particular structure of FMs, feature modeling is an error-prone activity. Currently, there exist feature modeling tools but as far as we know, none of them performs an error analysis where FM errors are detected and explained.

On the other hand, FMs are a very important source of information. From them it is possible to perform an analysis of a FM to obtain information such as the number of products that can be built, which percentage of products contains a feature (commonality), the SPL variability factor or which products can be built from a concrete set of features [2]. If FMs are extended with extra-functional information [1] such as feature development cost and time, we could determine which product has the lowest development cost, which product can be built in less time or within the budget. The automated analysis of FMs is a very promising field where there are few proposals and tools that support it.

Our objective is introducing FAMA, a tool designed for feature modeling and the automated analysis of FMs. It is implemented as an Eclipse plug in and it relies on different logic paradigms to implement analysis operations, being a flexible and extensible tools in the kind of operations and the logic paradigms it supports.

*This work has been partially founded by the European Commission (FEDER) and the Spanish Government by means of the CECYT project Web Factories (TIN2006-00472).
2 Feature Modeling
The main difference between FAMA and any other feature modeling tool is its support of error treatment. Two phases are considered to perform this operation: error detection and explanation. During detection, the erroneous features are detected such as dead features (those that appear in no product), false-mandatory features (those that are mandatory despite of being modeled as optional) and void feature models (those where no product can be instanced) [3]. In error explanation, it is intended to obtain a set of relationships that cause the errors to appear so that the user could correct them. FAMA supports the error treatment relying on theory of diagnosis. It makes the proposal an extensible one both in the kinds of error it supports and the logic paradigms it uses to treat them.

3 Automated Analysis of Feature Models
FAMA is able to perform a set of analysis operations that supports decision making during an SPL development process. Among the features and operations it currently supports, we stand out some of them

- Products counting and listing, containing a particular set of features.
- Feature selection and propagation.
- SPL variability calculation.
- Feature commonality calculation.

These operations can help on deciding for example, the priorities in the development of the features within an SPL or which features are part of an SPL architecture.

4 Multiparadigm Architecture
Currently, there exist several proposals on feature model analysis, most of them relying on different logic paradigms such as propositional logic, descriptive logic, constraint programming and ad-hoc algorithms. Conscious of this need, we have developed a multiparadigm architecture that allows the inclusion of new paradigms or algorithms to solve the operations supported by FAMA. Besides, FAMA is also extensible in the kinds of operations it supports, being able to add new analysis operations without affecting its architecture.

FAMA currently takes exploits its multiparadigm architecture to implement the analysis operations, making use of several logic solvers that solve Constraint Satisfaction Problems (CSP), boolean constraint propagation algorithms (SAT) and Binary Decision Diagrams (BDD), selecting the most suitable one for each case.

5 Future Extensions
We intend to equip FAMA with new functionalities for feature modeling and analysis. As regards modeling, we intend to add the support of automated error reparation from explanations. It is also intended to develop an SDK that will allow third-parties to add new analysis operations.

Lastly, we intend to extend FAMA to auto-configure itself, so it chooses the technique that performs better by means of autonomic computing.

References

