applied sciences

MDPI

Article

Reasoning Algorithms on Feature Modeling—A Systematic Mapping Study

Samuel Sepúlveda * and **Ania Cravero**

Departamento de Ciencias de la Computación e Informática, Centro de Estudios en Ingeniería de Software, Universidad de La Frontera, Temuco 4811230, Chile; ania.cravero@ufrontera.cl
* Correspondence: samuel.sepulveda@ufrontera.cl

Abstract: Context: Software product lines (SPLs) have reached a considerable level of adoption in the software industry. The most commonly used models for managing the variability of SPLs are feature models (FMs). The analysis of FMs is an error-prone, tedious task, and it is not feasible to accomplish this task manually with large-scale FMs. In recent years, much effort has been devoted to developing reasoning algorithms for FMs. Aim: To synthesize the evidence on the use of reasoning algorithms for feature modeling. Method: We conducted a systematic mapping study, including six research questions. This study included 66 papers published from 2010 to 2020. Results: We found that most algorithms were used in the domain stage (70%). The most commonly used technologies were transformations (18%). As for the origins of the proposals, they were mainly rooted in academia (76%). The FODA model continued to be the most frequently used representation for feature modeling (70%). A large majority of the papers presented some empirical validation process (90%). Conclusion: We were able to respond to the RQs. The FODA model is consolidated as a reference within SPLs to manage variability. Responses to RQ2 and RQ6 require further review.

Keywords: reasoning algorithms; automated analysis; feature modeling; software product lines; systematic mapping

Citation: Sepúlveda, S.; Cravero, A. Reasoning Algorithms on Feature Modeling—A Systematic Mapping Study. *Appl. Sci.* **2022**, *12*, 5563. https://doi.org/10.3390/app12115563

Academic Editor: Luis Javier Garcia Villalba

Received: 8 February 2022
Accepted: 25 May 2022
Published: 30 May 2022

1. Introduction

Today software customers are demanding new and better products and services, which has forced the software industry to devise new approaches that increase the productivity of their processes and the quality of their products. For some time now, researchers in the field of *software engineering* have studied various alternatives for software development; among these is the software product lines (SPLs) approach. There are several differences between traditional single-system development and SPLs. The main difference is a paradigm change from individual software systems to a product line (also known as a family product) approach. According to [1], adopting this new paradigm implies a change in strategy: from the ad hoc next-contract vision to a strategic view of a field of business.

SPLs are defined as a set of characteristics to satisfy the specific needs of a particular market segment [2]. The use of SPLs as a software development methodology provides a set of benefits, including a reduction in development times and increases in productivity, among others [2,3]. Furthermore, Van der Linden et al. argue that these improvements significantly affect the development process, particularly in relation to costs and time to market, but it is at the level of software reuse that it is possible to achieve unprecedented levels of reuse [1].

SPL development is based on a common set of fundamental elements: an architecture, a collection of software components, and a set of products [4]. One of the key concepts is variability, which provides SPLs with the flexibility required for product diversification and differentiation [5]. Variability refers to combining the different functionalities that each component gives to the LPS, and this can be represented graphically using variability models. Nowadays, there are various methods of representing the variability in an LPS; however, feature models (FMs) are the most widely used method [6].

SPLs and variability are currently a fully active research area, showing their validity and relevance in the software engineering community. This relevance can be seen in a series of tertiary studies, i.e., studies that identify how variability is modeled [7]. Additionally, we can see how SPL engineering and variability management has been applied along with the Internet of Things [8].

Building and maintaining an FM is considered an expensive and error-prone task [9,10]. Moreover, the evolution and changes in the FM can introduce redundancy into the models, leading to the information being modeled in a contradictory way, resulting in modeling errors [11,12].

Throughout the SPL framework, starting from the creation of the FMs, the validation of the SPLs, the derivation of products, and even the modification or extension of the product family, it is essential to consult the FMs to obtain relevant information on the processes mentioned above. However, providing answers to these queries is not trivial because, given the structure of FMs, this process requires algorithms that support a set of rules and constraints that tend to be more complex depending on the model's size. As we can see, the information we can obtain from FMs is extensive. This process of securing information is known as the *automated analysis* of *feature models*(AAFM), and it has been identified as one of the most critical areas in the SPL community [13]. According to [14], it is possible to propose ad hoc algorithms to perform AAFM.

In this study we aimed to account for and synthesize the current state of the reported scientific literature about the use of reasoning algorithms in FMs, as implemented in the stages and activities that comprise the SPL framework. We conducted a systematic mapping study (SMS) to identify a set of relevant papers that could help us answer the research questions (RQs). In this SMS, we aimed to collect the evidence present in the literature from the last ten years. We established some parametersin respect to the application domain, underlying model, origin, and degree of empirical validation of the proposals, in order to collect the most significant amount of data from the proposals found, and for the analysis, synthesis, and subsequent publication of the results generated. This paper may interest researchers and practitioners looking for an updated view of the use of reasoning algorithms or automated analysis on FMs and the gaps in research areas. The results of this study could provide a foundation for developing a proposal for reasoning algorithms based on model-driven development approaches.

The remainder ot this paper is structured as follows. Section 2 presents the background. Section 3 presents some related studies. Section 4 presents the methodology. Sections 5 and 6 present the results and discussion, respectively. Finally, Section 7 presents the conclusions and future work.

2. Background

This section provides information on the definitions and characteristics of SPLs, FMs, and reasoning algorithms.

2.1. Software Product Lines

SPLs are defined as a set of similar software products created from reusable artifacts in the context of a specific application domain [15]. SPLs are developed in two stages: *domain engineering* and *application engineering* [16]. In the *domain engineering* stage, the common and variant elements are described. The *application engineering* stage is where the individual products of the SPL are built by reusing domain devices and exploiting the variability of the SPLs. Figure 1 shows the SPL framework, including both stages and their interactions.

