Computing manuscript No.
(will be inserted by the editor)

Automated analysis of feature models: Quo vadis?

José A. Galindo · David Benavides · Pablo Trinidad · Antonio-Manuel Gutiérrez-Fernández · Antonio Ruiz-Cortés.

Received: date / Accepted: date

Abstract Feature models have been used since the 90's to describe software product lines as a way of reusing common parts in a family of software systems. In 2010, a systematic literature review was published summarizing the advances and settling the basis of the area of Automated Analysis of Feature Models (AAFM). From then on, different studies have applied the AAFM in different domains. In this paper, we provide an overview of the evolution of this field since 2010 by performing a systematic mapping study considering 423 primary sources. We found six different variability facets where the AAFM is being applied that define the tendencies: product configuration and derivation; testing and evolution; reverse engineering; multi-model variability–analysis; variability modelling and variability–intensive systems. We also confirmed that there is a lack of industrial evidence in most of the cases. Finally, we present where and when the papers have been published and who are the authors and institutions that are contributing to the field. We observed that the maturity is proven by the increment in the number of journals published along the years as well as the diversity of conferences and workshops where papers are published. We also suggest some synergies with other areas such as cloud or mobile computing among others that can motivate further research in the future.

Keywords Software product lines · Automated analysis · Feature models · Variability-intensive systems.

1 Introduction

Software Product Lines (SPLs) are about developing a set of different software products that share some common functionality [7]. Documented benefits of SPLs are the increment of reuse, quality and reduction of time to market to achieve mass customization of software products in order to satisfy the customer needs as well as decreasing the effort of personalization.

Feature Models (FMs) [15] are tree–like structures that define the set of products that belongs to a SPL and have become the *de facto* standard to represent common and variable characteristics in a SPL. It is easy to find different graphical and textual notations in the literature to represent FMs [4]. The information that can be obtained from FMs is extensive, and the mechanisms for obtaining it are likewise varied. In fact, this area known as Automated Analysis of Feature Models (AAFM) [3] has recently been identified as one of the most important areas in the SPL [12] community.

The AAFM is the computer–aided extraction of information from feature models [4] and can be summarized in three steps. First, FMs are translated to a logical representation. Second, an off–the–shelf solver or specific algorithm is used to perform a given analysis operation (such as counting the number of products or checking the consistency of a FM). Finally, the result is obtained and used in a determined context to perform other tasks such as product configuration or derivation.

A comprehensive list of proposals and operations for the AAFM was presented by Benavides et al. [4] in 2010 that settled the conceptual underpinnings of the discipline. In particular, thirty analysis operations

José A. Galindo · David Benavides · Pablo Trinidad · Antonio-Manuel Gutiérrez-Fernández · Antonio Ruiz-Cortés
Dept. Lenguajes y Sistemas Informáticos, University of Seville,
Avda. Reina Mercedes s/n, 41012, Seville, Spain.
E-mail: {jagalindo,benavides,ptrinidad,amgutierrez,aruiz}@us.es

were presented, and further formalized in [8], different automated mechanisms identified such as SAT, CSP or BDD solvers, a conceptual framework described and directions for future work established. Operations range from determining if a product is valid with respect to a FM to the calculation of the number of different products in a product line.

AAFM has been applied in different activities along the SPL process such as product configuration and derivation, reverse engineering or SPL testing. In this paper, we present a Systematic Mapping Study (SMS) to identify the evolution and trends in the application of the AAFM since 2010. Concretely, we have performed a search on different databases of AAFM–related papers. We selected 423 primary sources (papers) that followed the defined inclusion and exclusion criteria. The primary sources were classified according to different variability facets that were found during the reading and key–wording phase. It is important to remark that before 2010, AAFM was not well defined and it was referenced using an amalgam of names and concepts. Therefore, we consider that in 2010 the concept of AAFM was coined and then used in different domains and scenarios. This paper studies how AAFM has been used since its definition.

We discovered six different variability facets where the AAFM is being applied: *i*) product configuration and derivation. Automated support is used to guide the configuration process and the derivation of specific products. This is the most traditional usage of automated analysis mechanisms; *ii*) testing and evolution. Specific configurations are selected for testing purposes using automated mechanisms. These can support the automated-guided evolution of feature models; *iii*) reverse engineering. Extracting feature models from product descriptions and, in some situations, from logical formulas; *iv*) multi-model variability–analysis. Traditionally, automated analysis operations have been proposed over a single model. However, there are situations where analyses are performed over more than one model; *v*) variability modelling. The basic modelling constructs are not enough in some specific situations and other information such as attributes are used for modelling different situations and analysis are performed with these new modelling elements, and; *vi*) variability–intensive systems. AAFM is used in other application domains not directly related with SPLs.

We also observed that there are only a few industrial and real evidences of the application of AAFM techniques in most of the cases. We detect in detail where and when the papers have been published and who are the authors and institutions that are contributing to the field. We saw that the maturity is proven by the increment in the number of journals published along the years as well as the diversity of conferences and workshops where papers are presented. Finally, we devise some research opportunities and applications in the future as well as synergies with other research areas.

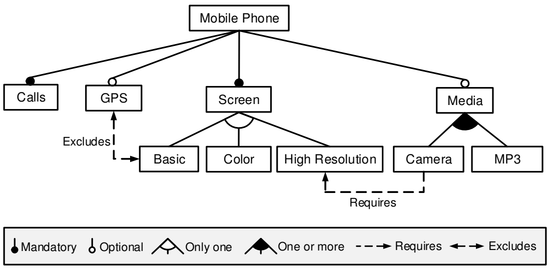
The remainder of this paper is organized as follows: Section 2 presents the background required to understand the scope of this paper; Section 3 discusses the previous related works that are related to this research area; Section 4 covers the methodology used in this mapping study; Section 5 presents results from analyzing the primary studies; and Section 6 presents concluding remarks.

2 Background

2.1 Feature models

Feature Models (FMs) are one of the most used artefacts to describe the set of products in a SPL in terms of *features* and relationship among them. In FMs, features are hierarchically arranged in a tree–like structure. In addition, cross–tree constraints can be used to connect features. Figure 1 shows an example of a FM describing a mobile-phone SPL, using the most common notation (Figure 1a) and its represented products (Figure 1b). There are different proposals for FM notations (see [26] for a detailed survey) but most of the proposals have the following common elements:

- *Mandatory*: a *child feature* has a *mandatory* relationship with its *parent feature* when it appears in a product whenever its parent does. In Figure 1, Screen has a mandatory relationship with Mobile Phone, i.e. any mobile phone must have a screen.
- *Optional*: a child feature has an *optional* relationship with its parent feature when the child can appear or not in a product whenever its parent does. In the example in Figure 1, Media has an optional relationship with Mobile Phone, i.e. A mobile phone can have support for media features or not depending on the configuration chosen.



(a) A sample feature model of a SPL for mobile phones using a FODA–like notation. (Source: [4])

#	Features
1	Mobile Phone, Calls, Screen, Basic
2	Mobile Phone, Calls, Screen, Color
3	Mobile Phone, Calls, Screen, High Resolution
4	Mobile Phone, Calls, Screen, High Resolution, Media, Camera
5	Mobile Phone, Calls, Screen, Basic, Media, MP3
6	Mobile Phone, Calls, Screen, Color, Media, MP3
7	Mobile Phone, Calls, Screen, High Resolution, Media, MP3
8	Mobile Phone, Calls, Screen, High Resolution, Media, Camera, MP3
9	Mobile Phone, Calls, Screen, Color, GPS
10	Mobile Phone, Calls, Screen, High Resolution, GPS
11	Mobile Phone, Calls, Screen, Color, GPS, Media, MP3
12	Mobile Phone, Calls, Screen, High Resolution, GPS, Media, MP3
13	Mobile Phone, Calls, Screen, High Resolution, GPS, Media, Camera
14	Mobile Phone, Calls, Screen, High Resolution, GPS, Media, Camera, MP3

(b) The set of products depicted by the feature model

Fig. 1: A sample feature model among it set of represented products

- *Or–relationship* (also known as *OneOrMore*): a set of child features has an *or–relationship* with its parent when one or more child features can be selected when the parent is. Figure 1 contains an *or–relationship* between Camera and MP3 with Media. Whenever Media is present in a product, Camera, MP3, or both have to be present.
- *Alternative* (also known as *OnlyOne*): a set of child features have an *alternative* relationship with their parent when one and only one of them can be selected in a given product whenever their parent is selected. Figure 1 shows an alternative relationships among Basic, Colour and High Resolution, so a given mobile phone can only have a specific type of Screen in a product.
- *Requires*, *Excludes*: Cross–tree relationships like A requires B means that whenever feature A appears in a product, feature B must also appear. Also, a relationship like A excludes B means that both features cannot appear in the same product at the same time. Figure 1 shows two examples of these kinds of relationship: the Camera requires a High Resolution screen and the GPS excludes the Basic screen.

2.2 Automated analysis of feature models

The AAFM deals with extracting information from FMs by using computer–aided mechanisms. SPL engineers use the information to improve their business strategies as well as to take technical decisions. The process to extract such information is shown in Figure 2. It starts by translating the features and relationships encoded in the FM and any other additional information (e.g. market share [10]) to a knowledge base described in a logic paradigm. Later, queries to the knowledge base can be performed using existing solvers or tools thus, obtaining the analysis results. In [4], different analysis operations on FMs were reported. According to that study, we present some of them:

- *Finding out if a product is valid.* This operation checks if a product (i.e. set of features) belongs to the set of products represented by a FM or not. It is helpful for SPL engineers and managers to determine whether a given product is available in a SPL.
- *Obtaining all products.* This operation lists the products represented by a FM. It allows practitioners to identify the final products that they can manage in their SPL. For example, the model shown in Figure 1a represents the products in Figure 1b.
- *Calculating the number of products.* This operation counts the number of products of a FM. This provides information about the size and complexity of the SPL represented by a FM. It is commonly used to perform more complex operations such as calculating the amount of reuse metrics of a SPL. For example, in Figure 1a there are fourteen products.
- *Detecting errors.* The large number of different features used in a FM increases its complexity as well as the probability of introducing errors. There are several types of errors that can be detected by using the AAFM. For example, it is important to determine if a FM is void, i.e. whether it represents no product at all because of contradicting relationships. Another common error is the detection of dead features, i.e. features that cannot appear in any of the products derived from the model. Dead features are clearly undesired since they are the result of a wrong domain modelling.
- *Explaining errors.* As shown before, there are circumstances where FMs contain errors. In such situations, it is important to assist on resolving them. An explanation operation takes a FM as input and a set of previously identified errors, trying to provide insights to correct them.

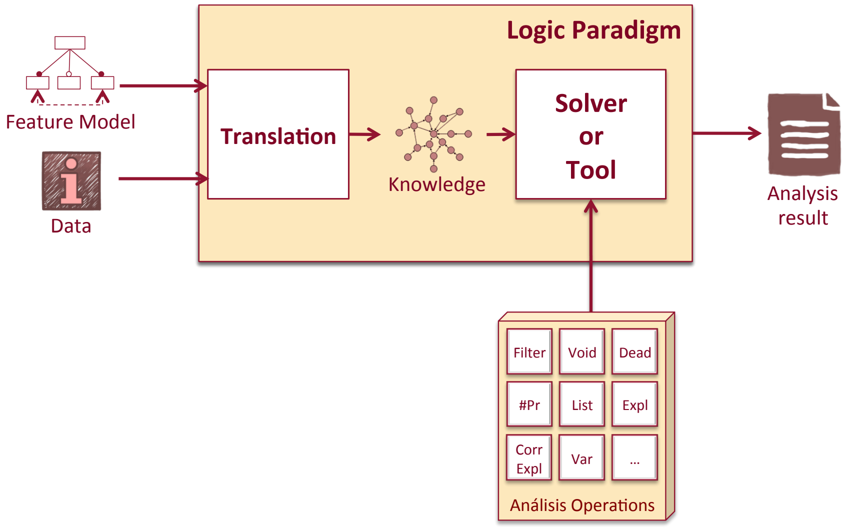


Fig. 2: AAFM framework.

These operations are performed automatically using different approaches. Most of them translate FMs into specific logic paradigms such as propositional logic, constraint programming or description logic. Others propose *ad–hoc* algorithms and solutions to perform these analyses [4]. Finally, these analysis capabilities can also be found in several commercial and open source tools such as *pure::variants*[1], *SPLOT* [21], FaMa [5], FeatureIDE [245] or FAMILIAR [1]

2.3 Literature review methods

To crawl the existing knowledge in the literature there are different kinds of reviews [11], but we focus on two of them. Figure 3 depicts the main difference among them and how they are related.

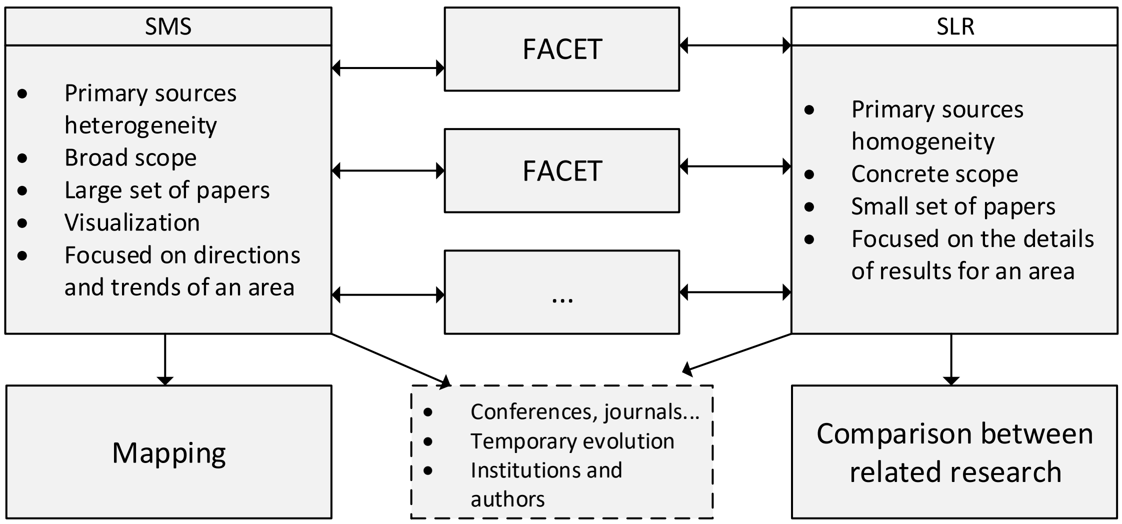


Fig. 3: Systematic mapping studies and literature reviews

[1] `http://www.pure-systems.com/`

– *Systematic Literature Reviews (SLRs)* take primary sources, i.e. papers under study, that are homogeneous in terms of comparability and compare them to get conclusions [17]. All papers of the review are comparable among different aspects. SLRs provide a synthesis of the knowledge existing in the core content of primary sources of a specific field and, to do so, researchers have to read, understand and classify the whole content of the studies. SLRs provide as output as many details and comparisons as possible among the content of primary sources. In other words, they are focused on *"how"* the things have been done in a concrete field. This is, SLRs target a reduced set of papers and extract all the insights possible from them.
– *Systematic Mapping Studies (SMSs)* take primary sources that are heterogeneous in terms of comparability but are related to a broader area and provide a mapping and categorization of the different facets detected in the studies [24]. Researchers read the title, abstract, and optionally other parts of the paper stepwise. The idea is to get the whole picture of a broad research area. SMSs provide visual outputs that allow an easy identification of research gaps. For example, most of SMSs provide bubble plots or heatmaps that show how well covered is an intersection between the categories analysed. Note that SMSs do not analyse specific research but allow the characterization and classification of more heterogeneous papers. In other words, the main focus of an SMS is to detect *"what have been done"*. Petersen *et al.* [24] detailed the process of building systematic mappings and compared them with SLRs. SMSs take as input as many papers as possible reducing the bias of only reading portions of the papers in the case of a mistake in the classification.

There are potential relations between SMSs and SLRs as shown in Figure 3. The results of an SMS can be the basis for a more in depth SLR of one or several of the research facets that are part of the mapping. For instance, an SLR of testing in mobile phone applications can then serve as the basis for performing an SLR of techniques for automated unit tests in that context, which is a more concrete field. SLRs and SMSs usually provide common outputs such as the fora where authors have published, temporary evolution of primary sources as well as authors and institutions contributing to the area of study.

3 Related work

As explained in Section 4 there are different kinds of review methods. In this Section we go through other systematic reviews identifying the context, year and period. In the area of SPLs, several SMSs or SLRs can be found [4, 9, 18, 19, 22, 23] as shown in Table 1, most of them being published in recent years. The period of time used for the reviews varies depending on the topic, ranging from 1990 (where SPLs started to be popular) to the date where the study was performed [2]. Also, the number of analysed contributions of our work reviewed more studies. As explained in Section 4.2, we used 423 papers for extracting some general data and 242 for more concrete insights.

Review work	Kind	Year	Context	Period	# papers studied
Benavides *et al.* [4]	SLR	2010	Automated analysis of feature models	1990–2010	53
Da Mota *et al.* [23]	SMS	2011	SPL testing	1993–2009	120
Engström and Runeson [9]	SMS	2011	SPL testing	1990–2008	64
Laguna and Crespo [18]	SMS	2013	SPL evolution	1990–2011	74
López-Herrejón et al. [19]	SMS	2015	Search-based SPLs	2011–2014	77
Montalvillo and Díaz [22]	SMS	2016	Requirement-driven evolution in SPLs	1990-2015	107
This paper	SMS	2018	Automated analysis of feature models	2010–2017	423/242

Table 1: Comparison of related reviews

Heradio *et al.* [13] presented a bibliometric analysis of 20 years of SPLs, from 1995 to 2014. One of the conclusions was that "*feature modelling has been the most important topic for the last fifteen years, having the best evolution behaviour in terms of number of published papers and received citations*"[13].

