# Automated Analysis of Stateful Feature Models

**Pablo Trinidad, Antonio Ruiz-Cortés, and David Benavides**

**Abstract** In CAiSE 2005, we interpreted the extraction of relevant information from extended feature models as an automated reasoning problem based on constraint programming. Such extraction is driven by a catalogue of basic and compound operations. Much has been done since, renaming the problem as the automated analysis of feature models, a widely accepted problem in the *Software Product Line (SPL)* community. In this chapter, we review this seminal contribution and its impact in the community, highlighting the key milestones up to a more complete problem formulation that we coin as the *Automated Analysis of Stateful Feature Models (AASFM)*. Finally, we envision some breakthroughs and challenges in the AASFM.

## 1 Original Contribution

SPL engineering [1] is an emerging paradigm to build families of software products in a given domain considering systematic reuse as a must since very early stages of development. *Feature Models (FMs)* are one of the most widely used models to manage variability and compactly represent the set of products in a SPL [2]. These products are defined as a set of features, each of which describe an increment in product functionality. Besides features, FMs can use attributes to model certain properties of products in so-called *Extended Feature Models (EFMs)*. The automated analysis of FM defined as the automated extraction of information from FMs is an important task to support decision making such as product configuration or model debugging.

In CAiSE 2005 [3] we interpreted the automated analysis of FM and EFMs as an automated reasoning problem based on *Constraint Satisfaction Problems (CSPs)*.

–

Our main contribution was supporting the automated analysis as a catalogue of five basic operations (cardinal, filter, products, valid model and optimum product) and two derived operations (commonality and variability factor), giving a semantics to these operations in terms of a unique semantics domain: constraint programming. The main advantage of CSPs over other automated reasoning techniques was its high declarativity and the wide offer of off-the-shelf solvers to build a reference implementation of the operations by the catalogue.

## 2   Impact and Evolution

An analysis on the references to our work inclines us to think that its main attraction lies in four factors: (i) we pioneered the use of off-the-shelf solvers for the automated analysis. (ii) despite EFMs had already been used for modelling purposes, it was the first approach that enabled their automated analysis. (iii) the proposal of the optimum product operation, which was the first specific analysis operation of EFMs. (iv) we envisioned the composability of the analysis operations, opening the door to the future addition of new analysis operations.

This work has been a reference for many authors [4] who (i) have extended the operations catalogue up to more than 30 operations [5], and (ii) have proposed new techniques which offer a better performance for certain operations. For the authors of the paper, it has been the base of Benavides' and Trinidad's doctoral dissertations [6, 7]. Benavides's work formalises a catalogue of operations for the *Automated Analysis of Feature Models (AAFM)*. The more recent Trinidad's work proposes *Stateful Feature Models (SFMs)* as a kind of fully-configurable FM, and a catalogue of analysis operations on them, which subsumes the AAFM adding a full support for explanatory operations in the so-called *Automated Analysis of Stateful Feature Models (AASFM)*. In Sect. 3 and 4 we provide an overview on the AAFM and the AASFM respectively.

## 3   Automated Analysis of Feature Models

The AAFM can be seen as a black-box process that takes an FM (with maybe some additional information) as an input and outputs a result which depends on an analysis operation (see Fig. 1). For some analysis operations, additional information is needed such as a minimality criterion to find an optimum product, a feature to calculate its commonality, or a *Configuration Model (CM)*, which collects the decisions made by users along a configuration process, to check if there exists at least one product in the FM that fits into their decisions.

The use of declarative approaches is a trend in the AAFM. In [3] we proposed the first interpretation of FMs in terms of CSPs. The seven analysis operations were defined in terms of CSP operations. So for example, searching for all the solutions in a CSP provides the list of products in a FM; or counting all the solutions it is possible
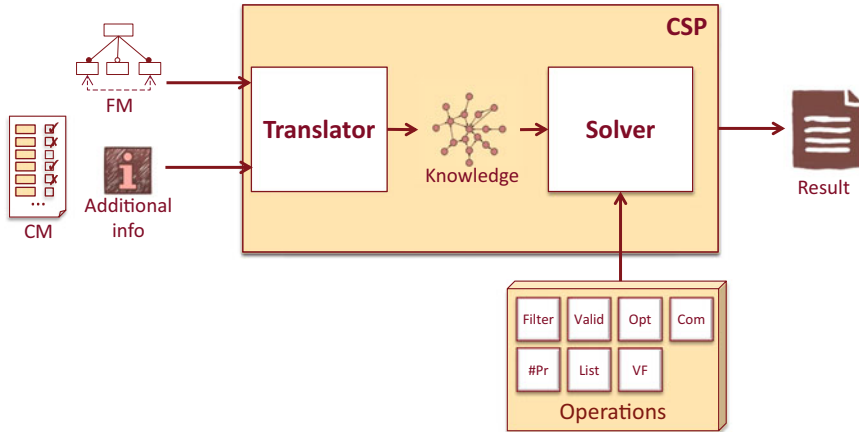
**Fig. 1** Schema of the automated reasoning of feature models proposed in [3]

to obtain the number of products defined within a FM. This was the first proposal to date that were able use attributes for analysis purposes, enabling an optimisation operation that searches for the best product according to a given criterion.