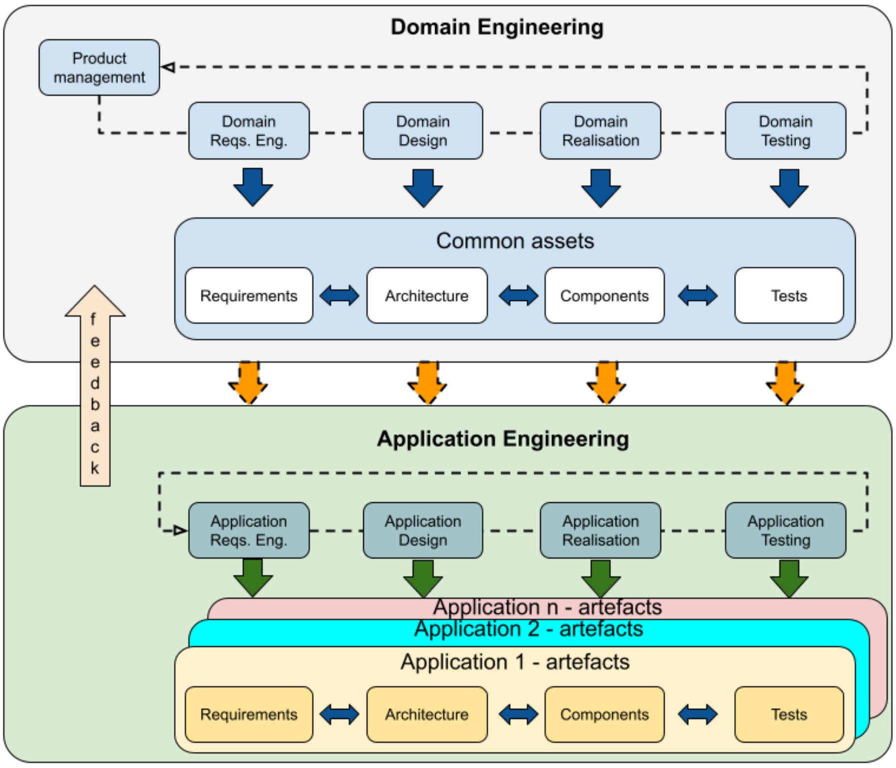


Figure 1. SPL framework, stages and their relationships.

Software development based on SPLs has brought about benefits such as the reuse of components, increases in productivity, reductions in development times, relatively fewer major errors,improvements in product quality, and lower costs, among others [2,3,15,16]. Contrary to what one may initially believe, the successful implementation of SPLs is not a phenomenon exclusive to large development companies but is also feasible in small and medium-sized companies, as demonstrated in [17].

2.2. Variability

One of the main concepts in SPLs development is *variability*, which gives SPLs the flexibility required to diversify and differentiate products [18].

Variability is introduced by defining reusable artifacts, such as architectures or components. These artifacts are included in the definition of a product family, depending on their inclusion or exclusion in each final product, giving rise to particular products [19]. Several authors have proposed models to manage the variability of SPLs. Most of these proposals are based on the FODA model [20]. This FODA model consists of characteristics and relationships that are graphically extended in the form of a tree.

For example, a software product must be able to adapt to the needs of each client or allow options for some specific configuration so that the products can reach different market segments [3]. In *domain engineering*, it is common to describe SPL and manage its *variability* with the aid of FMs [6].

2.3. Feature Models

The origin of FMs can be traced to the FODA method [20]. This model is still present but with slight variations and adaptions in some SPL methods based on visual representations of the product's features.

The structure of an FM is a type of tree of which the root node represents the product family, and the features are organized throughout the tree. These features can be assembled to give rise to particular software products [21]. FMs have been a relevant topic for SPLs in recent years, showing the best evolution behavior in terms of the number of published papers and references [13].

To illustrate the concepts present in an FM, consider the following scenario. A mobile phone must have the possibility of making a call and have a screen, but not all mobile phones must have a GPS. Furthermore, some of these features can depend on others for their inclusion or exclusion. For example, if a mobile phone has a GPS, it cannot have a basic screen. See details in Figure 2.

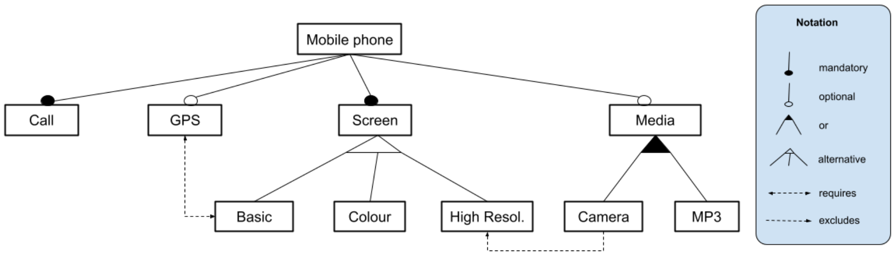


Figure 2. Example of an FM for a Mobile Phone SPL.

2.4. Automated Analysis of FMs–Reasoning Algorithms

Automatic analysis of FMs (AAFM) extracts information from such models using automated mechanisms [22]. This information includes verifying whether a given product represents a valid combination of features or checking the similarity between FMs. The analysis of FMs is an error-prone, tedious task, and it is not feasible to achieve this task manually with large-scale FMs. AAFM is an active area of research and is gaining importance for both practitioners and researchers in the SPL community [23].

Benavides et al. mention that AAFM can be defined as the computer-assisted extraction of information from FMs [24]. Different proposals for extracting this information have been made, based on specific algorithms or binary decision diagrams, such as BDD, SAT, and CSP [25]. Table 1 presents a summary of these proposals.

Table 1. Proposals to extract information from FMs.

Proposal	Characteristics
Constraint Satisfaction Problem (CSP)	This consists of a set of variables, finite domains for those variables, and a set of constraints that restrict the values of the variables. It can perform most of the operations currently identified in feature models [23].
Boolean Satisfiability Problem (SAT)	This consists of a set of Boolean variables connected by logical operators. The SAT problem consists of deciding whether a given propositional formula satisfies whether logical values can be assigned to its variables so that the formula is true [26].
Binary Decision Diagrams (BDD)	A data structure is used to represent a boolean function. A BDD is an acyclic, directed, rooted graph composed of a group of decision nodes and two terminal nodes called 0-terminal and 1-terminal. Each node of the graph represents a variable in a Boolean function and has two child nodes representing an assignment of the variable to 0 and 1 [27].