[2] López-Herrejón *et al.* [19] reduce the period of time due to the topic handled.

Also, in the 2009-2014 period, the AAFM was the most influential topic according to the number of papers produced and the centrality and density of papers. Note that [13] is not an SMS nor an SLR and only performs data analysis based on automated mechanisms.

3.1 Need for a systematic mapping study

Benavides *et al.* [4] analysed in detail the relevant literature that builds up the body of knowledge of the AAFM. That work analysed and categorized concrete methods for performing automated analysis of feature models. Also, authors identified different analysis operations and several automated mechanisms for performing the operations (see Section 2.2). We conjecture that in [4] the grounds of the field of AAFM were settled and from then, most of the papers are using analysis operations to solve different problems in SPL engineering instead of defining new operations or proposing new analysis mechanisms or algorithms.

In this paper, we want to detect the main trends where the AAFM is being used, this is, we want to include in this study any paper that uses the AAFM in any area of the software product line engineering process (e.g. testing, requirements, derivation,...). In this context, we performed an SMS which is the most suitable review method to cover a wider set of publications and understand the current and future state of this area. Section 4.1.1 shows the set of questions that will guide our study and will help us to achieve our goal.

The authors of this work belong to one of the most active research groups in the area of feature modelling, and automated analysis according to [13], being these topics on the ones that attracted more attention in recent years. As the field evolves, there is a need to evaluate the trends in the area and assess their maturity. With this work, we aim to serve as evidence for research opportunities both in the kind of research (i.e. more empirical vs more theoretical) and in the topics to be covered.

4 SMS process definition

This SMS follows a process inspired by the one proposed by Petersen *et al.* [25] summarized in Figure 4. The main steps to perform a systematic mapping study are:

1. Planning the review, which includes the definition of the search protocol, the survey of the literature and the definition of the research questions;
2. Study identification; where the databases are crawled and the primary studies selected;
3. Data extraction and classification, where the mapping is developed and conclusions are obtained.

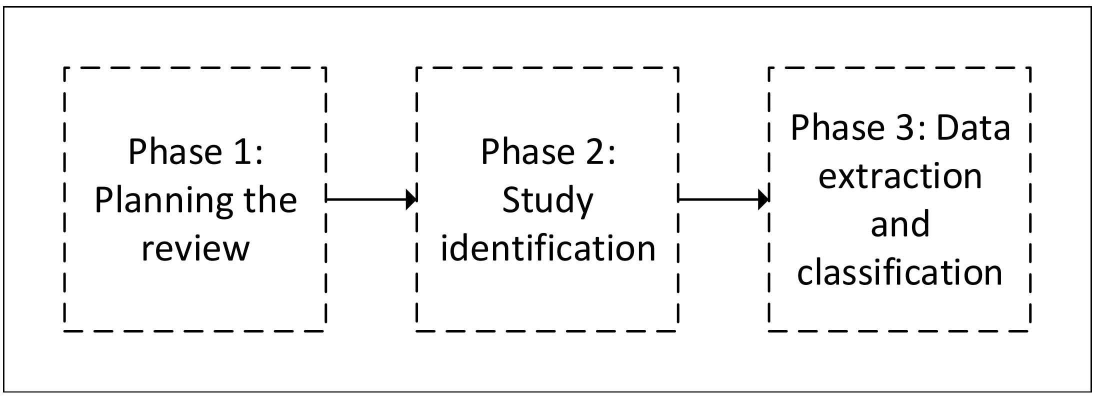


Fig. 4: Systematic mapping study process [22, 25]

Next, we detail the process that guides this SMS.

4.1 Phase 1: Planning the Review

This phase comprises the execution of three process steps as shown in Figure 5: i) protocol definition, where we decide how to do the review and how to minimize the threats to validity; ii) literature survey, that consists of getting the base of the body of knowledge of the area, and; iii) the definition of research questions, where the questions to be answered are stated. Next, we detail how we performed each of these steps.

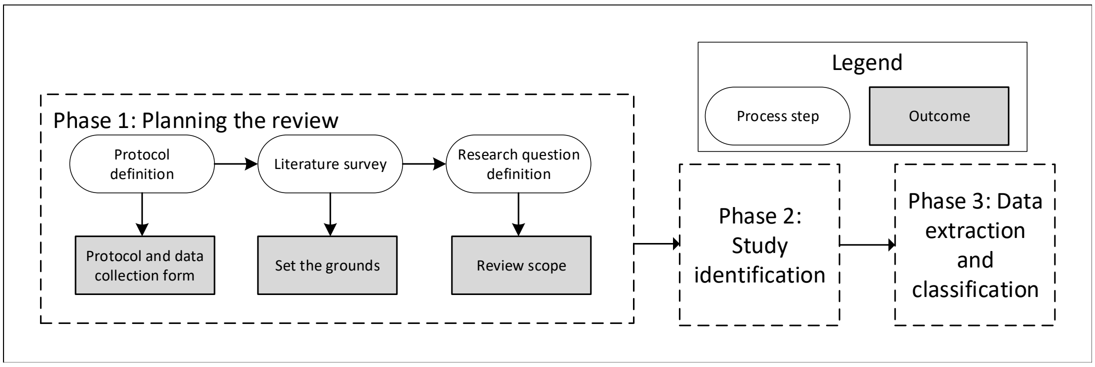


Fig. 5: Planning the review phase

Protocol definition. The protocol that we defined follows the guidelines of Petersen *et al.* [25] to perform systematic mapping studies. Then, we compared the research questions with Jia *et al.* [14] 5W+1H model to check their accuracy (see Section 4.1.1). Also, we modified the protocol to obtain not only the results from databases but also considering the papers quoting Benavides *et al.* [4] (see Section 4.2).

Literature survey. In order to survey the current state of the art in the AAFM, we read the papers published in the main conferences by means of authors expertise and journals as well as taking the following papers as the main input defining the body of knowledge [4, 8]. Also, we relied on the expertise of some of the authors of this paper that have been working in the AAFM and collaborating with the community for the last ten years or so.

4.1.1 Definition of research questions

Defining a set of research questions that guide an SMS is a difficult task that could lead to a biased study, which is aggravated if the authors have experience in the area. To guide and reduce the arbitrariness of the questions definition, we were inspired by the 5W+1H pattern proposed by Jia *et al.* [14] using it to verify the completeness of our proposed questions. This pattern relies on the 5W+1H model, a pillar of journalism to report stories originally proposed by Kipling [16]. 5W+1H is an abbreviation of *Who*, *Why*, *What*, *Where*, *When* and *How*, the six questions to be answered in order to know the most important aspects about a story. Thus, we defined the following research questions for our study:

RQ1 *Where are the papers published?* Which fora are being targeted by practitioners in the last six years? We think that this question can help researchers to know which communities are interested in which topics. Therefore, to target the most suitable forum for a new contribution.

RQ2 *Who are the authors and institutions that make research into AAFM?* Which authors have been more active regarding the AAFM area and which institutions are currently hosting them? This question aims at fostering researchers collaboration and to determine which institutions hold the knowledge of a topic.

RQ3 *What are the areas for which AAFM has been applied?* What topics have attracted more researchers attention in the last years? This research question aims at discovering "Why" the researchers have applied AAFM techniques in the sense of the applications where AAFM has been attracting researchers attention.

RQ4 *Which kind of publications are used to address the challenges?* This question gets insights about the maturity of the area by relying on Wieringa *et al.* [27] taxonomy research which includes validation research, evaluation research, solution proposal, philosophical papers, opinion papers and experience papers.

RQ5 *When have the papers been published?* This question tries to determine the temporary evolution of the publication types and fora. This question aims at helping researchers to determine the current state of a topic, thus, to see if there are chances of further collaboration in that area or not.

RQ6 *How are the interrelationships among the papers?* This question tries to find research gaps whether to invest more research efforts in the future by analysing the already covered areas.

Note that most of the questions could be rewritten differently and that the pattern [14] is used for the sake of coverage w.r.t. the research questions.

4.2 Phase 2: Study identification

In this section we explain how we identified the studies to include in the mapping study as shown in Figure 6. Please note that in Figure 7 we provide details about the evolution of the selection of papers used in this review: i) conducting search, where we perform a raw search in different data sources for primary studies; ii) filtering studies, that consists in applying inclusion and exclusion criteria. iii) deep search, where new studies are added manually to improve the quality of the primary sources. iv) evaluate search, where we get the final set of primary sources after an evaluation. Next, we detail how we performed each of these steps.

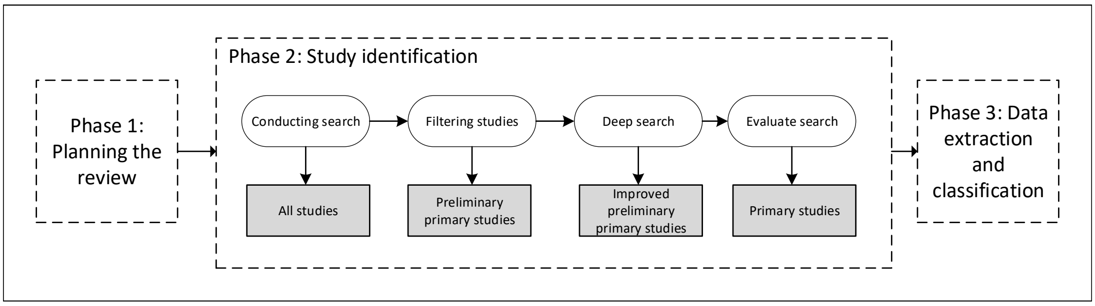


Fig. 6: Study identification phase

4.2.1 Conduct search for primary sources

The process followed to conduct the search for primary studies is shown in Figure 7 (step 1). We decided to crawl Google Scholar [3] and Scopus [4] databases because both of them provide easy mechanisms to export the resulting data ordered by number of citations. In each database, we performed two different queries [5]: one with the following search string *"feature model"* AND (*"reasoning"* OR *"analysis"* OR *"automated"* OR *"analyses"*) and the other with the papers citing [4]. We specified that the publication date should be between 2010 and 2017. We obtained 15,300 (using the search string) and 945 (citing [4]) for Google scholar queries and 1167 (using the search string) and 572 (citing [4]) for Scopus queries. Then, we selected only the first 200 most cited papers from each data source and removed duplicates getting a total of 445 initial studies. Also note that in this paper, we want to determine in which scenarios the AAFM has has been used. It is important to remark that [4] was published in 2010 and that is the reason why we start the search for papers from that date.

[3] `http://scholar.google.com`

[4] `http://www.scopus.com`

[5] Note that, as well as the standard process proposed defined by [25] to query bibliographic databases, we added a second group of papers citing the paper that settle the body of knowledge of AAFM [4] as justified in Section 3.1.

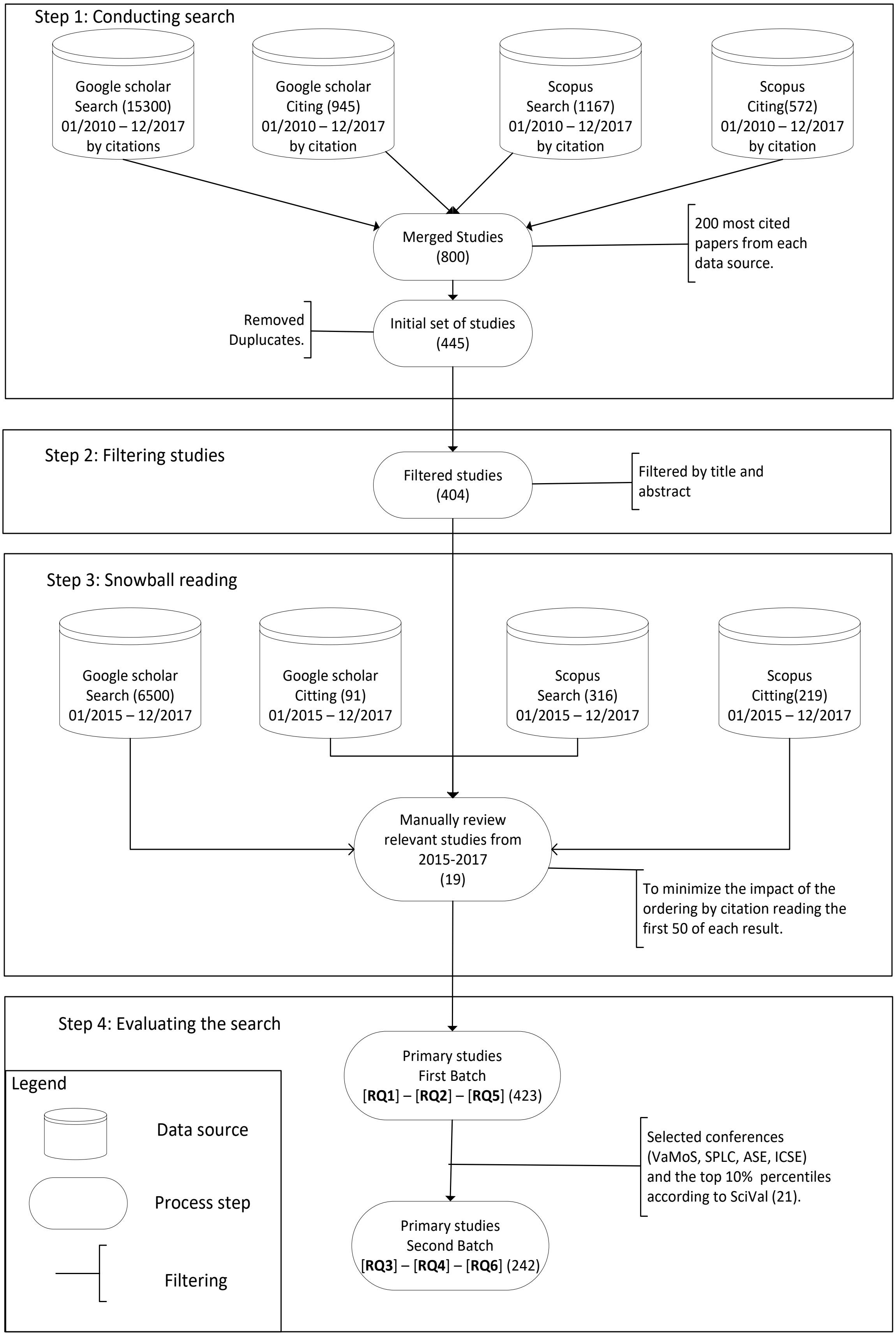


Fig. 7: Search citing publications process

4.2.2 Filtering studies

From the 445 papers of previous step, we discarded non-peer reviewed material such as technical reports, secondary studies on AAFM and thesis documents and performed a detailed screening of the papers removing papers not related to AAFM. This resulted in a set of 402 primary studies.

4.2.3 Deep search

To prevent missing some relevant works, we conducted a special search for the papers published from 2015 to 2017 that had no time to gain citations and tried to detect some relevant papers missing from the initial search as proposed by [22]. This search was relying on snowball reading and the experience of the authors. We removed duplicates and added 19 additional papers, obtaining a total number of 423 papers for the next step. Also, we checked if there was a missing highly cited paper considering it for inclusion. This is done to reduce the chances that a non included paper biases the observations.

4.2.4 Evaluate search

The resulting 423 papers, which are taken as input for this step, were used for **RQ1**, **RQ2** and **RQ5**, this is, to determine where and when the papers were published and who are the authors and institutions that publish results. We were able to use this large amount of papers since we used automated mechanisms to gather this data from our local database.

For the remaining research questions, we only considered the papers published in journals, and conferences with a high acceptance rate of papers related to variability-management and AAFM, according to our own classification of the previous step, i.e. the answer of **RQ1** (see Table 4). This resulted in a total of 242 papers for questions **RQ3**, **RQ4** and **RQ6**. Also, for this second set of papers, we considered all contributions in the top 10% percentiles according to SciVal[6] to avoid missing relevant papers.In the study, we used two different set of papers to answer the RQs as illustrated in Figure 7. For RQ1, RQ2 and RQ5, we used 423 papers and for RQ3, RQ4 and RQ6 we used 242. Note that 242 is a subset of the 423 papers. The reason to use two different sets is due to the fact that, on the one side, for RQ1, RQ2 and RQ5 we used semi automated mechanisms that made it feasible to handle that big amount of papers. While, on the other side, for RQ3, RQ4 and RQ6 we had to analyse the content of the papers what would have been unfeasible with the bigger set.

4.3 Phase 3: Data extraction and classification

This phase comprises the execution of two process steps as shown in Figure 8: i) topics keywording, where we detect the topics to answer **RQ3**, **RQ4** and **RQ6**; ii) data extraction and mapping, where the mapping itself is done. Next, we detail how we performed each of these steps.