Since our work, other authors have proposed the use of other declarative paradigms such as binary decision diagrams, satisfiability problems or description logics. All these works interpret one or more analysis operations in terms of reasoning problems under these declarative paradigms, generally improving the performance or the expressiveness of previous approaches. In 2010, we counted up to 30 analysis operations [5] using up to 10 different solvers or techniques. Nowadays, the AAFM keeps on being an ongoing discipline in the SPL community with more than 40 proposed operations.

The homogeneity of the AAFM proposals allowed us to develop the FAMA Framework [8], an open-source tool for the AAFM widely used by several research institutions and companies. This tool offers an easy-to-use interface with the most updated catalogue of analysis operations, each of them implemented in several off-the-shelf solvers.

The verification of AAFM tools, i.e. the detection of inconsistencies between AAFM implementation and specification, is an important task in our research. Testing techniques have been developed to verify any AAFM tool in general and FAMA Framework in particular [9].

## 4 Automated Analysis of Stateful Feature Models

Since 2006, one of the open issues in the AAFM was the explanatory analysis [10], i.e. a set of analysis operations to obtain the reasons why another analysis operation has provided a certain result. So for example, if a filter operation detects that there exist no product for a given CM, the explanatory analysis provides for the user

decisions that may be in conflict; or if a FM is detected to be void, i.e. it defines no product due to the use of contradictory relationships, the explanatory analysis can suggest the relationships in conflict.

In [11], we proposed a catalogue of 11 explanatory operations. In [7] we establish the inability of making reference to cardinalities and attributes in CMs. This limitation prevents the representation of decisions such as "I want a system that costs less than a given price" or "I want two different Internet connections", necessary to solve all the explanatory operations. We propose SFMs as a new kind of model that (i) enables user decisions on cardinalities and attributes. (ii) joins FMs and CMs in a single model.

Thanks to SFMs, we provided a semantics for all the explanatory operations but also enabled a new formal specification framework for analysis operations. We interpret all the analysis operations as particular cases of deductive and abductive reasoning, two well-known forms of reasoning in the Artificial Intelligence community. With this approach, we propose a reasoner-independent semantics for all the analysis operations on SFMs in the so-called AASFM.

Besides improving the expressiveness, SFMs aim to improve the analysis capabilities of the AAFM in two aspects: (i) giving a semantics to all the explanatory operations defined in [11], and (ii) as formal specification frameworks are as difficult to build and taking into account that currently there are more than 40 analysis operations, which 11 of them are explanatory operations, we aim to provide a formal specification framework that overcomes these drawbacks.

With this two main goals in mind, the AASFM provides a simple, configurable and expressive semantics to all the explanatory operations, even for those that remained undiscovered in the AAFM. It enables the definition of the following minimal set of core analysis operations whose semantics is given in terms of deductive and abductive operations:

- Validation: this operation determines if a SFM is valid, i.e. it checks if the decisions within the SFM satisfy all the relationships.
- Product listing: this operation obtains a list of all the products in a SFM that satisfy all the user decisions and relationships.
- Propagation: this operation calculates a new SFM where user decisions are automatically suggested from previous user decisions.
- Relationship explanation: this operation explains the possible causes why an SFM is invalid in terms of the relationships that can be causing it [12].
- Configuration explanation: this operation explains the possible causes why an SFM is invalid in terms of the user decisions that must be repaired to reach a valid configuration [13].

Besides these core operations, we provide the composition mechanisms that enables the definition of a set of compound operations inspired in the catalogue of analysis operations proposed for the AAFM, while we open the door for the definition of new operations. Figure 2 shows an overview of the AASFM where we can see the differences with respect to AAFM.
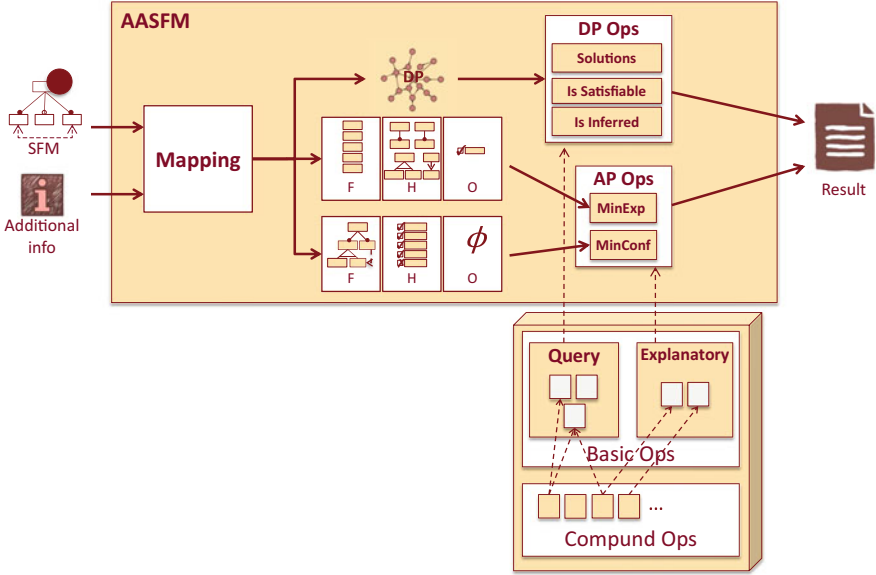
**Fig. 2** General schema of the automated analysis of stateful feature models