The process of extracting information from an FM starts with the translation of the features and relationships encoded in the FMs and any additional information into a knowledge base described in a logical paradigm [28]. Subsequently, queries to the knowledge base can be performed using solvers.These operations are performed automatically using different approaches. Most of them translate FMs into specific logical paradigms, such as propositional logic, constraint programming, and description logic [14].

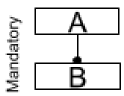
A classification of different proposals related to automatic or semi-automatic FM construction is presented in [29]. The authors of that study conceptualized an analysis framework for work in the field of automated FM construction. The framework considers four dimensions (proposal, input, tasks, and output) and fifteen sub-dimensions.

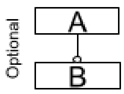
Next, in Table 2, we present specific examples describing analysis operations on FMs and possible practical applications of this automation. Table 3 summarizes two relationships of FMs (mandatory and optional), depicting their representation using *propositional logic (PL)* and *constraint programming (CP)*, and examples of their application using the Mobile Phone SPL example presented in Figure 2. A detailed compilation of operations, formal definitions, and solution proposals can be seen in [14,23,28]. Finally, a synthesis of FM data extraction process can be seen in Figure 3.

Table 2. Examples of analysis operations on FMs.

Operation	Definition	Possible Applications
Void feature model	This operation takes an FM as the input and returns a value indicating whether such model is void.	Automating this operation helps to debug large-scale FMs.
Valid product	This operation takes an FM and a product as the input and returns a value that determines whether the product belongs to the set of products represented by the FM or not.	This operation may help ti determine whether a given product is available in an SPL.
All products	This operation takes an FM as the input and returns all the products represented by the model.	This operation may help to identify new valid requirement combinations not considered in the initial scope of the SPL.

Table 3. Summary of representations of relationships of FMs.

Relationship	PL Mapping	CP Mapping	Examples
	A <–> B	A = B	**PL-Mapping:** Mobile Phone <–> Calls Mobile Phone <–> Screen **CP-Mapping:** Mobile Phone = Calls Mobile Phone = Screen

	A –> B	if (A = 0)
		B = 0
			PL-Mapping:
			GPS –> Mobile Phone
			Media –> Mobile Phone
			CP-Mapping:
			if (Mobile Phone = 0) GPS = 0
			if (Mobile Phone = 0) Media = 0

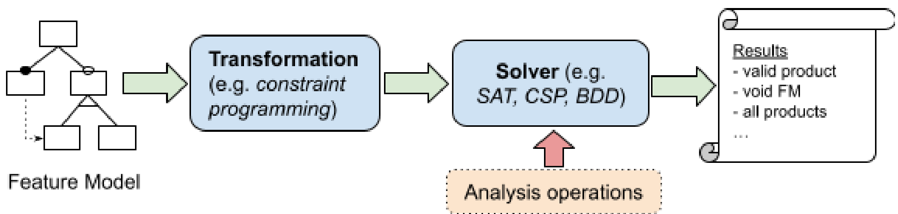


Figure 3. Synthesis of FM data extraction process [30].

3. Related Work

To date and as far as we could ascertain, there have been no other secondary studies dedicated to reviewing the use of reasoning algorithms and FMs. Therefore, this section presents a summary of three proposals related to automated feature modeling analysis [14,28,31].

Furthermore, we considered an extra systematic review dealing with FM defects and their improvement [32]. The overlapping RQs for the related work are shown in Table 4 (✓symbol).

Table 4. RQs tackled in previous related work.

Ref.	RQ1	RQ2	RQ3	RQ4	RQ5	RQ6
[14]		✓			✓	
[28]				✓		
[31]		✓				
[32]		✓				✓

Benavides et al. [14] presented a systematic literature review on the automated analysis of FMs. Their review included 53 papers from 1990 to 2010. The review presented a catalog with 30 analysis operations identified in the literature. It also provided information about the tools used to perform the analyses and the results. The authors concluded that the automated analysis of FMs was maturing, with an increasing number of contributions, operations, tools, and empirical works. They also identified some challenges for future research.

Galindo et al. [28] present an overview of the evolution of the automated analysis of FMs. The authors performed a systematic mapping study considering 423 papers from 2010 to 2017. The authors found six different facets of variability with the automated analysis of FMs having been applied to product configuration and derivation, testing and evolution, reverse engineering, multi-model variability analysis, variability modeling, and variability-intensive systems. They also confirmed the lack of industrial evidence in most of the cases. Finally, they suggested some synergies with other areas that could motivate further research in the future.

Benavides [31] presented an overview of the history and the importance of variability modeling and analysis. The author tracesd 30 years of history, from the models proposed by Kang in 1990 [20] to the present day. The work examined the beginnings, evolution, and maturity of variability modeling. This overview included FMs, their formal modeling, and the automatic analysis of variability models.

Bhushan et al. [32] presented a summary and critical research issues related to FM defects in SPL. The authors performed a systematic literature review, considering 77 papers from 1990 to 2015. According to the authors, the paper considered five main contributions. The first was a classification of FM defects in the form of a typology. Then, they presented the identification of various types of FM defects and their explanations. Third, the description, identification, explanation, and formalization of a possible set of cases of FM defects and their sub-case(s) were carried out. Fourth, corrective explanations were proposed to fix defects. Finally, the authors provided some insights on their classification and review and inferred some future research directions.

For further details of these related works, including their goals, RQs, and results, see Appendix A.

Although the studies of Segura et al. [33] and Pohl et al. [34] do not represent secondary studies, they have been considered in this section because they present proposals that study and compare alternatives related to AAFM.

Segura et al. [33] presented BeTTy, a framework for benchmarking and testing in the analysis of FMs. This framework enables the automated detection of faults in feature model analysis tools. It also supports the generation of motivating test data to evaluate the performance of analysis tools in both average and pessimistic cases.