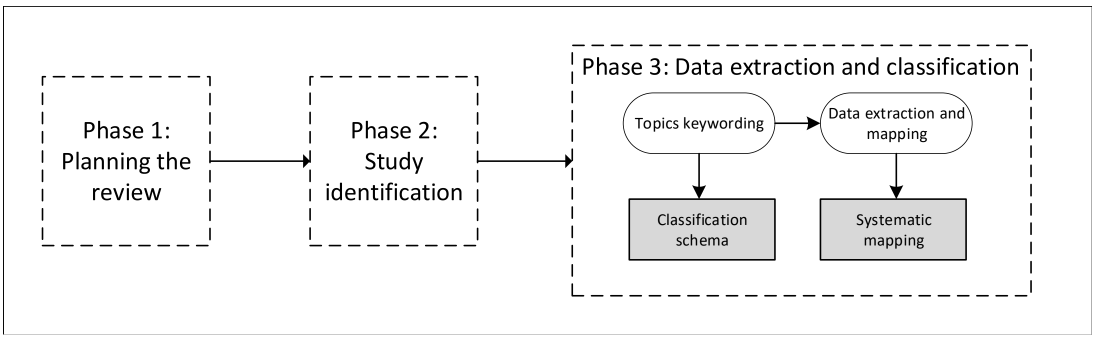


Fig. 8: Systematic mapping study process [25]

[6] www.scival.com

4.3.1 Topics keywording

Publications are classified in two dimensions: i) the variability context facet; and ii) the research type. We defined these dimensions by following up the process described by Petersen et al.[24]. Petersen *et al.*, propose to use a keywording method to define the research focus to group the papers. This process is divided in two steps. First, researchers read the abstracts in the collection to review and identify the keywords and concepts that reflects the paper contribution. If the abstracts are not enough, then researchers take a look at the introduction and optionally more parts of the papers until the concepts can be identified. Finally, researchers define the set of categories to do the mapping by identifying the main paper contributions. Also, if after this reading is unclear where a paper belongs, we take a look at the conclusions and then, the category where the paper is more focused on. As a result of the keywording process, the following variability context facets were defined. Note that we understand the term *"variability context"* as the different AAFM application domains. It is worth noting that a paper can belong to more than one category:

- **Product configuration and derivation**, papers coping with the derivation or the configuration of products. For example, [237] enables the configuration of FMs but taking into account the preferences of multiple stakeholders.
- **Testing and evolution**, these papers focus on the use of automated analysis techniques to optimize the testing of the products derived from a FM such as those presented in [41] to detect feature interactions which are error sources.
- **Reverse engineering**, these papers describe different techniques to build up FMs from a variety of product descriptions such as product description matrices [36] or lists of products.
- **Multi-model variability analysis**, these papers focus on the analysis when variability is not described in a single model but in several models (e.g. merging and slicing operations between models). For example, Dhungana *et al.* [90] presented a solution to enable the configuration of diverse and inter-operable variability models.
- **Variability modelling**, papers focussing on the description and modelling of the variability to perform further analysis and extraction of relevant information. For example, Berger *et al.* [62] presented a study on the variability models and languages in the systems software domain.
- **Variability intensive systems analysis**, these papers focus on applying automated analysis techniques into variability intensive systems (usually not categorized as SPLs) that have to cope with variability requirements such as the Linux kernel [232].

Different kinds of publications were also taken into account for this mapping study. Concretely, we followed up the proposal of Wieringa *et al.* [27] and encouraged in [24], which propose the following types:

- **Opinion papers**, papers showing the author opinions over a concrete technique but not relying on methodologies or related work.
- **Philosophical papers**, these papers help structuring the area such as taxonomies, literature reviews and mapping studies.
- **Solution proposal**, in these papers authors propose solutions to problems but relying in existing techniques – even improving them. There is no need for evaluation or validation.
- **Evaluation research**, implemented techniques with evaluation and conclusions. Here the technique has been evaluated in front of examples or small datasets but there is no validation by final users yet.
- **Validation research**, this research type focuses on techniques already evaluated with end users.
- **Experience papers**, papers explaining industrial or personal experiences in the field.

4.3.2 Data extraction and mapping

Once the facets and topics are gathered in the previous steps, the mapping is ready to be performed. When classifying the data obtained from Phase 3, we proceeded with the following steps to increase the confidence about its correctness.

1. We created a database with all the papers to be classified. In our case, we decided to use a bibtex database and exploited it by using JabRef [7]. The database contained the following fields for each

[7] `www.jabref.com`

paper: i) title ; ii) abstract in plain text ; iii) link to reach the full paper; iv) proposed variability facet; and v) type of research facet.
2. We formed two groups of researchers to carry the classification task independently.
3. Each group performed the keywording process in order to identify the minimal set of categories that provides a good separation of concerns within the papers. Then, we had a meeting and decided the final classification schema to use.
4. Then, the two teams performed the classification individually on the two proposed dimensions (variability context and research type).
5. A single member identify the papers that were classified differently.
6. We held a meeting in which all the papers classification identified in the previous step were discussed until a consensus was reached.

4.4 Threats to validity

Even though the research we present in this paper aims to be as systematic as possible, there are some assumptions that we made that may affect its validity.

External validity. As mentioned before, we included works directly citing Benavides *et al.* [4] previous work and the result of different queries in scientific databases. This may have biased the process by increasing the papers that cited a single paper (e.g. authors not knowing that study). However, we minimised the impact of this threat by including as many papers as possible.

The major threats to the external validity are:

- *Population validity*, we included a large set of papers to reduce the possibility of missing relevant works. However, using the number of citations as threshold might prevent the inclusion of some interesting works. However, we consider that with a set with 423 papers for the **RQ1**, **RQ2** and **RQ5** and 242 papers for **RQ3**, **RQ4** and **RQ6** we provide a good coverage of the existing research and reduced the probability of missing interesting works and the impact of a bad classification in the keywording process. Moreover, we cross-checked other SMS verifying that this amount is above the average number of papers (See Section 3).
- *Ecological validity*, it is focused on possible errors in the experiment materials and tools used. We relied on automated mechanism when possible instead of relying on manual methods to prevent this error. Also, there is a threat inherent to the use of the number of papers published in a venue to map dimensions and discard papers in SMSs. To prevent this thread, we also included those papers ranked in the top 10% percentile according to SciVal.

Internal validity is a measure which ensures that a researcher's experiment design closely follows the principle of cause and effect. In this mapping study we have tried to be as methodologically exhaustive as possible. However, a manual classification process such as keywording might introduce some errors. Again, we considered a large set of papers to minimise the impact of an erroneous classification.

5 Results

In this section we revisit the different research questions defined in Section 4.1.1.

5.1 RQ1: Where are the papers published?

This question is used twofold. First, to identify which conferences and journals are accepting AAFM results. Second, to reduce the number of papers of the first batch as detailed in Section 4.3 by means of a clear criteria.

Table 2, shows the top-ten journals depending on the number of papers published on them for this mapping study. A total of fifty-eight different journals were detected in this study (see Appendix B). We noted that the top-ten journals were indexed in journal quality rankings such as JCR [8] or SCImago [9].

[8] `https://jcr.incites.thomsonreuters.com`
[9] `https://www.scimagojr.com`

	#papers	Journal	Acronym	SCImago	JCR
1	13	Journal of Systems and Software	JSS	✓	✓
2	11	International Journal on Software and Systems Modeling,Software and Systems Modeling	SOSYM	✓	✓
3	9	Software Quality Journal	SQJ	✓	✓
4	7	IEEE Transactions on Software Engineering	TSE	✓	✓
5	6	Information and Software Technology	IST	✓	✓
6	6	Science of Computer Programming	SCP	✓	✓
7	5	International Journal on Software Tools for Technology Transfer	STTT	✓	
8	4	Expert Systems with Applications	ESA	✓	✓
9	3	Computer	Computer	✓	✓
10	3	Empirical Software Engineering	ESE	✓	✓

Table 2: Top-ten journals

The Software Quality Journal appeared in third place, this might be indicative of the interest in testing variability-intensive systems and SPL

Rank	Journal	Variability intensive systems analysis	Multi-model variability analysis	Product configuration and derivation	Testing and evolution	Variability modelling	Reverse engineering	Experience Report	Philosophical Paper	Opinion Paper	Solution proposal	Evaluation Research	Validation Research
1	JSS	1	1	4	8	1	1	1	1	1	1	9	0
2	SOSYM	2	1	4	4	3	2	0	2	0	1	7	1
3	SQJ	0	2	2	5	2	0	0	1	0	1	7	0
4	TSE	0	0	3	2	1	3	0	0	0	0	5	2
5	IST	1	1	3	2	3	0	0	0	0	1	4	1
6	SCP	1	1	4	0	3	0	0	0	0	2	4	0
7	STTT	0	1	3	2	3	0	0	1	0	1	2	1
8	ESA	0	0	1	2	2	0	0	0	0	0	4	0
9	Computer	1	1	2	1	0	0	1	0	2	0	0	0
10	ESE	0	0	0	2	1	2	0	0	0	0	2	1

Table 3: Top-ten journals with variability facets and research type

Table 3 presents the journal names among the number of papers classified in each category. For the category of each paper we used the second set of papers only containing all journals and conferences with more than ten contributions. This table aims at helping deciding where to submit a new contribution. We observe that testing and evolution is the most common topic on the first members in the classification. Also, the same happens with the type of research, where the most common type for the first members of the rank is evaluation research.

	#papers	Conference	Acronym
1	42	Software Product Line Conference	SPLC
2	33	International Workshop on Variability Modelling of Software-intensive Systems	VAMOS
3	13	International Conference on Software Engineering	ICSE
4	11	International Conference on Automated Software Engineering	ASE
5	9	International Conference on Model Driven Engineering Languages and Systems	MODELS
6	9	International Conference on Software Reuse	ICSR
7	7	International Conference on Generative Programming	GPCE
8	6	International Conference in Software Testing	ICST
9	6	International Conference on Advanced Information Systems Engineering	CAISE
10	5	International Conference on Fundamental Approaches to Software Engineering	FASE

Table 4: Top-ten conferences

Table 4, shows the top-ten conferences based on the number of papers related to automated analysis in their proceedings. We see that the MODELS conference is accepting several AAFM related contributions. This might be indicative of the increasing importance of variability modelling in modelling specific conferences. Also, it is remarkable that even though there are conferences traditionally accepting AAFM papers, there are other conferences such as ICSE, MODELS, ECSA among others that are also accepting contributions.

To reduce the number of papers to review in **RQ3**, **RQ4** and **RQ6**, we only considered papers published in journals and in the conferences having more than ten contributions (i.e. SPLC, VAMOS, ASE, ICSE). Note that we also considered for this second batch the papers within the top 10% percentile according to SciVal. The full list of journals and conferences is provided in Appendix B.

Acronym	Variability intensive systems analysis	Multi-model variability analysis	Product configuration and derivation	Testing and evolution	Variability modelling	Reverse engineering	Experience Report	Philosophical Paper	Opinion Paper	Solution proposal	Evaluation Research	Validation Research
SPLC	3	5	23	11	16	2	3	0	0	11	24	4
VAMOS	3	8	12	11	14	4	1	1	3	10	12	6
ICSE	1	0	9	2	1	3	0	0	0	2	11	0
ASE	1	1	4	3	2	1	0	1	0	2	7	1

Table 5: Top-ten conferences with variability facets and research type

Table 5 presents the conferences among the number of observations done in for each topic. This table was built using the second batch of papers only containing papers from journals and conferences with more than ten contributions as well as the papers present in the top 10% percentile according to SciVal. This table shows that more traditional subareas such as product configuration and derivation are more present in more generalist conferences such as ICSE and ASE while the niche conferences of the area such as VAMOS or SPLC hold a more varied set of contributions. Also it worth mentioning that regarding the research type it was similarly distributed among all conferences.

5.2 RQ2: Who are the authors and institutions that research on AAFM?

In this question, we want to discover the most active researchers and institutions using AAFM. For that, we took the first batch of papers without excluding any conferences. Then, we counted how many papers were published by each author. Table 6, presents those first authors with more than three AAFM related publications since 2010. Also, we added the columns were the author contributions were categorized. Note that while for the ranking of papers we relied on the first batch of papers, we used the second batch for the categories. This table aims at fostering future collaboration between researchers working on similar areas. [10].

After identifying the most prolific first authors in the area, we wanted to highlight which institutions are currently hosting them. This is, we care about the current institution of an author, not the one at the time of writing the paper. This is done to identify which institutions currently have the know-how in AAFM. To perform this analysis, we relied on Google Scholar author search. Again, we used the first batch of papers without excluding any conferences. We searched by each first author in the paper and retrieved its current verified institutional email in Google Scholar and then, we manually looked for the institutions associated with the email domain. Table 7 presents the institutions publishing more than four AAFM related papers. It is remarkable that the AAFM is mostly attracting European institutions.

[10] To get the full list of first authors you can take a look to the URL provided in the additional material section.

#Papers	Author	Variability intensive systems analysis	Multi-model variability analysis	Product configuration and derivation	Testing and evolution	Variability modelling	Reverse engineering	Experience Report	Philosophical Paper	Opinion Paper	Solution proposal	Evaluation Research	Validation Research
14	Acher, Mathieu	0	4	1	1	0	3	0	0	0	2	3	2
7	Lopez-Herrejon, Roberto E	0	0	0	4	0	2	0	0	1	2	2	0
7	ter Beek, Maurice H	3	0	3	0	2	1	0	0	0	0	2	1
6	Segura, Sergio	0	0	0	5	0	0	0	0	0	0	5	0
6	Wang, Shuai	0	0	2	4	1	0	0	0	0	0	3	2
6	Henard, Christopher	0	0	1	3	0	1	0	0	0	0	5	0
5	Pereira, J.A.	0	0	1	0	1	0	0	0	0	0	0	1
5	Thüm, Thomas	1	0	2	1	2	1	0	1	0	1	1	0
5	Galindo, José A	0	1	2	1	1	0	0	0	0	0	3	0
5	Pleuss, Andreas	0	0	3	1	2	0	0	0	0	4	0	0
4	Apel, Sven	0	0	1	1	0	0	0	0	0	0	2	0
4	Berger, Thorsten	0	0	1	0	3	0	0	0	0	1	0	3
4	Arcaini, P.	0	0	0	2	0	0	0	0	0	1	1	0
4	Mazo, Raúl	0	0	1	1	0	0	0	1	0	0	1	0
4	Bagheri, Ebrahim	0	1	2	1	2	0	0	1	0	0	3	0

Table 6: First authors having more than four papers in the survey

# papers	Domain	Country	Institution
15	irisa.fr	France	Institut National de Recherche en Informatique et Automatique de Rennes
12	us.es	Spain	University of Seville
7	uni.lu	Luxembourg	Université du Luxembourg
7	berkeley.edu	USA	University of Berkeley
7	etsmtl.ca	Canada	Université du Québec
7	isti.cnr.it	Italy	Institute of the National Research Council of Italy
5	osumc.edu	USA	Ohio State university
5	lero.ie	Ireland	Lero - The Irish Software Research Centre
4	uni-passau.de	Germany	Universität Passau
4	univ-paris1.fr	France	Unversité Paris 1
4	tu-dresden.de	Germany	Technische Universität Dresden
4	unamur.be	Belgium	Université de Namur
4	chalmers.se	Sweden	Chalmers University of technology
4	ryerson.ca	Canada	Ryerson University
4	mcmaster.ca	Canada	McMaster University
4	ovgu.de	Denmark	Otto-von-Guericke-Universitaet Magdeburg

Table 7: Institutions working on AAFM

5.3 RQ3: What are the areas for which AAFM has been applied?

To discover why researchers are using AAFM methods, we explored the different identified AAFM topics showing our major findings and identifying potential research gaps. These research gaps are subjective interpretations according to the screening of papers and the experience of the authors. Also, we show quantitative data such as the number of identified papers per topic. This is the output of the process defined in Section 4.3.1, where keywords were assigned to papers after reading their titles and abstracts and required portions. For this purpose, only the filtered batch of papers was used (see Figure 7).

Product configuration and derivation. Papers that deal with the configuration and/or further product derivation of feature models are in this category. The configuration of FMs can be defined as the process of selecting and deselecting features in a FM to obtain a concrete product instance. After and complementary to the previous task, the product derivation process is activated. Product derivation uses concrete composition mechanism and variability management techniques to obtain a working product [6].

A total of sixty-nine papers were also classified in this category, making this topic the one that attracted more attention. For example, the use of feature-oriented techniques to implement SPLs has been extensively documented (e.g. [245]). Also, researchers proposed the improvement of the scalability of product configuration [216]. Finally, there are works that focus on configuration, such as Asadi et al.[47], who provide mechanisms to configure feature models while optimizing non-functional properties, and [245], where a tool to guide practitioners while configuring SPLs is presented.

Research opportunities

Traditionally, configuration technologies are applied in a closed SPL context, this is, all or most of the features are available in a controlled and closed environment, where all the decisions for configuring a product are taken by a very small group of people. An emerging challenge in this area is the configuration of a diversity of distributed SPL descriptions, sometimes known as *software ecosystems*. As SPLs grow, configuring and maintaining them become an unfeasible task for a small group. Also, different stakeholders and privacy policies encourage the use of visibility restrictions for the configurable parts. We think that there is still work to be done in this area like, for example, to enable the parallel configuration of those open and distributed SPLs.

Testing and evolution. When a large number of products is encoded in a FM the testing and evolution processes become expensive and tedious. Two main approaches have been followed to reduce the testing costs. First, combinatorial testing and more concretely T-wise methods to narrow the number of products [130]. Second, test prioritisation to order the execution of critical tests in a time or resource constrained environment [214]. These techniques become also relevant when coping with software evolution. SPL evolution happens when it is required to add, remove or modify features or relationships to an existing SPL. In this scenario, it is required to test if the SPL is error-free before, during and after the evolution actually happens. In this case, the rationale for grouping evolution and testing is that, after the keywording process, some papers were in both sub-areas, probably, because testing is a key aspect to consider when evolving variability-intensive systems.

We have identified sixty-two papers referring to SPL testing and evolution. We also noticed that the works coping with testing costs reduction are focusing in scenarios where there is more than one objective –maybe contradictory– to be satisfied at the same time. This is, to optimize different aspects of the same test-suite [130]. For example, to find the test-suite that minimises the testing cost while maximising the market-share. Recently, researchers have been looking for the most convenient evolutionary algorithm to test and select best SPL products, finding that IBEA was returning better results than NSGA-II [216] when coping with multi-objective testing objectives. In terms of evolution management different automated analysis have been proposed to guarantee the safe transition between the different evolution phases of the SPL. For example, White et al. [258], propose the use of CSP solvers to grant the validity of so-called FM drifts.