As a last contribution, we have explored the use of *Model-driven Engineering (MDE)* approaches to build engines for the AASFM. The large number of solvers and analysis operations in FAMA Framework becomes its maintenance in a time consuming task. With MDE, models are transformed into other models in a chain of transformations until a suitable representation is obtained. The AAFM and the AASFM can be seen as a chain of transformations, what makes the MDE a natural approach to implement them that reduces the maintenance costs.

## 5 Breakthroughs, Challenges and Applications

The AASFM subsumes the AAFM, providing new analysis operations and covering some important gaps in the AAFM, such as the lack of expressiveness of CMs and an incomplete support of explanatory operations. The extensibility of the new catalogue of operations lets the users define their own analysis operations in terms of basic operations, even reducing the efforts to build AASFM engines.

One of the main challenges in the AASFM is the verification of all the operations in the catalogue, adapting existing proposals from the AAFM to the AASFM. Besides verification, obtaining a good performance for AASFM engines in order to incorporate them to the most widespread SPL tools is another challenge in the AASFM.

We conjecture that the results in the AASFM are very close to the automated analysis of other variability models. We understand that it is possible to apply our results to those domains where variability models are used in general such as cloud infrastructures, configuration management and autonomic computing. We envision that the key for this approach to real-world applications resides in not reinventing the wheel but focusing on interpreting real-world problems as a combination of the existing analysis operations in the AASFM catalogue.

# References

1. P. Clements and L. Northrop. *Software Product Lines: Practices and Patterns*. SEI Series in Software Engineering. Addison–Wesley, 2001.
2. K. Kang, S. Cohen, J. Hess, W. Novak, and S. Peterson. Feature–Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University, nov 1990.
3. D. Benavides, A. Ruiz-Cortés, and P. Trinidad. Automated reasoning on feature models. *LNCS, Advanced Information Systems Engineering: 17th International Conference, CAiSE 2005*, 3520:491–503, 2005.
4. Microsoft Academic Research. Automating reasoning of feature models references. http://goo.gl/Yj5iL, 2012.
5. D. Benavides, S. Segura, and A. Ruiz Cortés. Automated analysis of feature models 20 years later: A literature review. *Information Systems*, 35(6):615–636, 9 2010.
6. D. Benavides. *On the Automated Analysis of Software Product Lines Using Feature Models. A framework for developing automated tool support.* PhD thesis, University of Seville, 2007.
7. P. Trinidad. *Automating the Analysis of Stateful Feature Models*. PhD thesis, University of Seville, http://www.lsi.us.es/~trinidad, 2012.
8. P. Trinidad, D. Benavides, A. Ruiz-Cortés, S. Segura, and A.Jimenez. Fama framework. In S. Thiel and K. Pohl, editors, *Software Product Lines, 12th International Conference, SPLC 2008, Limerick, Ireland, September 8–12, 2008, Proceedings. Second Volume (Workshops)*. Lero Int. Science Centre, University of Limerick, Ireland, 2008.
9. Sergio Segura, Robert M. Hierons, David Benavides, and Antonio Ruiz-Cortés. Automated metamorphic testing on the analyses of feature models. *Information and Software Technology*, 53(3):245–258, 2011.
10. D. Batory, D. Benavides, and A. Ruiz-Cortés. Automated analysis of feature models: Challenges ahead. *Communications of the ACM*, 49(12):45–47, December 2006.
11. P. Trinidad and A. Ruiz-Cortés. Abductive reasoning and automated analysis of feature models: How are they connected? In *3rd. International Workshop VAMOS'09*, pages 145–153, Sevilla, Spain, Jan 2009. ICB Research Report N. 29.
12. P. Trinidad, D. Benavides, A. Durán, A. Ruiz-Cortés, and M. Toro. Automated error analysis for the agilization of feature modeling. *Journal of Systems and Software*, 81(6):883–896, 2008.
13. J. White, D. Benavides, D.C. Schmidt, P. Trinidad, B. Dougherty, and A. Ruiz-Cortes. Automated diagnosis of feature model configurations. *Journal of Systems and Software*, 83(7):1094–1107, 2010.