Pohl et al. [34] presented a performance comparison regarding nine contemporary high-performance solvers, three for each base problem structure (BDD, CSP, and SAT). Four operations on 90 feature models were run on each solver. The experiment results indicated that different solvers can display superior performance on specific models or perform specific operations, with the BDD solvers producing the best results in most situations.

In addition, to complement this analysis, a summary of four tools (S.P.L.O.T., FaMiLiaR, FaMa, FeatureIDE), that support AAFM is presented in Appendix B.

Finally, we can state that our SMS shares a thematic context with the previous papers in relation to SPLs, variability modeling, and FM analysis. However, this SMS is oriented towards reasoning algorithms applied to FMs. Furthermore, it considers aspects such as the application domain, underlying model, origin, and degree of empirical validation of the proposals.

4. Methodology

This section describes the definition of protocols required to conduct an SMS according to the guidelines defined by Petersen [35].

Based on the studies of Kitchenham (2010) and Petersen (2015), we can state that an SMS aims to identify all research related to a specific topic, and it can be seen as a method to classify and structure a field of interest in *software engineering* [35,36].

Next, in Section 4.1, we define the SMS protocol. Then, in Section 4.2 we describe the study selection and data extraction processes. Finally, in Section 4.3, we provide a brief description of the tool support used in our SMS. For a better understanding of the whole process, we provide Figure 4.

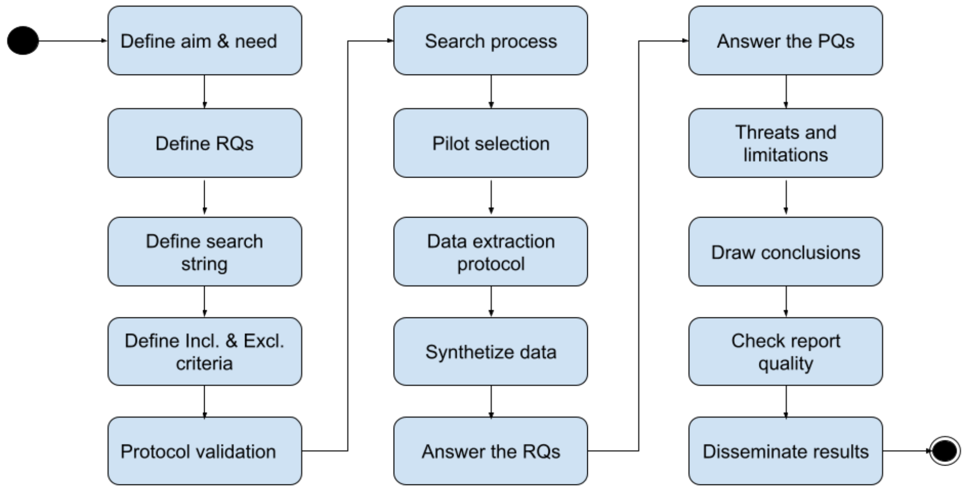


Figure 4. SMS Process and stages.

4.1. Protocol Definition

This section presents the main steps performed in the protocol definition. The first step consisted in determining the aim and need for the SMS (Section 4.1.1). Then, we defined the set of RQs that drive this SMS (Section 4.1.2). Based on these RQs, we defined the search string used to select the primary studies (Section 4.1.5). Furthermore, based on the RQs, we defined a set of inclusion/exclusion criteria (Section 4.1.6). Finally, we performed a validation of the defined protocol (Section 4.1.7).

4.1.1. Aim and Need

In this SMS we aimed to collect reasoning algorithm proposals for FMs, present in the literature from the past ten years. We established some parameters regarding the application domain, underlying model, origin, and degree of empirical validation of the proposals.

We see this study as the foundation to generate a proposal for reasoning algorithms based on model-driven development approaches. To accomplish this, it is necessary to understand in detail the proposals that exist today within the area. This analysis will allow us to:

- Understand the requirements to create algorithms of this nature.
- Understand what technologies, tools, approaches, etc., are used for building these algorithms, as well as the justifications for using them in each case.
- Avoid activities or processes that have already been carried out by other authors.

Moreover, the last study that collected this information is ten years old [14]. A similar and more recent state-of-the-art report has emerged, although it has a different focus than the one we wish to address in this paper [28,37]. In particular, [37] is an extension of [14] and seeks to answer questions related to FM reasoning algorithms that can be applied in *configuration modeling*. On the other hand, Galindo presented results focused mostly on bibliometrics [28].

The importance of this study lies in the systematic gathering and reporting of an updated view of the state of feature modeling tools for SPLs in terms of their origin, development process, underlying modeling notations, and empirical validation. We also included other aspects in this study, namely, the origins of the papers, as well their context of application, year of publication, publisher, and target audience.

Providing a clear picture of all these characteristics may help professionals by diminishing the risks associated with choosing a tool, and it may also contribute to the field by providing a framework against which new tools may be compared. Furthermore, we aim to foster a discussion among the community about the qualities that feature modeling tools for SPLs should have to facilitate the creation of high-quality specifications.

4.1.2. Research Questions

We define a context for the RQs guiding this study [38]. This context arises from a general question. Will it be possible to build a set of reasoning algorithms based on a modeling language composed by a meta-model for FMs?

To answer this question, it is necessary to have knowledge of the existing proposals in the literature related to reasoning algorithms. This information will allow us to understand the technologies and the context in which these algorithms have been used. The general question was therefore broken down into six questions related to origin, validation level, and technologies, among other issues, for the selected papers. Table 5 shows the RQs, the aim that the RQs seek to clarify, and a possible classification schema.

Table 5. RQs and details.

ID	RQs	Aim and Classification Schema
RQ1	In which SPL stage are these algorithms used?	To highlight the area where the algorithms are applied: domain engineering, application engineering
RQ2	What type of technologies do algorithms mainly use?	To understand which technologies the algorithms are most often based on: meta-model, UML, OCL, transformations, solver, other.
RQ3	What is the origin of the proposal?	To identify the origins of the papers: academia, industry, or jointly.
RQ4	What is the level of validation?	To gain insights into the maturity level of the research based on the Wieringa research taxonomy [39].
RQ5	What kind of FM does the algorithm work on?	To know what type of FMs are the most used: FODA FM, extended FM, multiplicity, orthogonal model, multi FM, complex FM, others.
RQ6	What problems does the algorithm solve?	To highlight which problems have more solutions and which do not: null FMs, valid product, valid partial configuration, etc.