Research opportunities

A challenging task is to evolve a SPL while maintaining support for existing products. This is, to verify that the products already being used are still valid in the next evolution of the SPL. SPL researchers use information encoded in variability models when selecting and prioritising test-cases. However, other information related to SPL activities such source code management, bug tracking system can be considered to select and prioritise test-cases. The exploration of different testing techniques such as metamorphic, mutation, graph-based among others can still explored in the SPL context.

Reverse engineering. There are two main strategies to adopt an SPL approach. First, a proactive approach when a company already starts by planning the construction of the SPL. Second, a reactive approach when individual products are first developed one after another and at a certain point, when the number of similar products is big enough, the company transitions to an SPL engineering approach. To help in this transition, researchers have proposed several reverse-engineering methods.

The last years have been fruitful in this area with a total of twenty-two papers classified. Researchers have extracted variability encoded in product comparison matrices[215] and CNF formulas [39]. Moreover, Becan et al. [54] worked on benefiting from ontological knowledge to help in the task of reverse engineering.

Research opportunities

Nowadays, the variability description of variability-intensive systems is getting more complex. This is done by introducing non-boolean information [145], and by using several variability models [118] in a multi-layer fashion. However, we found no proposals to reverse engineer the existing variability information of variability-intensive systems with more than one variability model or non-boolean information. Also, researchers had not relied on low level assets such as source code or package management systems descriptions when reverse engineering feature models. This kind of assets can provide valuable information because they can contain implicit variability information to be exploited in the reverse engineering task.

Multi-model variability analysis. The AAFM started by only considering one FM description at a time. However, the more the SPL becomes larger, the more complex the variability description becomes. Nowadays, it is common to find the variability description of a variability-intensive system in several variability models that can be depicted using different formats.

A total of twenty-five papers cope with multi-model SPLs. For example, Dhungana et al. [90], proposed the distributed modelling of SPLs providing examples from the industry. Also, new tools have been provided to determine the set of implementation artifacts that supports a concrete feature specification [29]. But also, because of the need of describing a variability-intensive system in smaller artifacts so it can be maintained by different practitioners, new operations to merge models appeared, thus introducing new FM operations such as model merging or slicing [30].

Research opportunities

There is a lack of support for quality attributes when coping with multi-model SPL descriptions. For example, we did not find techniques showing how quality attributes domains are impacted by the selection of certain features at specification level. Also, it is possible to implement distributed analyses of FMs when having multiple models. For example, to first slice the model, and then distribute the analysis in different computation nodes and finally merge the results. This can be helpful in scenarios with large models such as the Linux kernel containing more than 8,000 features.

Variability modelling. Encoding the variabilities and commonalities of a SPL requires to find a trade–off between the expressivity of the language and its usability [20]. There are multiple ways of encoding variability such as the one described in [4]. Some variability modelling languages in real context have been analysed [85] to learn how variability is managed and modelled in realistic systems [61].

This area has gained momentum in the last years (see Figure 9), grouping a total of sixty papers. Several proposals appeared to cover different domain specific requirements. For example, new cardinalities regarding features have been considered [175] to enable the proper description of systems when a feature is present multiple times in a configuration. Existing languages for feature modelling have been documented[101] and new language constructs have been introduced to improve the analysis of FMs. For example, FAMILIAR [35] introduced different constructs and translations to solvers that make the AAFM scaling over previous approaches.

Research opportunities

Nowadays, AAFM is applied beyond product lines in scenarios like cloud computing or the operating systems domain [94] (see the next challenge). We think that the use of new available information in those systems can be used to leverage the AAFM in other domains by means of the definition of new analysis operations or the composition of existing ones. This motivates the need of new constructs encoding this information and thus, improving the AAFM.

Variability intensive systems analysis. The AAFM was initially developed thinking in SPLs. However, nowadays it is used for a widespread number of applications. Currently, the scenarios where AAFM is used range from variability analysis in other domains such as mobile apps testing [117], cloud systems [119] or bioinformatics [84]. Note that there is a clear trend of using AAFM for systems beyond SPL however, authors tend to use it for a concrete scenario such as testing [117] or configuration [119].

A total of twenty-three papers referring to this trend were identified within the corpus of papers. Different variability models benefit from automated analysis techniques to extract information in a similar way. While diverse models such as OVM [209] and DOPLER [170] models had automated analysis support in 2010, now, there are other models such as BPMNs [87], that use similar techniques to extract information and metrics. In the open source domain, the AAFM was applied to existing systems such as the Linux [232] kernel. Also, the AAFM has profited from the open-source community which helped to understand the nature of models and the constraints flavours existing in the wild [180].

Research opportunities

The number of different scenarios where AAFM can be applied have been proved to be very large and we envision that there will be more and more scenarios where it will be applied in the future. Ecosystems, cyber-physical systems, robotics, big data or the internet of the things are only some examples where variability is a first-class citizen and can use AAFM techniques in their development.

5.3.1 Quantitative analysis

The variability context dimension of this mapping study aims to identify the current trends in the area of AAFM. In the previous Section, we already presented them while highlighting some of the most interesting findings. Now we present quantitative data that provides the weight of each trend in the area.

Table 8 presents the distribution of the different trends of the full set of papers. We see that the areas that more attracted researchers attention were testing and evolution, product configuration and variability modelling. When we take a look to the list of papers we see that the product configuration was more present within the time lapse considered for this research while the use of AAFM in a diversity of systems is more recent. Also, we see that, because of this new AAFM usage the variability, modelling papers started to introduce new forms of variability in the reasoning process such as features cardinalities.

Variability context	Citations	#papers
Product configuration and derivation	[35, 37, 42, 46, 47, 49, 50, 53, 55, 56, 57, 58, 59, 60, 65, 66, 68, 72, 74, 75, 76, 77, 80, 82, 83, 89, 93, 95, 98, 99, 100, 103, 105, 106, 107, 108, 109, 110, 112, 114, 116, 118, 119, 120, 123, 125, 126, 128, 133, 134, 135, 136, 138, 140, 144, 145, 146, 149, 153, 154, 155, 156, 161, 169, 171, 172, 173, 174, 176, 177, 178, 182, 183, 185, 189, 190, 191, 194, 195, 197, 198, 199, 201, 204, 205, 206, 208, 209, 210, 216, 219, 221, 222, 224, 234, 236, 237, 238, 242, 244, 245, 246, 247, 249, 250, 251, 252, 253, 259, 260, 263, 264, 265, 266, 267]	115
Testing and evolution	[28, 33, 37, 38, 41, 43, 44, 45, 48, 52, 64, 67, 69, 72, 73, 83, 91, 92, 97, 104, 105, 107, 111, 113, 114, 115, 117, 121, 122, 124, 128, 129, 130, 132, 134, 135, 139, 141, 143, 144, 150, 152, 159, 160, 161, 162, 164, 165, 166, 168, 170, 176, 183, 186, 187, 188, 192, 193, 195, 196, 200, 207, 212, 213, 214, 222, 225, 226, 227, 228, 229, 230, 235, 239, 241, 244, 248, 250, 253, 254, 255, 256, 257, 258, 261]	85
Reverse engineering	[31, 34, 36, 39, 48, 54, 56, 70, 86, 126, 127, 131, 134, 144, 148, 149, 151, 157, 163, 164, 180, 181, 191, 215, 233, 244, 268]	27
Multi-model variability analysis	[29, 30, 32, 33, 34, 50, 65, 71, 73, 81, 89, 90, 100, 113, 118, 121, 134, 144, 158, 179, 182, 206, 210, 220, 222, 237, 241, 246, 262]	29
Variability modelling	[40, 47, 49, 51, 54, 55, 56, 58, 61, 62, 63, 64, 70, 71, 78, 79, 80, 81, 84, 85, 88, 91, 94, 101, 102, 108, 109, 110, 118, 134, 136, 137, 139, 141, 142, 144, 145, 146, 147, 149, 157, 158, 167, 169, 175, 179, 183, 184, 194, 197, 199, 202, 203, 210, 211, 217, 218, 219, 221, 223, 224, 225, 231, 232, 240, 242, 243, 244, 252, 261, 262, 265, 266, 269]	74
Variability intensive systems analysis	[28, 46, 55, 56, 57, 65, 68, 73, 84, 86, 87, 91, 94, 119, 120, 134, 153, 156, 167, 178, 190, 204, 213, 214, 232, 244, 259, 267, 269]	29

Table 8: Classification of papers based on the variability context facet

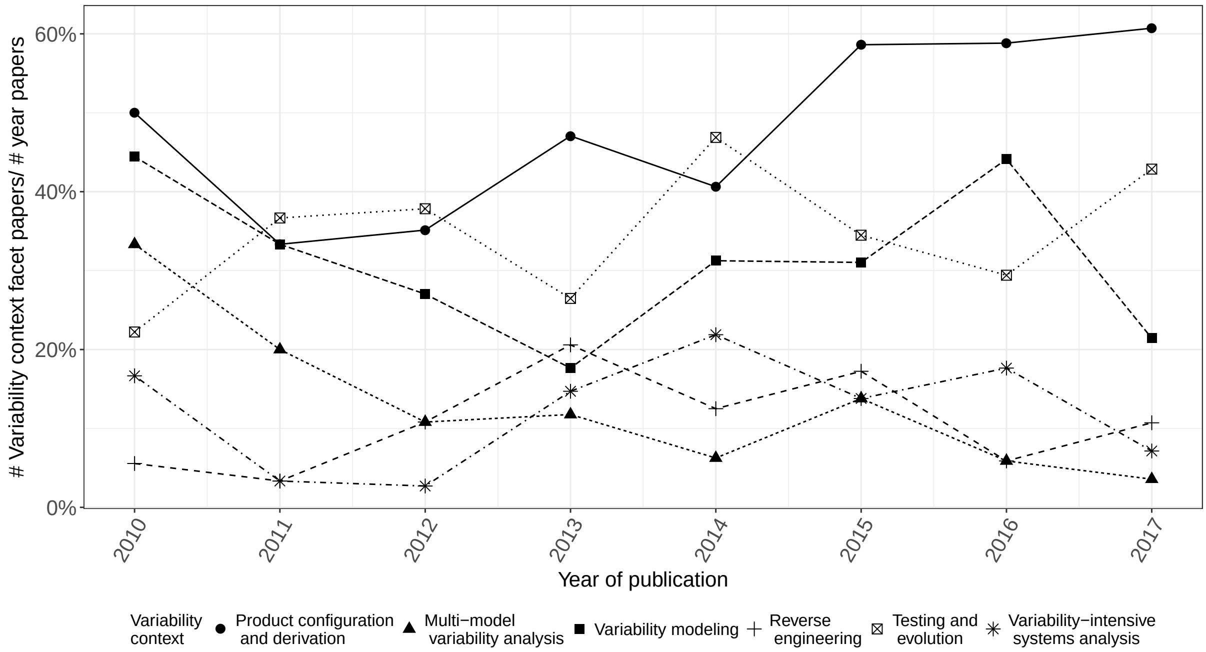


Fig. 9: Temporary distribution of the variability context facet

Figure 9 shows the evolution of the different trends depending on the year of publication. In the y axis, the percentage of papers per year of each variability context facet is presented which can show the interest of the facet in the corresponding year. We see that there are some areas that stayed with minor variations, such as multi-model variability analysis and variability modelling, while there are others that clearly have hyped during the last years, such as reverse engineering.

We want to highlight that *"Product configuration and derivation"* has a good percentage of papers over the time span of the review. This is probably due to the fact that it is one of the initial and more established usages of AAFM. Also, the contributions targeting *"Multi-model variability analysis"* had an stable percentage of papers every year, and has not hyped yet. We conjecture that this is due to the fact of the complexity of the underling analysis on multiple models and the lack of well recognized case studies and examples on this field. Finally, another observation is that before 2014 there were two groups of trends that evolved similarly. First, *"Multi-model variability analysis"*, *"Reverse engineering"* and *"Variability intensive systems analysis"*; and second, *"Product configuration and derivation"*, *"Testing and evolution"* and *"Variability modelling"*. In the last years, we see the hype of *"Variability modelling"*. We conjecture that the hype of this last trend is due to industrial needs of specifying specific variability properties within their domains [2].

5.4 RQ4: Which kind of publications are used to address the challenges?

The research facet classifies the papers depending on its research type. In Table 9 we observe the number of papers within our set of papers that fits each research type.

We observe that there is a clear higher amount of papers on the evaluation, philosophical and solution proposal papers categories. This indicates that even this area is maturing and we can foresee some papers focusing on real-world problems, most papers are not related to the industry. We notice that most of the research lacks validation and focuses on evaluating concrete theoretical approaches. This pinpoints the need for this community to work closer to the industry and to validate methods and techniques in a more practical way.

Research facet	Citations	#papers
Opinion Paper	[66, 73, 75, 144, 164, 193, 248]	7
Philosophical Paper	[51, 74, 99, 101, 134, 137, 171, 175, 196, 201, 244]	11
Solution Proposal	[30, 35, 44, 58, 60, 65, 72, 78, 79, 80, 81, 93, 102, 106, 140, 141, 142, 145, 151, 158, 162, 165, 172, 176, 184, 185, 191, 195, 197, 198, 199, 200, 203, 205, 207, 210, 217, 219, 220, 225, 236, 239, 243, 246, 247, 262, 264, 265, 268]	49
Evaluation Research	[28, 29, 32, 33, 36, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 48, 49, 50, 52, 54, 55, 56, 59, 67, 68, 69, 70, 71, 77, 82, 83, 85, 86, 87, 88, 89, 90, 92, 94, 96, 97, 98, 100, 103, 104, 107, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 135, 138, 139, 143, 146, 147, 148, 149, 150, 152, 153, 154, 155, 157, 159, 160, 161, 163, 166, 167, 168, 169, 170, 173, 174, 177, 178, 179, 180, 181, 182, 183, 186, 188, 189, 190, 192, 202, 204, 206, 208, 209, 212, 214, 215, 216, 218, 222, 223, 224, 226, 227, 228, 229, 230, 231, 233, 234, 235, 237, 238, 240, 241, 245, 249, 250, 252, 255, 256, 257, 258, 260, 261, 263, 266, 267, 269]	148
Validation Research	[31, 34, 57, 61, 62, 63, 84, 91, 105, 136, 156, 187, 194, 211, 213, 221, 232, 251, 253, 254, 259]	21
Experience Report	[53, 64, 76, 95, 108, 242]	6

Table 9: Classification of papers based on the research facet

5.5 RQ5: When have the papers been published?

In this question, we want to determine the dates where the papers were published. Also, we want to see how the number of papers published in the different fora types used across the paper evolved across the different years. To answer this question we relied on the first batch of papers containing 423 contributions. Note that the papers are normalized by the number primary studies published that year.

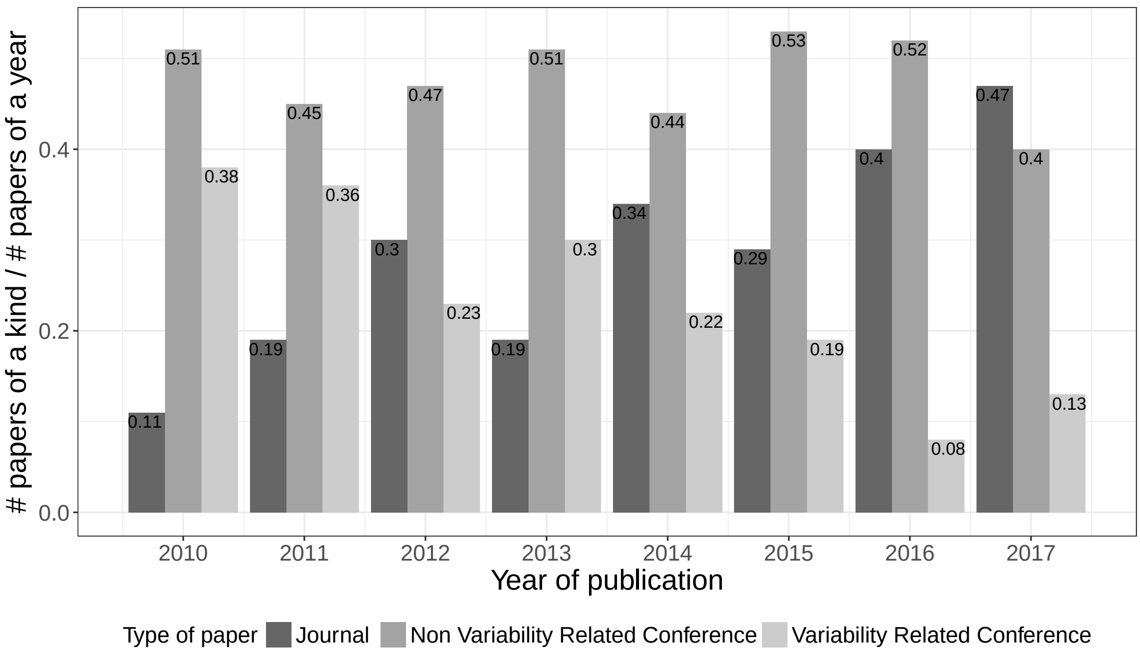


Fig. 10: Temporary distribution of papers.

We observe in Figure 10 that there is an increment of papers published in journals. This is specially remarkable in the years 2017 were the papers published in journals were more than in conferences and workshops. This points out that AAFM research works have matured when published. Also, the number of papers published in SPLC, VAMOS, ICSE, and ASE is lower than the number of papers published in the rest of conferences when taking into account the publication year.

The difference between variability related conferences and the non related ones stays mostly unaltered in the last three years covered by this study. However, we can see that the percentage of variability related conferences paper is decreasing since 2013. We conjecture that this change in the trend is due to the fact that the AAFM is now more widespread in other communities and therefore, non variability related conferences accept more papers on this topic.

5.6 RQ6: How are the papers related among them?

This question aims at finding research gaps to drive future work in the area of AAFM. By identifying the variability-context and type of papers where there are fewer publications we can infer the lacks of current research.

To answer this question we relied on the reduced dataset only considering variability-related conferences and journals. It is interesting to see the global distribution (see Figure 11) of the number of papers based on the two main dimensions considered in this mapping study. This heatmap shows the research gaps where the SPL community should invest more efforts.

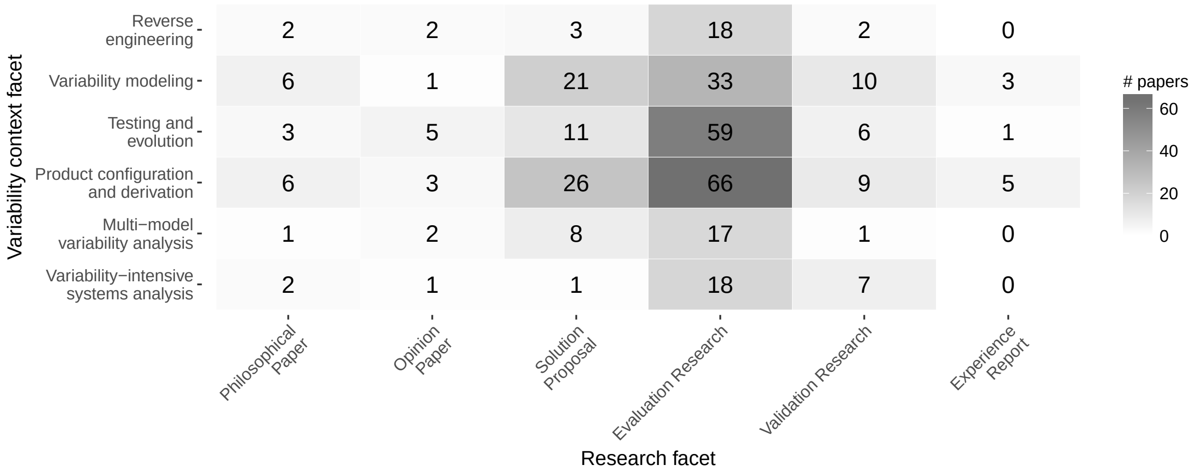


Fig. 11: Visualization of the systematic map

Firstly, we observe that there is still room for improvement regarding experience reports because only eleven contributions fall in this category. Moreover, we almost found papers about product configuration and derivation and variability modelling in such research facet. This is a clear gap that evidences the distance to industry of the other trends. Secondly, we observe that opinion and philosophical papers have low weight which may show that the discipline is getting mature because there are taxonomies, literature reviews or mapping studies on the different trends. Finally, we see that most of the papers are on the evaluation research category and testing and evolution context facet, i.e. the papers evaluate the solution in front of small examples or datasets. We conjecture that this is due to the fact of the availability of well known examples and datasets on these trends.

Generally speaking, we observe that the area is getting mature but, still misses some more collaboration with industry. In fact, those variability contexts that lately attracted the most researchers attention did not have time to report on experience reports. Also, we observe a tendency of publishing conceptual ideas that are later evaluated in terms of efficiency (evaluation) instead of customer satisfaction (validation).

6 Conclusions

In this paper we went through different research questions to understand the current state of the area of AAFM by applying a systematic mapping method. We identified where relevant papers are being published; who are the authors and institutions that currently holds the know-how of AAFM; which trends in the usage of AAFM attracted more researchers attention in the last years; what kind of publications are being proposed; when and where the papers were published.

The surveyed primary sources point out that the AAFM is a subject that is getting mature and that has driven some other subjects such as product configuration, derivation, testing, evolution and reverse engineering. However, regarding the distribution depending on the nature of primary sources, the main fact detected is that FM practitioners are not validating the research as much as they evaluate it. This actually should encourage the community to work closer to industry and provide better ready-to-the-market solutions instead of toy techniques and prototypes.

The aim of this study is to guide future research on the application of the AAFM in new domains such as images creation, cloud management, mobile computing, operating systems dependencies or internet of the things among others. We believe that it is time to stop gleaming luster to the analysis techniques that are in general mature enough and find application domains and real evidence where the AAFM can be successfully applied.

Acknowledgements

This work was supported, in part, by the European Commission (FEDER), by the Spanish government under BELi (TIN2015-70560-R) project and by the Andalusian government under the COPAS (TIC-1867) project. You can find all the material used in this paper in the website `https://isa-group.github.io/aafm-quo-vadis/`.

References

1. Acher M, Collet P, Lahire P, France RB (2013) FAMILIAR: A domain-specific language for large scale management of feature models. Science of Computer Programming (SCP) 78(6):657–681
2. Alférez M, Acher M, Galindo JA, Baudry B, Benavides D (2018) Modeling variability in the video domain: language and experience report. Software Quality Journal DOI 10.1007/s11219-017-9400-8
3. Batory D, Benavides D, Ruiz-Cortes A (2006) Automated analysis of feature models: Challenges ahead. Commun ACM 49(12):45–47, DOI 10.1145/1183236.1183264
4. Benavides D, Segura S, Ruiz-Cortés A (2010) Automated analysis of feature models 20 years later. Information Systems 35(6):615–636
5. Benavides D, Trinidad P, Cortés AR, Segura S (2013) FaMa, Springer Berlin Heidelberg, chap FaMa, pp 163–171. DOI 10.1007/978-3-642-36583-6-11
6. Capilla R (2013) Variability realization techniques and product derivation. In: Systems and Software Variability Management, Springer, pp 87–99
7. Clements P, Northrop L (2002) Software product lines. Addison-Wesley Boston
8. Durán A, Benavides D, Segura S, Trinidad P, Ruiz-Cortés A (in press) FLAME: a formal framework for the automated analysis of software product lines validated by automated specification testing. Software & Systems Modeling pp 1–34, DOI 10.1007/s10270-015-0503-z
9. Engström E, Runeson P (2011) Software product line testing – a systematic mapping study. Information and Software Technology 53(1):2 – 13, DOI http://dx.doi.org/10.1016/j.infsof.2010.05.011
10. Galindo J, Turner H, Benavides D, White J (2014) Testing variability-intensive systems using automated analysis: an application to android. Software Quality Journal pp 1–41, DOI 10.1007/s11219-014-9258-y
11. Grant MJ, Booth A (2009) A typology of reviews: an analysis of 14 review types and associated methodologies. Health Information & Libraries Journal 26(2):91–108
12. Heradio R, Perez-Morago H, Fernandez-Amoros D, Cabrerizo FJ, Herrera-Viedma E (2015) A science mapping analysis of the literature on software product lines. In: Fujita H, Guizzi G (eds) Intelligent Software Methodologies, Tools and Techniques, Communications in Computer and Information Science, vol 532, Springer International Publishing, pp 242–251, DOI 10.1007/978-3-319-22689-718

13. Heradio R, Perez-Morago H, Fernandez-Amoros D, Cabrerizo FJ, Herrera-Viedma E (2016) A bibliometric analysis of 20 years of research on software product lines. Information and Software Technology 72:1 – 15, DOI http://dx.doi.org/10.1016/j.infsof.2015.11.004
14. Jia C, Cai Y, Yu YT, Tse TH (2016) 5W+1H pattern: A perspective of systematic mapping studies and a case study on cloud software testing. Journal of Systems and Software 116:206–219, DOI 10.1016/j.jss.2015.01.058
15. Kang KC, Cohen SG, Hess JA, Novak WE, Peterson AS (1990) Feature-oriented domain analysis (foda) feasibility study. Tech. rep., DTIC Document
16. Kipling R (1902) Just so stories. MacMillan, London, UK
17. Kitchenham B, Brereton OP, Budgen D, Turner M, Bailey J, Linkman S (2009) Systematic literature reviews in software engineering – a systematic literature review. Information and Software Technology 51(1):7 – 15, DOI http://dx.doi.org/10.1016/j.infsof.2008.09.009, special Section - Most Cited Articles in 2002 and Regular Research Papers
18. Laguna MA, Crespo Y (2013) A systematic mapping study on software product line evolution: From legacy system reengineering to product line refactoring. Science of Computer Programming 78(8):1010 – 1034, DOI http://dx.doi.org/10.1016/j.scico.2012.05.003
19. Lopez-Herrejon RE, Linsbauer L, Egyed A (2015) A systematic mapping study of search-based software engineering for software product lines. Information & Software Technology 61:33–51, DOI 10.1016/j.infsof.2015.01.008
20. Méndez-Acuña D, Galindo JA, Degueule T, Combemale B, Baudry B (2016) Leveraging software product lines engineering in the development of external dsls: A systematic literature review. Computer Languages, Systems & Structures 46:206–235, DOI 10.1016/j.cl.2016.09.004
21. Mendonca M, Branco M, Cowan D (2009) S.p.l.o.t.: Software product lines online tools. In: Proceedings of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications, ACM, New York, NY, USA, OOPSLA '09, pp 761–762, DOI 10.1145/1639950.1640002
22. Montalvillo L, Díaz O (2016) Requirement-driven evolution in software product lines: A systematic mapping study. Journal of Systems and Software 122:110 – 143, DOI http://dx.doi.org/10.1016/j.jss.2016.08.053
23. da Mota Silveira Neto PA, do Carmo Machado I, McGregor JD, de Almeida ES, de Lemos Meira SR (2011) A systematic mapping study of software product lines testing. Information and Software Technology 53(5):407 – 423, DOI http://dx.doi.org/10.1016/j.infsof.2010.12.003, special Section on Best Papers from {XP2010}
24. Petersen K, Feldt R, Mujtaba S, Mattsson M (2008) Systematic mapping studies in software engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, British Computer Society, Swinton, UK, UK, EASE'08, pp 68–77
25. Petersen K, Feldt R, Mujtaba S, Mattsson M (2008) Systematic mapping studies in software engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, BCS Learning & Development Ltd., Swindon, UK, EASE'08, pp 68–77
26. Schobbens P, Heymans P, Trigaux J, Bontemps Y (2007) Generic semantics of feature diagrams. Computer Networks 51(2):456–479, DOI 10.1016/j.comnet.2006.08.008
27. Wieringa R, Maiden N, Mead N, Rolland C (2006) Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requirements Engineering 11(1):102–107

Appendix A. Primary sources

28. Abal I, Brabrand C, Wasowski A (2014) 42 variability bugs in the linux kernel: a qualitative analysis. In: ASE, DOI 10.1145/2642937.2642990
29. Abele A, Papadopoulos Y, Servat D, Törngren M, Weber M (2010) The cvm framework-a prototype tool for compositional variability management. In: VAMOS, vol 10, pp 101–105
30. Acher M, Collet P, Lahire P, France R (2011) Slicing feature models. In: ASE, pp 424–427, DOI 10.1109/ASE.2011.6100089
31. Acher M, Cleve A, Perrouin G, Heymans P, Vanbeneden C, Collet P, Lahire P (2012) On extracting feature models from product descriptions. In: VAMOS, pp 45–54, DOI 10.1145/2110147.2110153
32. Acher M, Collet P, Gaignard A, Lahire P, Montagnat J, France R (2012) Composing multiple variability artifacts to assemble coherent workflows. SQJ 20(3-4):689–734, DOI 10.1007/s11219-011-9170-7
33. Acher M, Collet P, Lahire P, France R (2012) Separation of concerns in feature modeling: Support and applications. In: AOSD, pp 1–12, DOI 10.1145/2162049.2162051
34. Acher M, Baudry B, Heymans P, Cleve A, Hainaut JL (2013) Support for reverse engineering and maintaining feature models. In: VAMOS, ACM, p 20, DOI 10.1145/2430502.2430530
35. Acher M, Collet P, Lahire P, France RB (2013) Familiar: A domain-specific language for large scale management of feature models. SCP 78(6):657–681, DOI 10.1016/j.scico.2012.12.004
36. Acher M, Cleve A, Collet P, Merle P, Duchien L, Lahire P (2014) Extraction and evolution of architectural variability models in plugin-based systems. SOSYM 13(4):1367–1394, DOI 10.1007/s10270-013-0364-2
37. Ajoudanian S, Hosseinabadi SH (2015) Automatic promotional specialization, generalization and analysis of extended feature models with cardinalities in alloy. LAMP 84(5):640–667, DOI 10.1016/j.jlamp.2014.11.005
38. Al-Hajjaji M, Thüm T, Meinicke J, Lochau M, Saake G (2014) Similarity-based prioritization in software product-line testing. In: SPLC, ACM, pp 197–206, DOI 10.1145/2648511.2648532
39. Andersen N, Czarnecki K, She S, Wa¸sowski A (2012) Efficient synthesis of feature models. In: SPLC, vol 1, pp 106–115, DOI 10.1145/2362536.2362553
40. Antkiewicz M, Bąk K, Murashkin A, Olaechea R (2013) Clafer tools for product line engineering. In: SPLC, DOI 10.1145/2499777.2499779
41. Apel S, Speidel H, Wendler P, Von Rhein A, Beyer D (2011) Detection of feature interactions using feature-aware verification. In: ASE, IEEE Computer Society, pp 372–375, DOI 10.1109/ase.2011.6100075
42. Apel S, Von Rhein A, ThüM T, KäStner C (2013) Feature-interaction detection based on feature-based specifications. CNJ 57(12):2399–2409, DOI 10.1016/j.comnet.2013.02.025
43. Arcaini P, Gargantini A, Vavassori P (2015) Generating tests for detecting faults in feature models. In: ICST, DOI 10.1109/ICST.2015.7102591
44. Arcaini P, Gargantini A, Vavassori P (2017) Automated repairing of variability models. In: SPLC, vol 1, pp 9–18, DOI 10.1145/3106195.3106206
45. Arrieta A, Sagardui G, Etxeberria L, Zander J (2017) Automatic generation of test system instances for configurable cyber-physical systems. SQJ 25(3):1041–1083, DOI 10.1007/s11219-016-9341-7
46. Asadi M, Mohabbati B, Gröner G, Gasevic D (2014) Development and validation of customized process models. JSS 96:73–92, DOI 10.1016/j.jss.2014.05.063
47. Asadi M, Soltani S, Gasevic D, Hatala M, Bagheri E (2014) Toward automated feature model configuration with optimizing non-functional requirements. IST 56(9):1144–1165, DOI 10.1016/j.infsof.2014.03.005
48. Assunção W, Lopez-Herrejon R, Linsbauer L, Vergilio S, Egyed A (2017) Multi-objective reverse engineering of variability-safe feature models based on code dependencies of system variants. ESE 22(4):1763–1794, DOI 10.1007/s10664-016-9462-4
49. Bagheri E, Asadi M, Gasevic D, Soltani S (2010) Stratified analytic hierarchy process: Prioritization and selection of software features. In: SPLC, vol 6287 LNCS, pp 300–315, DOI 10.1007/978-3-642-15579-6_21
50. Bagheri E, Di Noia T, Ragone A, Gasevic D (2010) Configuring software product line feature models based on stakeholders' soft and hard requirements. In: SPLC, Springer, pp 16–31, DOI 10.1007/978-3-642-15579-6_2