4.1.3. Publication Questions

Additionally to RQs, publication questions (PQs) have been included to complement the gathered information and to characterize the bibliographic and demographic space. These PQs include the type of venue and publisher of each paper and the number of papers per year. The details are shown in Table 6.

Table 6. PQs and details.

ID	RQs	Aim & Classification Schema
PQ1	Where was the paper published?	To help researchers know which journals or conferences are most interested in each topic.
PQ2	What was the year of publication of each paper?	To highlight how the algorithms have progressed through the years from 2010 to 2020.

4.1.4. Data Sources

We considered the data sources detailed in Table 7. These sources are recognized as being among the most relevant in the SE community [38,40].

Table 7. Data sources.

Source	URL (Last Access)
ACM Digital Library	https://dl.acm.org (accessed on 5 January 2022)
IEEE Xplore	https://ieeexplore.ieee.org/Xplore/home.jsp (accessed on 10 December 2021)
Springer Link	https://www.springer.com (accessed on 10 November 2021)
Wiley Inter-Science	https://www.onlinelibrary.wiley.com (accessed on 25 November 2021)

4.1.5. Search String

According to Kitchenham and Charters [38], the search string was constructed as follows:

- From the RQs, we obtained keywords.
- For every keyword, we considered a set of synonyms.
- We applied the Population-Intervention-Comparison-Outcomes-Context (PICOC [41]) criteria.

The keywords and synonyms were as follows:

- feature model/modeling/diagram, variability model/modeling
- software product family/lines
- reasoning/reasoner, automated support/verification, computer aided
- algorithm, solver, reasoner
- model checking/validation/verification/querying

Next, we detail the application of the PICOC criteria, considering the guidelines of Kitchenham and Charters [38].

- A population in the SE community is defined as a specific role, category of software engineering, an application area, or an industry group. In our case, an application area was selected, specifically *feature modeling* in SPLs.
- Intervention is defined as a methodology, tool, technology, or procedure addressing a specific issue. In our case, a technology was selected, specifically *reasoning algorithms*.
- Comparison does not apply to our study because the RQs did not consider the comparison of gathered papers versus a common reasoning algorithm (control condition).
- Outcomes for our RQs were the origin, level of validation, type of FMs, and problems solved for each proposal.

- The context for this study includes SPLs, specifically *feature modeling*, and *reasoning and (semi-) automated algorithms*.

Then, we used the *"AND/OR"* Boolean operators to join all the terms. All different terms were joined with *AND*. All the synonyms were joined using *OR*. Figure 5 shows the final search string.

```
("feature model" OR "feature models" OR "feature modelling" OR
"feature diagram" OR "configuration model" OR "variability model" OR
"variability modeling") AND
("reasoning" OR "analysis" OR "analyses") AND
("algorithm" OR "automated" OR "computer aided") AND
("software product line" OR "software product lines" OR
"product family" OR "product families" OR "product line" OR
"product lines")
```

Figure 5. Search string.

4.1.6. Inclusion and Exclusion Criteria

To decide if a paper is relevant or not for this study, we applied a set of filters.

The first filter applied to the papers was the inclusion criteria (IC), and the remaining papers were filtered by applying the exclusion criteria (EC). Tables 8 and 9 present the definitions of IC and EC, respectively.

Table 8. Inclusion criteria definition.

ID	Criteria
IC1	Papers with more than one version—the most recent version will be included and the others will be excluded.
IC2	Works written in English.
IC3	Type of paper: • Conference proceedings • Journal article
IC4	Papers published between 2010 and 2020.
IC5	Papers of which the abstracts show the study's relationship with the automatic analysis of FMs.
IC6	General Topic: • Computer science • Software engineering

Table 9. Exclusion criteria definition.

ID	Criteria
EC1	Duplicated papers will be excluded.
EC2	The following types of papers will be excluded: • Tutorial • Short paper (4 pages or less). • Poster • Keynote • Paper in progress (incomplete). • Book • Book chapter
EC3	Papers of which the abstracts do not show the study's relationship with the automatic analysis of FMs will be excluded.
EC4	Secondary studies will be excluded. If they are relevant, they could be added as related work.
EC5	Papers that can not be accessed will not be considered.

4.1.7. Protocol Validation

The validation was performed along with the definition of each step of the protocol. This validation was carried out byone of the authors, who had extensive experience in completing secondary studies. This expert independently assessed each step of the protocol. These actions resulted in the protocol being published on the arXiv platform (https://arxiv.org/, accessed on 27 March 2022) [42]. The information presented in this paper corresponds to the final result (definition plus validation) of each step.

4.2. Primary Study Selection

We sought to compile a complete list of papers related to reasoning algorithms, FMs, and SPLs. This SMS dates back to 2010 since the previous work by Benavides was published in that year [14]. We conducted the search between August and November 2020.

The search strategy consisted of an automatic search of electronic databases, using the defined search string and selected data sources (see Figure 5 and Table 7).

4.2.1. Pilot Selection

We performed a pilot selection and extraction process to assure the protocol's reliability.

To avoid any potential bias due to a particular researcher examining each paper, we verified that the application of the IC and EC criteria was similar among the two researchers and two assistants involved in the search (inter-rater agreement). This verification was achieved by each team member individually, deciding on the IC and EC of a set of 10 papers that were randomly chosen from those retrieved in this pilot selection process. We performed a test of concordance based on the *Fleiss Kappa statistic* as a means of validation [43]. The first attempt failed (*Kappa* = 0.63).

Then, the research team carried out a set of virtual meetings to discuss the differences of opinion regarding the meaning of the content-related criteria. We rewrote the criteria accordingly. We selected another group of ten papers, and the protocol was applied independently again. This time we obtained *Kappa* = 0.81, a value that suggests that the criteria were clear enough for the research team to apply the IC and EC consistently [44].