51. Bagheri E, Noia TD, Gasevic D, Ragone A (2012) Formalizing interactive staged feature model configuration. JSEP 24(4):375–400, DOI 10.1002/smr.534
52. Bagheri Eb, Gasevic D (2011) Assessing the maintainability of software product line feature models using structural metrics. SQJ 19(3):579–612, DOI 10.1007/s11219-010-9127-2
53. Baresi L, Guinea S, Pasquale L (2012) Service-oriented dynamic software product lines. Computer 45(10):42–48, DOI 10.1109/MC.2012.289
54. Bécan G, Acher M, Baudry B, Nasr SB (2015) Breathing ontological knowledge into feature model synthesis: an empirical study. ESE pp 1–48, DOI 10.1007/s10664-014-9357-1
55. ter Beek M, Legay A, Lafuente A, Vandin A (2016) Statistical model checking for product lines. In: ISOLA, vol 9952 LNCS, pp 114–133, DOI 10.1007/978-3-319-47166-2_8
56. ter Beek M, Reniers M, de Vink E (2016) Supervisory controller synthesis for product lines using cif 3. In: ISOLA, vol 9952 LNCS, pp 856–873, DOI 10.1007/978-3-319-47166-2_59
57. ter Beek MH, Legay A, Lafuente AL, Vandin A (2015) Statistical analysis of probabilistic models of software product lines with quantitative constraints. In: SPLC, ACM, pp 11–15, DOI 10.1145/2791060.2791087
58. Beek MHt, De Vink EP (2014) Using mcrl2 for the analysis of software product lines. In: FormaliSE, ACM, pp 31–37, DOI 10.1145/2593489.2593493
59. Benavides D, Felfernig A, Galindo J, Reinfrank F (2013) Automated analysis in feature modelling and product configuration. In: ICSR, vol 7925 LNCS, pp 160–175, DOI 10.1007/978-3-642-38977-1_11
60. Berger T, She S, Lotufo R, Czarnecki K, Wasowski A (2010) Feature-to-code mapping in two large product lines. In: SPLC, Citeseer, pp 498–499, DOI 10.1007/978-3-642-15579-6_48
61. Berger T, She S, Lotufo R, Wa¸sowski A, Czarnecki K (2010) Variability modeling in the real: A perspective from the operating systems domain. In: ASE, pp 73–82, DOI 10.1145/1858996.1859010
62. Berger T, She S, Lotufo R, Wasowski A, Czarnecki K (2013) A study of variability models and languages in the systems software domain. TSE 39(12):1611–1640, DOI 10.1109/TSE.2013.34
63. Berger T, Lettner D, Rubin J, Grünbacher P, Silva A, Becker M, Chechik M, Czarnecki K (2015) What is a feature? a qualitative study of features in industrial software product lines. In: SPLC, vol 20-24-July-2015, pp 16–25, DOI 10.1145/2791060.2791108
64. Bezerra C, Andrade R, Monteiro J (2017) Exploring quality measures for the evaluation of feature models: a case study. JSS 131:366–385, DOI 10.1016/j.jss.2016.07.040
65. Boškovi M, Bagheri E, GaŠevi D, Mohabbati B, Kaviani N, Hatala M (2010) Automated staged configuration with semantic web technologies. IJSEKE 20(4):459–484, DOI 10.1142/S0218194010004827
66. Boucher Q, Perrouin G, Heymans Pb (2012) Deriving configuration interfaces from feature models: A vision paper. In: VAMOS, pp 37–44, DOI 10.1145/2110147.2110152
67. Brabrand C, Ribeiro M, Tolêdo T, Borba P (2012) Intraprocedural dataflow analysis for software product lines. In: AOSD, ACM, pp 13–24, DOI 10.1145/2162049.2162052
68. Buchmann T, Dotor A, Westfechtel B (2013) Mod2-scm: A model-driven product line for software configuration management systems. IST 55(3):630–650, DOI 10.1016/j.infsof.2012.07.010
69. Bürdek J, Kehrer T, Lochau M, Reuling D, Kelter U, Schürr A (2016) Reasoning about product-line evolution using complex feature model differences. ASEJ 23(4):687–733, DOI 10.1007/s10515-015-0185-3
70. Bécan G, Behjati R, Gotlieb A, Acher M (2015) Synthesis of attributed feature models from product descriptions. In: SPLC, vol 20-24-July-2015, pp 1–10, DOI 10.1145/2791060.2791068
71. Bąk K, Diskin Z, Antkiewicz M, Czarnecki K, Wąsowski A (2016) Clafer: unifying class and feature modeling. SOSYM 15(3):811–845, DOI 10.1007/s10270-014-0441-1
72. Camacho C, Llana L, Núñez A (2016) Cost-related interface for software product lines. LAMP 85(1):227–244, DOI 10.1016/j.jlamp.2015.09.009
73. Capilla R, Bosch J (2011) The promise and challenge of runtime variability. Computer 44(12):93–95, DOI 10.1109/MC.2011.382
74. Capilla R, Bosch J, Trinidad P, Ruiz-Cortés A, Hinchey M (2014) An overview of dynamic software product line architectures and techniques: Observations from research and industry. JSS 91(1):3–23, DOI 10.1016/j.jss.2013.12.038
75. Capilla R, Ortiz Ó, Hinchey M (2014) Context variability for context-aware systems. Computer 47(2):85–87, DOI 10.1109/mc.2014.33

76. Chavarriaga J, Rangel C, Noguera C, Casallas R, Jonckers V (2015) Using multiple feature models to specify configuration options for electrical transformers: An experience report. In: SPLC, vol 20-24-July-2015, pp 216–224, DOI 10.1145/2791060.2791091
77. Chen S, Erwig M (2011) Optimizing the product derivation process. In: SPLC, pp 35–44, DOI 10.1109/SPLC.2011.47
78. Chimiak-Opoka J, Demuth B (2011) Ocl tools report based on the ide4ocl feature model. ECEASST
79. Chrszon P, Dubslaff C, Klüppelholz S, Baier C (2016) Family-based modeling and analysis for probabilistic systems – featuring profeat. In: FASE, vol 9633, pp 287–304, DOI 10.1007/978-3-662-49665-7_17
80. Chrszon P, Dubslaff C, Klüppelholz S, Baier C (2017) Profeat: feature-oriented engineering for family-based probabilistic model checking. FAC 30(1):45–75, DOI 10.1007/s00165-017-0432-4
81. Classen A, Boucher Q, Heymans P (2011) A text-based approach to feature modelling: Syntax and semantics of tvl. SCP 76(12):1130–1143, DOI 10.1016/j.scico.2010.10.005
82. Classen A, Heymans P, Schobbens PY, Legay A (2011) Symbolic model checking of software product lines. In: ICSE, ACM, pp 321–330, DOI 10.1145/1985793.1985838
83. Cordy M, Schobbens PY, Heymans P, Legay A (2013) Beyond boolean product-line model checking: dealing with feature attributes and multi-features. In: ICSE, IEEE Press, pp 472–481, DOI 10.1109/icse.2013.6606593
84. Costa GCB, Braga R, David JMN, Campos F (2015) A scientific software product line for the bioinformatics domain. JBI 56:239–264, DOI 10.1016/j.jbi.2015.05.014
85. Czarnecki K, Grünbacher P, Rabiser R, Schmid K, Wąsowski A (2012) Cool features and tough decisions: a comparison of variability modeling approaches. In: VAMOS, ACM, pp 173–182, DOI 10.1145/2110147.2110167
86. Davril JM, Delfosse E, Hariri N, Acher M, Cleland-Huang J, Heymans P (2013) Feature model extraction from large collections of informal product descriptions. In: ESEC/FSE, pp 290–300, DOI 10.1145/2491411.2491455
87. Del-Río-Ortega A, Resinas M, Cabanillas C, Ruiz-Cortés A (2013) On the definition and design-time analysis of process performance indicators. IS 38(4):470–490, DOI 10.1016/j.is.2012.11.004
88. Dermeval Db, Tenório T, Bittencourt I, Silva A, Isotani S, Ribeiro M (2015) Ontology-based feature modeling: An empirical study in changing scenarios. ESA 42(11):4950–4964, DOI 10.1016/j.eswa.2015.02.020
89. Dhungana D, Seichter D, Botterweck G, Rabiser R, Grunbacher P, Benavides D, Galindo JA (2011) Configuration of multi product lines by bridging heterogeneous variability modeling approaches. In: SPLC, IEEE, pp 120–129, DOI 10.1109/SPLC.2011.22
90. Dhungana D, Seichter D, Botterweck G, Rabiser R, Grünbacher P, Benavides D, Galindo JA (2013) Integrating heterogeneous variability modeling approaches with invar. In: VAMOS, ACM, p 8, DOI 10.1145/2430502.2430514
91. Dintzner N, Van Deursen A, Pinzger M (2014) Extracting feature model changes from the linux kernel using fmdiff. In: VAMOS, ACM, p 22, DOI 10.1145/2556624.2556631
92. Dintzner N, van Deursen A, Pinzger M (2017) Analysing the linux kernel feature model changes using fmdiff. SOSYM 16(1):55–76, DOI 10.1007/s10270-015-0472-2
93. Diskin Z, Safilian A, Maibaum T, Ben-David S (2016) Faithful modeling of product lines with kripke structures and modal logic. SACS 26(1):69–122, DOI 10.7561/SACS.2016.1.69
94. Dougherty B, White J, Schmidt DC (2012) Model-driven auto-scaling of green cloud computing infrastructure. FGCS 28(2):371–378, DOI 10.1016/j.future.2011.05.009
95. Dumitrescu C, Mazo R, Salinesi C, Dauron A (2013) Bridging the gap between product lines and systems engineering: an experience in variability management for automotive model based systems engineering. In: SPLC, ACM, pp 254–263, DOI 10.1145/2491627.2491655
96. Dumitru H, Gibiec M, Hariri N, Cleland-Huang J, Mobasher B, Castro-Herrera C, Mirakhorli M (2011) On-demand feature recommendations derived from mining public product descriptions. In: ICSE, pp 181–190, DOI 10.1145/1985793.1985819
97. Duran M, Mussbacher G (2016) Investigation of feature run-time conflicts on goal model-based reuse. ISF 18(5):855–875, DOI 10.1007/s10796-016-9657-7
98. Duran-Limon H, Garcia-Rios C, Castillo-Barrera F, Capilla R (2015) An ontology-based product architecture derivation approach. TSE 41(12):1153–1168, DOI 10.1109/TSE.2015.2449854

99. Durán A, Benavides D, Segura S, Trinidad P, Ruiz-Cortés A (2017) Flame: a formal framework for the automated analysis of software product lines validated by automated specification testing. SOSYM 16(4):1049–1082, DOI 10.1007/s10270-015-0503-z
100. Díaz J, Pérez J, Garbajosa J (2015) A model for tracing variability from features to product-line architectures: A case study in smart grids. REJ 20(3):323–343, DOI 10.1007/s00766-014-0203-1
101. Eichelberger H, Schmid K (2014) Mapping the design-space of textual variability modeling languages: a refined analysis. STTT 17(5):559–584, DOI 10.1007/s10009-014-0362-x
102. El-Sharkawy S, Dederichs S, Schmid K (2012) From feature models to decision models and back again an analysis based on formal transformations. In: SPLC, ACM, pp 126–135, DOI 10.1145/2362536.2362555
103. El-Sharkawy S, Krafczyk A, Schmid K (2017) An empirical study of configuration mismatches in linux. In: SPLC, vol 1, pp 19–28, DOI 10.1145/3106195.3106208
104. Ensan F, Bagheri E, Gašević D (2012) Evolutionary search-based test generation for software product line feature models. AISE pp 613–628, DOI 10.1007/978-3-642-31095-9_40
105. Esfahani N, Elkhodary A, Malek S (2013) A learning-based framework for engineering feature-oriented self-adaptive software systems. TSE 39(11):1467–1493, DOI 10.1109/TSE.2013.37
106. Famelis M, Salay R, Chechik M (2012) Partial models: Towards modeling and reasoning with uncertainty. In: ICSE, IEEE, pp 573–583, DOI 10.1109/icse.2012.6227159
107. Felfernig A, Reiterer S, Stettinger M, Tiihonen J (2015) Intelligent techniques for configuration knowledge evolution. In: VAMOS, vol 21-23-January-2015, pp 51–58, DOI 10.1145/2701319.2701320
108. Felfernig A, Reiterer S, Stettinger M, Tiihonen J (2015) Towards understanding cognitive aspects of configuration knowledge formalization. In: VAMOS, vol 21-23-January-2015, pp 117–123, DOI 10.1145/2701319.2701327
109. Fernandes P, Werner C, Teixeira E (2011) An approach for feature modeling of context-aware software product line. JUCS 17(5):807–829
110. Fernandez-Amoros D, Heradio R, Cerrada C, Herrera-Viedma E, Cobo M (2017) Towards taming variability models in the wild. FAIA 297:454–465, DOI 10.3233/978-1-61499-800-6-454
111. Ferreira J, Vergilio S, Quinaia M (2017) Software product line testing based on feature model mutation. IJSEKE 27(5):817–839, DOI 10.1142/S0218194017500309
112. Ferreira T, Lima J, Strickler A, Kuk J, Vergilio S, Pozo A (2017) Hyper-heuristic based product selection for software product line testing. IEEECIM 12(2):34–45, DOI 10.1109/MCI.2017.2670461
113. Filho JBF, Barais O, Acher M, Le Noir J, Legay A, Baudry B (2014) Generating counterexamples of model-based software product lines. STTT 17(5):585–600, DOI 10.1007/s10009-014-0341-2
114. Finkel R, O'Sullivan B (2011) Reasoning about conditional constraint specification problems and feature models. AIEDAM 25(2):163–174, DOI 10.1017/S0890060410000600
115. Font J, Arcega L, Haugen O, Cetina C (2017) Leveraging variability modeling to address metamodel revisions in model-based software product lines. CLSS 48:20–38, DOI 10.1016/j.cl.2016.08.003
116. Galindo J, Acher M, Tirado J, Vidal C, Baudry B, Benavides D (2016) Exploiting the enumeration of all feature model configurations: A new perspective with distributed computing. In: SPLC, vol 16-23-September-2016, pp 74–78, DOI 10.1145/2934466.2934478
117. Galindo JA, Turner H, Benavides D, White J (2014) Testing variability-intensive systems using automated analysis: an application to android. SQJ pp 1–41, DOI 10.1007/s11219-014-9258-y
118. Galindo Je, Dhungana D, Rabiser R, Benavides D, Botterweck G, Grünbacher P (2015) Supporting distributed product configuration by integrating heterogeneous variability modeling approaches. IST 62(1):78–100, DOI 10.1016/j.infsof.2015.02.002
119. García-Galán J, Pasquale L, Trinidad P, Ruiz-Cortés A (2016) User-centric adaptation analysis of multi-tenant services. TAAS 10(4), DOI 10.1145/2790303
120. García-Galán J, García J, Trinidad P, Fernández P (2017) Modelling and analysing highly-configurable services. In: SPLC, vol 1, pp 114–122, DOI 10.1145/3106195.3106211
121. Ghanam Y, Maurer F (2010) Linking feature models to code artifacts using executable acceptance tests. In: SPLC, vol 6287 LNCS, pp 211–225, DOI 10.1007/978-3-642-15579-6_15
122. Gheyi R, Massoni T, Borba P (2011) Automatically checking feature model refactorings. JUCS 17(5):684–711
123. Guo J, White J, Wang G, Li J, Wang Y (2011) A genetic algorithm for optimized feature selection with resource constraints in software product lines. JSS 84(12):2208–2221, DOI 10.1016/j.jss.2011.06.026

124. Guo J, Wang Y, Trinidad P, Benavides D (2012) Consistency maintenance for evolving feature models. ESA 39(5):4987–4998, DOI 10.1016/j.eswa.2011.10.014
125. Guo J, Zulkoski E, Olaechea R, Rayside D (2014) Scaling exact multi-objective combinatorial optimization by parallelization. In: ASE, DOI 10.1145/2642937.2642971
126. Hariri N, Castro-Herrera C, Mirakhorli M, Cleland-Huang J, Mobasher B (2013) Supporting domain analysis through mining and recommending features from online product listings. TSE 39(12):1736–1752, DOI 10.1109/tse.2013.39
127. Haslinger EN, Lopez-Herrejon RE, Egyed A (2013) On extracting feature models from sets of valid feature combinations. In: FASE, Springer, pp 53–67, DOI 10.1007/978-3-642-37057-1_5
128. Heider W, Rabiser R, Grünbacher P (2012) Facilitating the evolution of products in product line engineering by capturing and replaying configuration decisions. STTT 14(5):613–630, DOI 10.1007/s10009-012-0229-y
129. Henard C, Papadakis M, Perrouin G, Klein J, Le Traon Y (2013) Assessing software product line testing via model-based mutation: An application to similarity testing. In: ICSTW, IEEE, pp 188–197, DOI 10.1109/ICSTW.2013.30
130. Henard C, Papadakis M, Perrouin G, Klein J, Le Traon Y (2013) Multi-objective test generation for software product lines. In: SPLC, pp 62–71, DOI 10.1145/2491627.2491635
131. Henard C, Papadakis M, Perrouin G, Klein J, Le Traon Y (2013) Towards automated testing and fixing of re-engineered feature models. In: ICSE, pp 1245–1248, DOI 10.1109/ICSE.2013.6606689
132. Henard C, Papadakis M, Perrouin G, Klein J, Heymans P, Traon Y (2014) Bypassing the combinatorial explosion: Using similarity to generate and prioritize t-wise test configurations for software product lines. TSE 40(7):650–670, DOI 10.1109/TSE.2014.2327020
133. Henard C, Papadakis M, Harman M, Le Traon Y (2015) Combining multi-objective search and constraint solving for configuring large software product lines. In: ICSE, IEEE, vol 1, pp 517–528, DOI 10.1109/icse.2015.69
134. Heradio R, Perez-Morago H, Fernandez-Amoros D, Cabrerizo F, Herrera-Viedma E (2015) A science mapping analysis of the literature on software product lines. CCIS 532:242–251, DOI 10.1007/978-3-319-22689-7_18
135. Heradio R, Perez-Morago H, Alférez M, Fernandez-Amoros D, Alférez GH (2016) Augmenting measure sensitivity to detect essential, dispensable and highly incompatible features in mass customization. EJOR 248(3):1066–1077, DOI 10.1016/j.ejor.2015.08.005
136. Heymans P, Boucher Q, Classen A, Bourdoux A, Demonceau L (2012) A code tagging approach to software product line development. STTT 14(5):553–566, DOI 10.1007/s10009-012-0242-1
137. Hidaka S, Tisi M, Cabot J, Hu Z (2016) Feature-based classification of bidirectional transformation approaches. SOSYM 15(3):907–928, DOI 10.1007/s10270-014-0450-0
138. Hierons R, Li M, Liu X, Segura S, Zheng W (2016) Sip: Optimal product selection from feature models using many-objective evolutionary optimization. TOSEM 25(2), DOI 10.1145/2897760
139. Hu J, Wang Q (2016) Extensions and evolution analysis method for software feature models. JS 27(5):1212–1229, DOI 10.13328/j.cnki.jos.004829
140. Hubaux A, Heymans Pb, Schobbens PY, Deridder D, Abbasi E (2013) Supporting multiple perspectives in feature-based configuration. SOSYM 12(3):641–663, DOI 10.1007/s10270-011-0220-1
141. Javed M (2014) Towards the maturity model for feature oriented domain analysis. CES 4(3):170
142. Jézéquel JM (2012) Model-driven engineering for software product lines. ISRN 2012
143. Johansen MF, Haugen Ø, Fleurey F (2012) An algorithm for generating t-wise covering arrays from large feature models. In: SPLC, ACM, pp 46–55, DOI 10.1145/2362536.2362547
144. Kang (2010) Foda: Twenty years of perspective on feature modeling. In: VAMOS
145. Karataş A, Oğuztüzün H, Doğru A (2010) Mapping extended feature models to constraint logic programming over finite domains. In: SPLC, vol 6287 LNCS, pp 286–299, DOI 10.1007/978-3-642-15579-6_20
146. Karatas A, Oguztüzün H, Dogru A (2013) From extended feature models to constraint logic programming. SCP DOI 10.1016/j.scico.2012.06.004
147. Karatas AS, Oguztüzün H (2016) Attribute-based variability in feature models. REJ 21(2):185–208, DOI 10.1007/s00766-014-0216-9
148. Kastner C, Dreiling A, Ostermann K (2014) Variability mining: Consistent semi-automatic detection of product-line features. TSE 40(1):67–82, DOI 10.1109/TSE.2013.45

149. Khoshnevis S, Shams F (2017) Automating identification of services and their variability for product lines using nsga-ii. FCS 11(3):444–464, DOI 10.1007/s11704-016-5121-6
150. Kim CHP, Marinov D, Khurshid S, Batory D, Souto S, Barros P, d'Amorim M (2013) Splat: lightweight dynamic analysis for reducing combinatorics in testing configurable systems. In: ESEC/FSE, ACM, pp 257–267, DOI 10.1145/2491411.2491459
151. Kolesnikov SS, Apel S, Siegmund N, Sobernig S, Kästner C, Senkaya S (2013) Predicting quality attributes of software product lines using software and network measures and sampling. In: VAMOS, ACM, p 6, DOI 10.1145/2430502.2430511
152. Kowal M, Ananieva S, Thüm T, Schaefer I (2017) Supporting the development of interdisciplinary product lines in the manufacturing domain. IFAC 50(1):4336–4341, DOI 10.1016/j.ifacol.2017.08.870
153. Leite A, Alves V, Rodrigues G, Tadonki C, Eisenbeis C, Melo A (2017) Dohko: an autonomic system for provision, configuration, and management of inter-cloud environments based on a software product line engineering method. UCCJournal 20(3):1951–1976, DOI 10.1007/s10586-017-0897-1
154. Lian XL, Zhang L (2017) Multi-objective optimization algorithm for feature selection in software product lines. IndianST 28(10):2548–2563, DOI 10.13328/j.cnki.jos.005130
155. Liang J, Ganesh V, Czarnecki K, Raman V (2015) Sat-based analysis of large real-world feature models is easy. In: SPLC, vol 20-24-July-2015, pp 91–100, DOI 10.1145/2791060.2791070
156. Liebig J, von Rhein A, Kästner C, Apel S, Dörre J, Lengauer C (2013) Scalable analysis of variable software. In: ESEC/FSE, ACM, pp 81–91, DOI 10.1145/2491411.2491437
157. Linsbauer L, Lopez-Herrejon R, Egyed A (2017) Variability extraction and modeling for product variants. SOSYM 16(4):1179–1199, DOI 10.1007/s10270-015-0512-y
158. Liu Y, Lai K, Dai G, Yuen M (2010) A semantic feature model in concurrent engineering. TASE DOI 10.1109/tase.2009.2039996
159. Lochau M, Oster S, Goltz U, Schürr A (2012) Model-based pairwise testing for feature interaction coverage in software product line engineering. SQJ 20(3-4):567–604, DOI 10.1007/s11219-011-9165-4
160. Lochau M, Schaefer I, Kamischke J, Lity S (2012) Incremental model-based testing of delta-oriented software product lines. In: TAP, Springer, pp 67–82, DOI 10.1007/978-3-642-30473-6_7
161. Lochau M, Bürdek J, Hölzle S, Schürr A (2017) Specification and automated validation of staged reconfiguration processes for dynamic software product lines. SOSYM 16(1):125–152, DOI 10.1007/s10270-015-0470-4
162. Lopez-Herrejon R, Montalvillo-Mendizabal L, Egyed A (2011) From requirements to features: An exploratory study of feature-oriented refactoring. In: SPLC, pp 181–190, DOI 10.1109/SPLC.2011.52
163. Lopez-Herrejon R, Linsbauer L, Galindo J, Parejo J, Benavides D, Segura S, Egyed A (2015) An assessment of search-based techniques for reverse engineering feature models. JSS 103:353–369, DOI 10.1016/j.jss.2014.10.037
164. Lopez-Herrejon R, Ferrer J, Chicano F, Egyed A, Alba E (2016) Evolutionary computation for software product line testing: An overview and open challenges. SCI 617:59–87, DOI 10.1007/978-3-319-25964-2_4
165. Lopez-Herrejon RE, Egyed A (2012) Towards fixing inconsistencies in models with variability. In: VAMOS, ACM, pp 93–100, DOI 10.1145/2110147.2110158
166. Lopez-Herrejon RE, Chicano F, Ferrer J, Egyed A, Alba E (2013) Multi-objective optimal test suite computation for software product line pairwise testing. In: ICSM, IEEE, pp 404–407, DOI 10.1109/ICSM.2013.58
167. Lotufo R, She S, Berger T, Czarnecki K, Wąsowski A (2010) Evolution of the linux kernel variability model. In: SPLC, Springer, pp 136–150, DOI 10.1002/smr.1595
168. Markiegi U, Arrieta A, Sagardui G, Etxeberria L (2017) Search-based product line fault detection allocating test cases iteratively. In: SPLC, vol 1, pp 123–132, DOI 10.1145/3106195.3106210
169. Mauro J, Nieke M, Seidl C, Yu I (2016) Context aware reconfiguration in software product lines. In: VAMOS, vol 27-29-January-2016, pp 41–48, DOI 10.1145/2866614.2866620
170. Mazo R, Grünbacher P, Heider W, Rabiser R, Salinesi C, Diaz D (2011) Using constraint programming to verify dopler variability models. In: VAMOS, ACM, pp 97–103, DOI 10.1145/1944892.1944904
171. Mazo R, Salinesi C, Diaz D, Djebbi O, Lora-Michiels A (2012) Constraints: The heart of domain and application engineering in the product lines engineering strategy. IJISMD 3(2):33–68, DOI 10.4018/jismd.2012040102

172. Meinicke J, Thüm T, Schröter R, Krieter S, Benduhn F, Saake G, Leich T (2016) Featureide: taming the preprocessor wilderness. In: ICSE, IEEE, pp 629–632, DOI 10.1145/2889160.2889175
173. Mendonca M, Cowan D (2010) Decision-making coordination and efficient reasoning techniques for feature-based configuration. SCP 75(5):311–332, DOI 10.1016/j.scico.2009.12.004
174. Merschen D, Polzer A, Botterweck G, Kowalewski S (2011) Experiences of applying model-based analysis to support the development of automotive software product lines. In: VAMOS, ACM, pp 141–150, DOI 10.1145/1944892.1944910
175. Michel R, Classen A, Hubaux A, Boucher Q (2011) A formal semantics for feature cardinalities in feature diagrams. In: VAMOS, ACM, pp 82–89, DOI 10.1145/1944892.1944902
176. Modrak V, Soltysova Z, Modrak J, Behunova A (2017) Reducing impact of negative complexity on sustainability of mass customization. Sustainability 9(11), DOI 10.3390/su9112014
177. Mohalik S, Ramesh S, Millo JV, Krishna SN, Narwane GK (2012) Tracing spls precisely and efficiently. In: SPLC, ACM, pp 186–195, DOI 10.1145/2362536.2362562
178. Murguzur A, De Carlos X, Trujillo S, Sagardui G (2014) Context-aware staged configuration of process variants@ runtime. In: CAISE, Springer, pp 241–255, DOI 10.1007/978-3-319-07881-6_17
179. Mussbacher G, Araújo J, Moreira A, Amyot D (2012) Aourn-based modeling and analysis of software product lines. SQJ 20(3-4):645–687, DOI 10.1007/s11219-011-9153-8
180. Nadi S, Berger T, Kästner C, Czarnecki K (2014) Mining configuration constraints: Static analyses and empirical results. In: ICSE, ACM, pp 140–151, DOI 10.1145/2568225.2568283
181. Nadi S, Berger T, Kästner C, Czarnecki K (2015) Where do configuration constraints stem from? an extraction approach and an empirical study. TSE 41(8):820–841, DOI 10.1109/TSE.2015.2415793
182. Narwane G, Galindo J, Krishna S, Benavides D, Millo JV, Ramesh S (2016) Traceability analyses between features and assets in software product lines. Entropy 18(8), DOI 10.3390/e18080269
183. Nešić D, Nyberg M (2016) Multi-view modeling and automated analysis of product line variability in systems engineering. In: SPLC, vol 16-23-September-2016, pp 287–296, DOI 10.1145/2934466.2946044
184. Novak M, Magdalenić I, Radošević D (2016) Common metamodel of component diagram and feature diagram in generative programming. JCS 12(10):517–526, DOI 10.3844/jcssp.2016.517.526
185. Ochoa L, Pereira J, González-Rojas O, Castro H, Saake G (2017) A survey on scalability and performance concerns in extended product lines configuration. In: VAMOS, pp 5–12, DOI 10.1145/3023956.3023959
186. Oster S, Markert F, Ritter P (2010) Automated incremental pairwise testing of software product lines. In: SPLC, vol 6287 LNCS, pp 196–210, DOI 10.1007/978-3-642-15579-6_14
187. Oster S, Zorcic I, Markert F, Lochau M (2011) Moso-polite: tool support for pairwise and model-based software product line testing. In: VAMOS, ACM, pp 79–82, DOI 10.1145/1944892.1944901
188. Parejo J, Sánchez A, Segura S, Ruiz-Cortés A, Lopez-Herrejon R, Egyed A (2016) Multi-objective test case prioritization in highly configurable systems: A case study. JSS 122:287–310, DOI 10.1016/j.jss.2016.09.045
189. Pascual G, Lopez-Herrejon R, Pinto M, Fuentes L, Egyed A (2015) Applying multiobjective evolutionary algorithms to dynamic software product lines for reconfiguring mobile applications. JSS 103:392–411, DOI 10.1016/j.jss.2014.12.041
190. Pascual GG, Pinto M, Fuentes L (2013) Run-time adaptation of mobile applications using genetic algorithms. In: ICSE, IEEE Press, pp 73–82, DOI 10.1109/seams.2013.6595494
191. Paškevičius P, Damaševičius R, Karčiauskas E, Marcinkevičius R (2012) Automatic extraction of features and generation of feature models from java programs. ITC 41(4):376–384, DOI 10.5755/j01.itc.41.4.1108
192. Paskevicius P, Damasevicius R, Štuikys V (2012) Change impact analysis of feature models. JKSU 319 CCIS:108–122, DOI 10.1007/978-3-642-33308-8_10
193. Passos L, Czarnecki K, Apel S, Wąsowski A, Kästner C, Guo J (2013) Feature-oriented software evolution. In: VAMOS, ACM, p 17, DOI 10.1145/2430502.2430526
194. Pereira J, Constantino K, Figueiredo E, Saake G (2016) Quantitative and qualitative empirical analysis of three feature modeling tools. CCIS 703:66–88, DOI 10.1007/978-3-319-56390-9_4
195. Perez-Morago H, Heradio R, Fernandez-Amoros D, Bean R, Cerrada C (2015) Efficient identification of core and dead features in variability models. ACCESS 3:2333–2340, DOI 10.1109/ACCESS.2015.2498764

196. Perrouin G, Oster S, Sen S, Klein J, Baudry B, le Traon Y (2012) Pairwise testing for software product lines: Comparison of two approaches. SQJ 20(3-4):605–643, DOI 10.1007/s11219-011-9160-9
197. Pleuss A, Botterweck G (2012) Visualization of variability and configuration options. STTT 14(5):497–510, DOI 10.1007/s10009-012-0252-z
198. Pleuss A, Botterweck G, Dhungana D (2010) Integrating automated product derivation and individual user interface design. In: VAMOS
199. Pleuss A, Rabiser R, Botterweck G (2011) Visualization techniques for application in interactive product configuration. In: SPLC, ACM, p 22, DOI 10.1145/2019136.2019161
200. Pleuss A, Botterweck G, Dhungana D, Polzer A, Kowalewski S (2012) Model-driven support for product line evolution on feature level. JSS 85(10):2261–2274, DOI 10.1016/j.jss.2011.08.008
201. Pohl R, Lauenroth K, Pohl K (2011) A performance comparison of contemporary algorithmic approaches for automated analysis operations on feature models. In: ASE, pp 313–322, DOI 10.1109/ASE.2011.6100068
202. Pohl R, Stricker V, Pohl K (2013) Measuring the structural complexity of feature models. In: ASE, IEEE, pp 454–464, DOI 10.1109/ASE.2013.6693103
203. Quinton C, Romero D, Duchien L (2013) Cardinality-based feature models with constraints: A pragmatic approach. In: SPLC, ACM, pp 162–166, DOI 10.1145/2491627.2491638
204. Quinton C, Romero D, Duchien L (2014) Automated selection and configuration of cloud environments using software product lines principles. In: CLOUD, IEEE, pp 144–151, DOI 10.1109/CLOUD.2014.29
205. Quinton C, Rabiser R, Vierhauser M, Grünbacher P, Baresi L (2015) Evolution in dynamic software product lines: Challenges and perspectives. In: SPLC, vol 20-24-July-2015, pp 126–130, DOI 10.1145/2791060.2791101
206. Rauber T, Boldt FdA (2015) Heterogeneous feature models and feature selection applied to bearing fault diagnosis. TIE DOI 10.1109/tie.2014.2327589
207. Rincón L, Giraldo GL, Mazo R, Salinesi C (2014) An ontological rule-based approach for analyzing dead and false optional features in feature models. ENTCS 302:111–132, DOI 10.1016/j.entcs.2014.01.023
208. Ripon S, Rahman M, Ferdous J, Hossain M (2016) Verification of spl feature model by using bayesian network. IndianST 9(31), DOI 10.17485/ijst/2016/v9i31/93731
209. Roos-Frantz F, Benavides D, Ruiz-Cortés A, Heuer A, Lauenroth K (2012) Quality-aware analysis in product line engineering with the orthogonal variability model. SQJ 20(3-4):519–565, DOI 10.1007/s11219-011-9156-5
210. Rosenmüller M, Siegmund N, Thüm T, Saake G (2011) Multi-dimensional variability modeling. In: VAMOS, pp 11–20, DOI 10.1145/1944892.1944894
211. Saeed M, Saleh F, Al-Insaif S, El-Attar M (2016) Empirical validating the cognitive effectiveness of a new feature diagrams visual syntax. IST 71:1–26, DOI 10.1016/j.infsof.2015.10.012
212. Sanchez A, Segura S, Ruiz-Cortes A (2014) A comparison of test case prioritization criteria for software product lines. In: ICST, pp 41–50, DOI 10.1109/ICST.2014.15
213. Sánchez AB, Segura S, Ruiz-Cortés A (2014) The drupal framework: A case study to evaluate variability testing techniques. In: VAMOS, ACM, p 11, DOI 10.1145/2556624.2556638
214. Sánchez AB, Segura S, Parejo JA, Ruiz-Cortés A (2015) Variability testing in the wild: the drupal case study. SOSYM DOI 10.1007/s10270-015-0459-z
215. Sannier N, Acher M, Baudry B (2013) From comparison matrix to variability model: The wikipedia case study. In: ASE, pp 580–585, DOI 10.1109/ASE.2013.6693116
216. Sayyad A, Ingram J, Menzies T, Ammar H (2013) Scalable product line configuration: A straw to break the camel's back. In: ASE, pp 465–474, DOI 10.1109/ASE.2013.6693104
217. Schaefer I (2010) Variability modelling for model-driven development of software product lines. In: VAMOS, vol 10, pp 85–92
218. Schmid K, Rabiser R, Grünbacher P (2011) A comparison of decision modeling approaches in product lines. In: VAMOS, pp 119–126, DOI 10.1145/1944892.1944907
219. Schnabel T, Weckesser M, Kluge R, Lochau M, Schürr A (2016) Cardygan: Tool support for cardinality-based feature models. In: VAMOS, vol 27-29-January-2016, pp 33–40, DOI 10.1145/2866614.2866619
220. Schroeter J, Cech S, Götz S, Wilke C, Aßmann U (2012) Towards modeling a variable architecture for multi-tenant saas-applications. In: VAMOS, ACM, pp 111–120, DOI 10.1145/2110147.2110160