4.2.2. Data Extraction Protocol

After validating the protocol, we launched the primary study retrieval and data extraction phase. This retrieval was carried out applying the inclusion/exclusion criteria (see Tables 8 and 9).

First, we ran the search string in the selected data sources (see Section 4.1.4). This process produced 1250 results. After eliminating duplicates (according to EC1), 1226 results were left in the list.

Then, we selected results by the type of paper (tutorials, posters, etc.), and after eliminating these (according to EC2), 1195 results remained in the list. Then, we eliminated the secondary studies (according to EC4), and 1144 results remained in the list. Then, 45 short papers were eliminated, and 1099 results remained on the list. The non-accessible papers were discarded (according to EC5), and 1064 were left on the list.

Next, the abstract of each paper was reviewed. We looked for a specific relationship with the topic of AAFM, reasoning algorithms, and related topics (according to EC3), and 991 papers were eliminated. The final list included 73 papers. Finally, after performing a detailed review and assessing the capability of each paper to answer the RQs, seven papers were eliminated, and 66 papers were finally selected. For a graphical evolution of the list of primary studies, see Figure 6. For details of selected papers, see Appendix C.

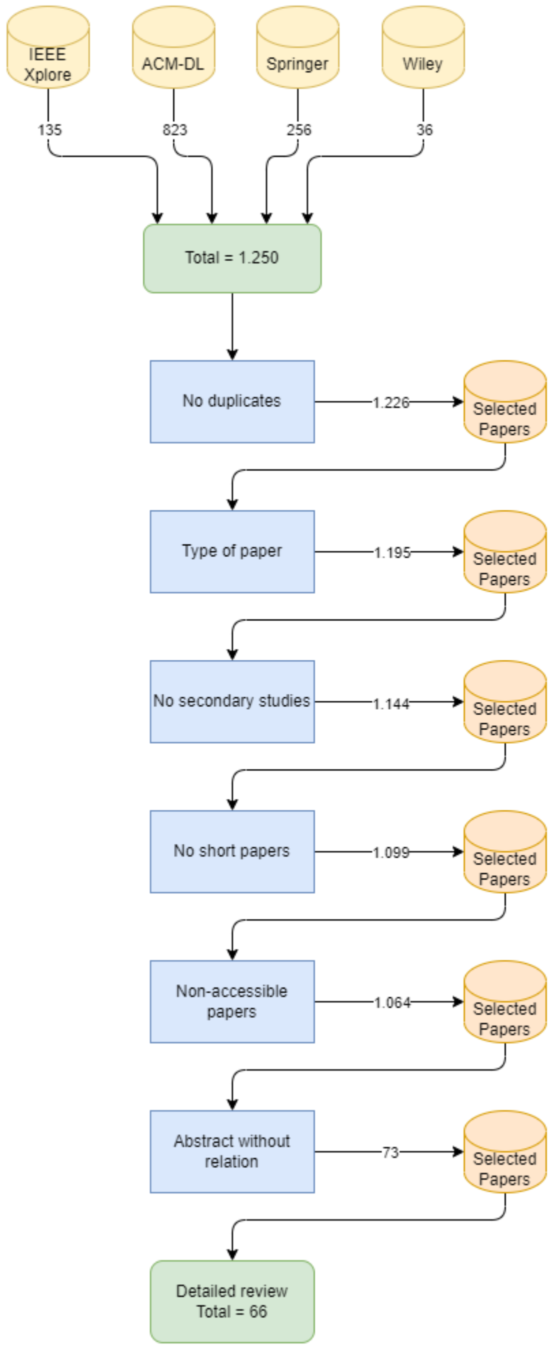


Figure 6. SMS primary study selection steps.

4.2.3. Preliminary Data Extraction and Assessment

For each of the 66 selected papers, we read them to extract relevant data to answer the RQs and PQs. The extracted data considered (i) the title, authors, and year; (ii) the type of publication (journal or conference proceedings) and the corresponding publisher; (iii) the type of experience reported; (iv) the variability model and algorithm used; and (v) whether the paper mentioned the development of tool support.

4.3. SMS Tool Support

To facilitate collaboration among the team members sharing the gathered information, the following support tools were used. Google Drive (https://drive.google.com/) and Google Sheets https://docs.google.com/spreadsheets were used for storing, finding, selecting, documenting, and analyzing the papers. Publish or Perish (https://harzing.com/resources/publish-or-perish/) and Google Scholar https://scholar.google.com/ were used for testing the search string. Overleaf (https://www.overleaf.com/) was used for editing and managing the files to create this paper. The Zoom (https://zoom.us/) and Slack (https://slack.com/) platforms were used to coordinate the research team (synchronously and asynchronously, respectively), considering the context of the COVID-19 pandemic. To create the weighted word cloud figure we used TagCrowd (https://tagcrowd.com/), and to create the Sankey diagram we used Sankeymatic (https://www.sankeymatic.com/). To find relationships between gathered data and for visualization, we used VOSviewer (https://www.vosviewer.com/, accessed on 27 March 2022).

5. Results

This section presents the answers to the RQs (Section 5.1) and PQs (Section 5.2) posed by this SMS.

5.1. Answers to RQs

5.1.1. RQ1: In Which SPL Stage Are These Algorithms Used?

To track the origins of the algorithms, we determined the SPL stages in which these algorithms were used. The origin of the algorithm's classification was a closed categorization, considering the levels:

- Domain (D): used at the domain engineering stage;
- Application (A): used at the domain engineering stage; and
- Both (D + A): used in both stages.

Forty-six papers (69.7%) reported that the algorithms were used at the domain engineering stage. Nine papers (13.6%) reported the use of algorithms in application engineering. The remaining eleven papers (16.7%) reported that the algorithms were used in both stages. See details in Figure 7. Table 10 shows the selected papers that corresponded to each of the stages.