221. Schroeter J, Mucha P, Muth M, Jugel K, Lochau M (2012) Dynamic configuration management of cloud-based applications. In: SPLC, ACM, pp 171–178, DOI 10.1145/2364412.2364441
222. Schröter R, Thüm T, Siegmund N, Saake G (2013) Automated analysis of dependent feature models. In: VAMOS, ACM, p 9, DOI 10.1145/2430502.2430515
223. Schröter R, Siegmund N, Thüm T, Saake G (2014) Feature-context interfaces: tailored programming interfaces for software product lines. In: SPLC, ACM, pp 102–111, DOI 10.1145/2648511.2648522
224. Schröter R, Krieter S, Thüm T, Benduhn F, Saake G (2016) Feature-model interfaces: The highway to compositional analyses of highly-configurable systems. In: ICSE, IEEE Computer Society, pp 667–678, DOI 10.1145/2884781.2884823
225. Schubanz M, Pleuss A, Botterweck G, Lewerentz C (2012) Modeling rationale over time to support product line evolution planning. In: VAMOS, pp 193–199, DOI 10.1145/2110147.2110169
226. Segura S, Benavides D, Ruiz-Cortés A (2011) Functional testing of feature model analysis tools: a test suite. IET 5(1):70–82, DOI 10.1049/iet-sen.2009.0096
227. Segura S, Hierons R, Benavides D, Ruiz-Cortés A (2011) Automated metamorphic testing on the analyses of feature models. IST 53(3):245–258, DOI 10.1016/j.infsof.2010.11.002
228. Segura S, Galindo JA, Benavides D, Parejo JA, Ruiz-Cortés A (2012) Betty: benchmarking and testing on the automated analysis of feature models. In: VAMOS, ACM, pp 63–71, DOI 10.1145/2110147.2110155
229. Segura S, Parejo J, Hierons R, Benavides D, Ruiz-Cortés A (2014) Automated generation of computationally hard feature models using evolutionary algorithms. ESA 41(8):3975–3992, DOI 10.1016/j.eswa.2013.12.028
230. Segura S, Durán A, Sánchez A (2015) Automated metamorphic testing of variability analysis tools. STVR DOI 10.1002/stvr.1566
231. Seidl C, Schaefer I, Assmann U (2014) Capturing variability in space and time with hyper feature models. In: VAMOS, ACM, p 6, DOI 10.1145/2556624.2556625
232. She S, Lotufo R, Berger T, Wasowski A, Czarnecki K (2010) The variability model of the linux kernel. In: VAMOS, vol 10, pp 45–51
233. She S, Lotufo R, Berger T, Wa¸sowski A, Czarnecki K (2011) Reverse engineering feature models. In: ICSE, pp 461–470, DOI 10.1145/1985793.1985856
234. Siegmund N, Kolesnikov SS, Kästner C, Apel S, Batory D, Rosenmüller M, Saake G (2012) Predicting performance via automated feature-interaction detection. In: ICSE, IEEE Press, pp 167–177, DOI 10.1109/icse.2012.6227196
235. Soltani S, Asadi M, Hatala M, Gašević D, Bagheri E (2011) Automated planning for feature model configuration based on stakeholders' business concerns. In: ASE, pp 536–539, DOI 10.1109/ASE.2011.6100118
236. Soltani S, Asadi M, Gašević D, Hatala M, Bagheri E (2012) Automated planning for feature model configuration based on functional and non-functional requirements. In: SPLC, ACM, pp 56–65, DOI 10.1145/2362536.2362548
237. Stein J, Nunes I, Cirilo E (2014) Preference-based feature model configuration with multiple stakeholders. In: SPLC, ACM, pp 132–141, DOI 10.1145/2648511.2648525
238. Strickler A, Prado Lima J, Vergilio S, Pozo A (2016) Deriving products for variability test of feature models with a hyper-heuristic approach. ASCJ 49:1232–1242, DOI 10.1016/j.asoc.2016.07.059
239. Tanhaei M, Habibi J, Mirian-Hosseinabadi SH (2016) Automating feature model refactoring: A model transformation approach. IST 80:138–157, DOI 10.1016/j.infsof.2016.08.011
240. Tawhid R, Petriu D (2011) Automatic derivation of a product performance model from a software product line model. In: SPLC, pp 80–89, DOI 10.1109/SPLC.2011.27
241. Teixeira L, Borba P, Gheyi R (2013) Safe composition of configuration knowledge-based software product lines. JSS 86(4):1038–1053, DOI 10.1016/j.jss.2012.11.006
242. Ter Beek M, Fantechi A, Gnesi S (2015) Applying the product lines paradigm to the quantitative analysis of collective adaptive systems. In: SPLC, vol 20-24-July-2015, pp 321–326, DOI 10.1145/2791060.2791100
243. Thüm T, Kästner C, Erdweg S, Siegmund N (2011) Abstract features in feature modeling. In: SPLC, pp 191–200, DOI 10.1109/SPLC.2011.53
244. Thüm T, Apel S, Kästner C, Schaefer I, Saake G (2014) A classification and survey of analysis strategies for software product lines. ACMCS 47(1), DOI 10.1145/2580950

245. Thüm T, Kästner C, Benduhn F, Meinicke J, Saake G, Leich T (2014) Featureide: An extensible framework for feature-oriented software development. SCP 79:70–85, DOI 10.1016/j.scico.2012.06.002
246. Tërnava X, Collet P (2017) Early consistency checking between specification and implementation variabilities. In: SPLC, vol 1, pp 29–38, DOI 10.1145/3106195.3106209
247. Vierhauser M, Grünbacher P, Egyed A, Rabiser R, Heider W (2010) Flexible and scalable consistency checking on product line variability models. In: ASE, ACM, pp 63–72, DOI 10.1145/1858996.1859009
248. Vogel-Heuser B, Fay A, Schaefer I, Tichy M (2015) Evolution of software in automated production systems: Challenges and research directions. JSS 110:54–84, DOI 10.1016/j.jss.2015.08.026
249. Von Rhein A, Apel S, Kästner C, Thüm T, Schaefer I (2013) The pla model: on the combination of product-line analyses. In: VAMOS, ACM, p 14, DOI 10.1145/2430502.2430522
250. Von Rhein A, Grebhahn A, Apel S, Siegmund N, Beyer D, Berger T (2015) Presence-condition simplification in highly configurable systems. In: ICSE, vol 1, pp 178–188, DOI 10.1109/ICSE.2015.39
251. Walter R, Felfernig A, Küchlin W (2017) Constraint-based and sat-based diagnosis of automotive configuration problems. JIIS 49(1):87–118, DOI 10.1007/s10844-016-0422-7
252. Wang S, Ali S, Gotlieb A (2013) Minimizing test suites in software product lines using weight-based genetic algorithms. In: GECCO, ACM, pp 1493–1500, DOI 10.1145/2463372.2463545
253. Wang S, Buchmann D, Ali S, Gotlieb A, Pradhan D, Liaaen M (2014) Multi-objective test prioritization in software product line testing: an industrial case study. In: SPLC, ACM, pp 32–41, DOI 10.1145/2648511.2648515
254. Wang S, Ali S, Gotlieb A, Liaaen M (2016) A systematic test case selection methodology for product lines: results and insights from an industrial case study. ESE 21(4):1586–1622, DOI 10.1007/s10664-014-9345-5
255. Wang S, Ali S, Gotlieb A, Liaaen M (2017) Automated product line test case selection: industrial case study and controlled experiment. SOSYM 16(2):417–441, DOI 10.1007/s10270-015-0462-4
256. Wang Sb, Ali S, Gotlieb A (2015) Cost-effective test suite minimization in product lines using search techniques. JSS 103:370–391, DOI 10.1016/j.jss.2014.08.024
257. White J, Benavides D, Schmidt D, Trinidad P, Dougherty B, Ruiz-Cortes A (2010) Automated diagnosis of feature model configurations. JSS 83(7):1094–1107, DOI 10.1016/j.jss.2010.02.017
258. White J, Galindo J, Saxena T, Dougherty B, Benavides D, Schmidt D (2014) Evolving feature model configurations in software product lines. JSS 87(1):119–136, DOI 10.1016/j.jss.2013.10.010
259. Wittern E, Zirpins C (2016) Service feature modeling: modeling and participatory ranking of service design alternatives. SOSYM 15(2):553–578, DOI 10.1007/s10270-014-0414-4
260. Xue Y, Zhong J, Tan T, Liu Y, Cai W, Chen M, Sun J (2016) Ibed: Combining ibea and de for optimal feature selection in software product line engineering. ASCJ 49:1215–1231, DOI 10.1016/j.asoc.2016.07.040
261. Yu W, Zhang W, Zhao H, Jin Z (2014) Tdl: a transformation description language from feature model to use case for automated use case derivation. In: SPLC, ACM, pp 187–196, DOI 10.1145/2648511.2648531
262. Zaid L, Kleinermann F, De Troyer O (2011) Feature assembly framework: Towards scalable and reusable feature models. In: VAMOS, pp 1–9, DOI 10.1145/1944892.1944893
263. Zhan Z, Luo W, Guo Z, Liu Y (2017) Feature selection optimization based on atomic set and genetic algorithm in software product line. AISC 686:93–100, DOI 10.1007/978-3-319-69096-4_14
264. Zhan Z, Zhan Y, Huang M, Liu Y (2017) Product configuration based on feature model. AISC 686:101–106, DOI 10.1007/978-3-319-69096-4_15
265. Zhang G, Ye H, Lin Y (2014) Quality attribute modeling and quality aware product configuration in software product lines. SQJ 22(3):365–401, DOI 10.1007/s11219-013-9197-z
266. Zhou F, Jiao J, Yang X, Lei B (2017) Augmenting feature model through customer preference mining by hybrid sentiment analysis. ESA 89:306–317, DOI 10.1016/j.eswa.2017.07.021
267. Zhu H, Wu L, Huang K, Zhou Z (2016) Research on methods for discovering and selecting cloud infrastructure services based on feature modeling. MPE 2016, DOI 10.1155/2016/8194832
268. Ziadi T, Frias L, da Silva MAA, Ziane M (2012) Feature identification from the source code of product variants. In: ECSMR, IEEE, pp 417–422, DOI 10.1109/CSMR.2012.52
269. Štuikys V, Burbaitė R, Bespalova K, Ziberkas G (2016) Model-driven processes and tools to design robot-based generative learning objects for computer science education. SCP 129:48–71, DOI 10.

1016/j.scico.2016.03.009

B Conferences and Journals

This section contains all the material used in this mapping study.

Rank	#Papers	**Conferences**
1	40	Software Product Line Conference
2	33	International Workshop on Variability Modelling of Software-intensive Systems
3	12	International Conference on Software Engineering
4	11	International Conference on Automated Software Engineering
5	9	International Conference on Model Driven Engineering Languages and Systems
6	9	International Conference on Software Reuse
7	7	International Conference on Generative Programming
8	6	International Conference in Software Testing
9	6	International Conference on Advanced Information Systems Engineering
10	5	International Conference on Fundamental Approaches to Software Engineering
11	4	Joint Meeting on Foundations of Software Engineering
12	4	International Symposium On Leveraging Applications of Formal Methods, Verification and Validation
13	4	Symposium on Applied Computing
14	3	International Symposium Search-Based Software Engineering
15	3	Annual Conference on Genetic and Evolutionary Computation
16	3	International Symposium on Software Engineering for Adaptive and Self-Managing Systems
17	3	Electronic Proceedings in Theoretical Computer Science
18	3	International Workshop on Modeling in Software Engineering
19	3	Working Conference on Reverse Engineering
20	3	Workshop on Scalable Modeling Techniques for Software Product Lines
21	3	International Conference on Evaluation of Novel Approaches to Software Engineering
22	2	International Conference on Software Language Engineering
23	2	Variability for You Workshop
24	2	International Joint Conference on Software Technologies
25	2	International Conference on Service Oriented Computing
26	2	International Conference of Software Engineering and Knowledge Engineering
27	2	Requirements Engineering Conference
28	2	Workshop on Product Line Approaches in Software Engineering
29	2	International Conference on Research Challenges in Information Science
30	2	European Conference on Software Architecture
31	2	International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
32	2	International Conference on Web Services
33	2	International Conference on Progress in Informatics and Computing
34	2	International Symposium on Software Reliability Engineering
35	2	Workshop on Configuration
36	2	International Workshop on Requirements Engineering Practices on Software Product Line Engineering
37	2	Software Testing Verification and Validation Workshop
38	2	International Conference on Conceptual Modeling
39	2	Requirements Engineering: Foundation for Software Quality
40	2	International Conference on Cloud Computing
41	2	Advances in Transdisciplinary Engineering
42	2	Software Engineering and Advanced Applications
43	2	IEEE International Conference on Software Maintenance
44	2	International Conference on Aspect Oriented Software Development
45	2	International Conference on Distributed Multimedia Systems
46	2	International Conference on Services Computing
47	2	International Conference on Software Engineering Companion
48	2	Workshop on Feature-Oriented Software Development
49	1	Modellierung
50	1	Brazilian Symposium on Software Components, Architectures and Reuse
51	1	Workshop on Formal Methods and Analysis in Software Product Line Engineering
52	1	International Conference on Industrial Informatics
53	1	International Workshop on Description Logics
54	1	International Conference on Fundamentals of Software Engineering
55	1	International SDL Forum
56	1	Brazilian Symposium on Software Engineering, SBES 2015
57	1	International Conference on Engineering Design
58	1	International Conference on Cloud Computing
59	1	European Dependable Computing Conference
60	1	International Workshop on the Verification of Model Transformation (VOLT)

61	1	International Workshop on Automated Configuration and Tailoring of Applications
62	1	International Conference on Software Engineering and Applications
63	1	International Workshop on Smalltalk Technologies
64	1	International Symposium on Software Reliability Engineering Workshops
65	1	Engineering Interactive Computing Systems
66	1	European Conference on Software Maintenance and Reengineering
67	1	International Colloquium on Theoretical Aspects of Computing
68	1	International Conference on Information and Communication Systems
69	1	Information Reuse and Integration
70	1	International Conference on Information, Communication and Computing Technology
71	1	International Conference on Formal Methods for Components and Objects
72	1	International Semantic Web Conference
73	1	International Working Conference on Source Code Analysis and Manipulation
74	1	International Conference on Computational Science and Engineering
75	1	FME Workshop on Formal Methods in Software Engineering
76	1	International Workshop on Variability and Complexity in Software Design
77	1	International conference on Principles and practice of constraint programming
78	1	Workshop on Recommendation Systems for Software Engineering
79	1	Central and East European Conference on Software Engineering Techniques
80	1	European Conference on Modelling Foundations and Applications
81	1	International Conference on Tests and Proofs
82	1	Modularity
83	1	Working Conference on Software Visualization
84	1	International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops
85	1	Conference on Telematics and Information Systems
86	1	International Conference of the Chilean Computer Science Society
87	1	International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software
88	1	International Conference on Model-Driven Engineering and Software Development
89	1	Workshop on Knowledge-Oriented Product Line Engineering
90	1	International Conference on Computing, Communication and Automation
91	1	Conference on Programming Language Design and Implementation
92	1	International workshop on Early aspects
93	1	European Conference on the Applications of Evolutionary Computation
94	1	International Conference on Program Comprehension
95	1	International Conference on Software and Systems Process
96	1	Workshop on software product line analysis tools
97	1	Latin American Computing Conference
98	1	The International Symposium on Software Testing and Analysis
99	1	First International Workshop on Multi Product Line Engineering
100	1	International Symposium on Formal Methods
101	1	International Conference on Software Composition
102	1	International Conference on Engineering of Complex Computer Systems
103	1	International Conference on Integrated Formal Methods
104	1	Conference on Cloud Computing and Services Science
105	1	International conference on Object oriented programming systems languages and applications
106	1	International Workshop on Search-Based Software Testing
107	1	International Conference on Software Engineering and Formal Methods
108	1	International Conference on Software Analysis, Evolution, and Reengineering
109	1	SCArVeS: Services, Clouds, and Alternative Design Strategies for Variant-Rich Software Systems
110	1	Colombian Computing Conference
111	1	International Symposium on Theoretical Aspects of Software Engineering
112	1	International Workshop on Formal Aspects of Component Software
113	1	International Conference on Industrial Engineering and Engineering Management
114	1	International Conference on Bio-Inspired Information and Communications Technologies
115	1	International Workshop on Dynamic Software Product Lines
116	1	International Conference on Enterprise Information Systems
117	1	Combining Modelling and Search-Based Software Engineering
118	1	International Conference on Artificial Intelligence and Statistics
119	1	International Conference on Concept Lattices and Their Applications
120	1	Proceedings of the on Future of Software Engineering

Table 10: Conferences and number of papers from the survey

Rank	#Papers	**Journals**
1	13	Journal of Systems and Software
2	11	International Journal on Software and Systems Modeling,Software and Systems Modeling
3	9	Software Quality Journal
4	7	IEEE Transactions on Software Engineering
5	6	Information and Software Technology
6	6	Science of Computer Programming
7	5	International Journal on Software Tools for Technology Transfer
8	4	Expert Systems with Applications
9	3	Computer
10	3	Empirical Software Engineering
11	2	International Journal of Software Engineering and Knowledge Engineering
12	2	Applied Soft Computing Journal
13	2	Communications in Computer and Information Science
14	2	Journal of Universal Computer Science
15	2	Indian Journal of Science and Technology
16	2	Advances in Intelligent Systems and Computing
17	2	Requirements Engineering
18	2	Journal of Logical and Algebraic Methods in Programming
19	1	Mathematical Problems in Engineering
20	1	MDPI Entropy
21	1	Electronic Notes in Theoretical Computer Science
22	1	Journal of Intelligent Information Systems
23	1	Journal of Biomedical Informatics
24	1	Computer Languages, Systems and Structures
25	1	MDPI Open Access Journal Sustainability
26	1	Journal of Software: Evolution and Process
27	1	Scientific Annals of Computer Science
28	1	Artificial Intelligence for Engineering Design, Analysis and Manufacturing
29	1	Transactions on Software Engineering and Methodology
30	1	International Journal of Information System Modeling and Design
31	1	ACM Computing Surveys
32	1	Information Systems
33	1	Computational Ecology and Software
34	1	Frontiers in Artificial Intelligence and Applications
35	1	Frontiers of Computer Science
36	1	Electronic Communication of the European Association of Software Science and Technology
37	1	IEEE Computational Intelligence Magazine
38	1	IET Software
39	1	Journal of Computer Science
40	1	Computer Networks
41	1	Journal of King Saud University
42	1	ISRN Software Engineering
43	1	China - Journal of Software
44	1	Automated Software Engineering
45	1	Future Generation Computer Systems
46	1	Cluster Computing
47	1	IEEE Transactions on Automation Science and Engineering
48	1	Information Systems Frontiers
49	1	IFAC Paper Online
50	1	IEEE Access
51	1	Advanced Information Systems Engineering
52	1	European Journal of Operational Research
53	1	Formal Aspects of Computing
54	1	IEEE Transactions on Industrial Electronics
55	1	Transactions on Autonomous and Adaptive Systems
56	1	Studies in Computational Intelligence
57	1	Information technology and control
58	1	Software testing, verification and reliability

Table 11: Journals and number of papers from the survey