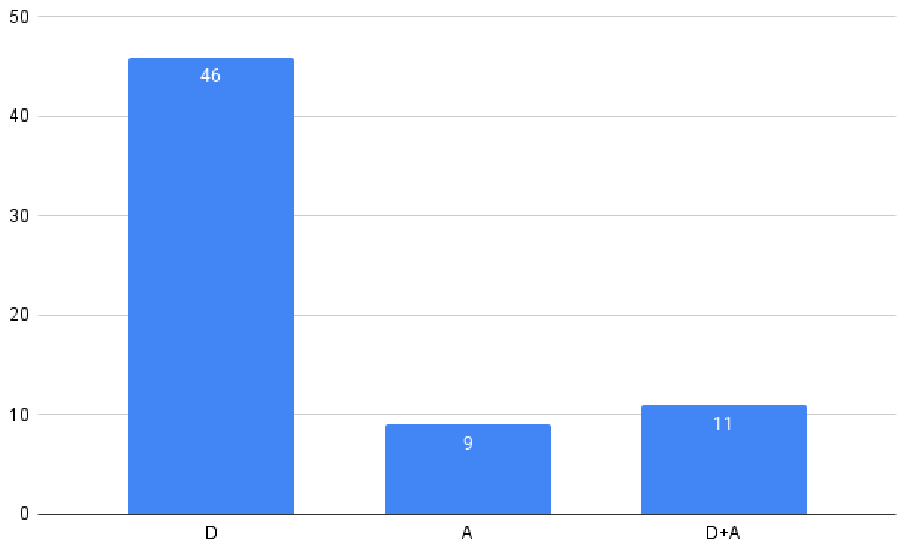


Figure 7. Stages and algorithms reported.

Table 10. Selected papers and SPL stages.

D	A	D + A
SP2, SP3, SP4, SP6, SP8, SP11, SP12, SP13, SP15, SP18, SP19, SP20, SP21, SP23, SP24, SP25, SP27, SP28, SP30, SP31, SP32, SP33, SP35, SP37, SP38, SP40, SP41, SP44, SP46, SP49, SP50, SP52, SP53, SP54, SP58, SP59, SP61, SP62, SP63, SP64, SP65, SP66, SP68, SP69, SP70, SP72	SP9, SP10, SP22, SP29, SP36, SP55, SP57, SP60, SP67	SP1, SP5, SP7, SP26, SP39, SP42, SP43, SP45, SP47, SP56, SP71

5.1.2. RQ2: What Type of Technologies Do Algorithms Mainly Use?

To compile the technologies used by the proposals, we recorded them as stated by the authors. The source of the classification was an open categorization, establishing the following levels:

- Meta-model: based on a meta-model for defining the problem domain or using some aspects of *modeling-driven development*.
- UML: based on formalisms of UML (class diagram, sequence diagram, state machines, etc.)
- OCL: based on extra constraints over models or languages, using OCL.
- Solver: based on the use of a constraint solving problem (CSP) to analyze the models.
- Transformations: based on using models or other representations as inputs and transforming them into another output to run some analysis.
- Other.

A major problem arose when it came to classifying the papers. Forty-six papers (app. 70%) were included in the category *Other*, and twelve (app. 18%) were classified in the *Transformations* category. This fact did not allow for a more detailed analysis of the type of technology considered in the proposals. After a detailed review of the papers classified in the *Other* category, a new categorization was proposed, which is presented below:

- Algorithm, or set of rules (ALG): based on algorithms or rules (i.e., OCL) to build or analyze the models.
- Framework (FRW): a framework considering a set of technologies and steps based on a framework.
- Graph (GRP): based on the use of directed or undirected graphs to model and analyze the variability.
- Model checking (MCK): based on model checking to verify SPLs.
- Modeling language (MLG): based on modeling languages used to map to code or other "assets".
- Natural language processing (NLP): based on NLP techniques to infer some characteristics about the models.
- Ontology (ONT): based on the use of ontologies to identify concepts or relations.
- Semantic (SEM): based on the use of semantic techniques to analyze the models.
- Solver (SOL): based on the use of tbe constraint solving problem (CSP) to analyze the models.
- State machines (STM): based on the use of state machines to analyze the models.
- Transformations (TRA): based on the use of models or other representations as inputs and transforming them into another output to run some analysis.
- Other: considers technologies named only once.

Twelve papers (18.2%) used *Transformations* as the main technology. Nine papers (13.6%) used some *Modeling language*. Seven papers (10.6%) used *Algorithms/Set of rules* or some kind of *Framework*. Six papers (9.1%) used *Model checking* or *Solvers*. Two papers (3.0%) used *Graphs, Natural Language Processing, Ontologies, Semantics, or State machines*. The remaining nine (13.6%) were classified as *Other*, including *Fuzzy logic, Goal-Oriented*

Requirements Engineering, *Markov chains, and Petri nets,* among others. See details in Figure 8. Table 11 shows the selected papers that corresponded to each technology.

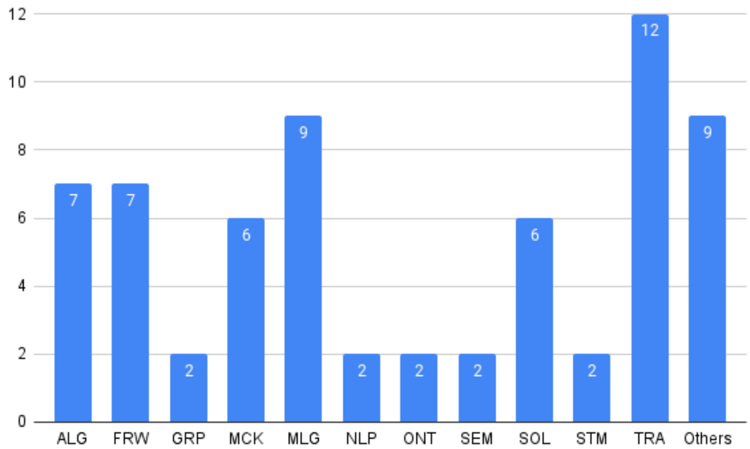


Figure 8. Technologies reported.

Table 11. Selected papers and technologies reported.

ALG	FRW	GRP	MCK	MLG	NLP	ONT	SEM	SOL	STM	TRA	Other
SP5, SP6, SP18, SP19, SP37, SP55, SP61	SP9, SP20, SP32, SP52, SP59, SP62, SP70	SP33, SP47	SP25, SP27, SP29, SP41, SP42, SP46,	SP1, SP12, SP15, SP36, SP38, SP44, SP50, SP53, SP58	SP2, SP72	SP8, SP54	SP13, SP64	SP21, SP30, SP35, SP57, SP63, SP71	SP4, SP26	SP3, SP10, SP11, SP23, SP31, SP39, SP43, SP45, SP49, SP60, SP66, SP68	SP7, SP22, SP24, SP28, SP40, SP56, SP65, SP67, SP69

5.1.3. RQ3: What Is the Origin of the Proposal?

To track the origins of the proposals, we determined where they were developed. The source of the classification was a closed categorization, leading us to establish the following levels:

- Academia (A): developed by research teams at universities.
- Industry (I): developed by commercial companies.
- Join development (A + I): joint development between academia and industry.

Fifty papers (75.8%) originated in academia. Twelve papers (18.2%) originated in industry. The four remaining papers (6.1%) were developed jointly by academia and industry. See details in Figure 9. Table 12 shows the papers according to the origin of each proposal.

Table 12. Selected papers and origins of the proposals.

A	I	A + I
SP1, SP2, SP3, SP4, SP5, SP6, SP7, SP8, SP9, SP10, SP11, SP12, SP13, SP15, SP18, SP19, SP20, SP21, SP22, SP23, SP24, SP25, SP26, SP27, SP28, SP29, SP30, SP31, SP32, SP33, SP35, SP37, SP38, SP39, SP40, SP41, SP42, SP43, SP44, SP46, SP49, SP50, SP52, SP53, SP54, SP58, SP59, SP61, SP62, SP63	SP36, SP55, SP57, SP60, SP64, SP65, SP66, SP67, SP68, SP69, SP70, SP72	SP45, SP47, SP56, SP71

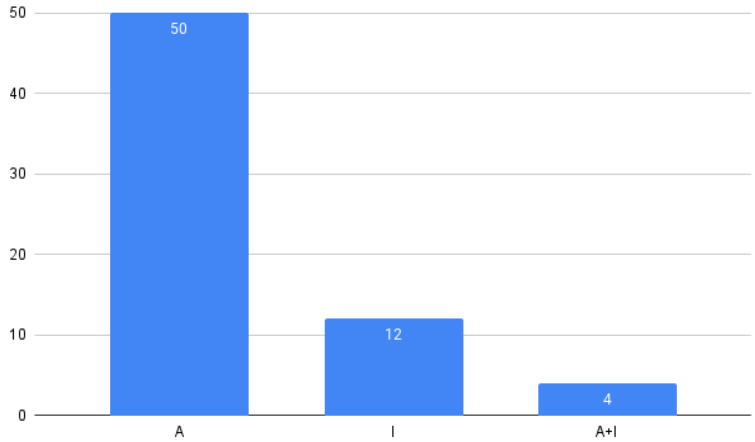


Figure 9. Origins of the proposals.

5.1.4. RQ4: What Is the Level of Validation?

To record the level of validation of the proposals and to gain an insight into the maturity level of the research, we used the taxonomy proposed by Wieringa [39]. This taxonomy considers the following categories:

- Evaluation research (EvR): the paper investigates a problem or an implementation of a technique in practice.
- Validation research (VaR): the paper investigates a solution proposal's properties that have not yet been implemented in practice.
- Solution proposal (SoP): the paper proposes a solution technique and argues for its relevance (not necessarily a full validation).
- Philosophical paper (PhP): the paper presents a new way of looking at things.
- Opinion paper (OpP): the paper contains opinions of the author about what is wrong or good about something.
- Experience paper (ExP): the paper contains a list of lessons learned by the author from his or her experience.

Twenty-four papers (36.4%) presented *Evaluation research*. Eighteen (27.3%) papers made a *Solution proposal*. Seventeen (25.8%) papers presented *Validation research*. Six (9.1%) papers were categorized as *Philosophical papers*. The remaining paper (1.5%) was categorized as an *Opinion paper*. See details in Figure 10. Table 13 shows the papers sorted according the validation of each proposal.

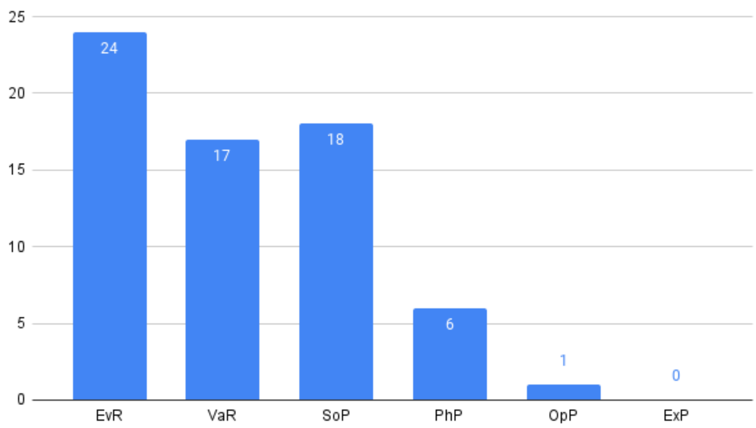


Figure 10. Validation of the proposals.

Table 13. Selected papers and validation of the proposals.

EvR	VaR	SoP	PhP	OpP	ExP
SP1, SP2, SP3, SP4, SP5, SP6, SP7, SP8, SP9, SP10, SP11, SP12, SP13, SP15, SP18, SP19, SP20, SP21, SP26, SP36, SP39, SP55, SP64, SP65	SP45, SP47, SP49, SP50, SP52, SP53, SP54, SP56, SP58, SP59, SP60, SP61, SP62, SP63, SP67, SP71, SP72	SP22, SP29, SP30, SP31, SP32, SP33, SP35, SP37, SP38, SP40, SP41, SP42, SP43, SP44, SP46, SP57, SP69, SP70	SP23, SP24, SP25, SP27, SP28, SP68	SP66	–

5.1.5. RQ5: What Kind of FM Does the Algorithm Work On?

We checked the variability of declared modelsin order to gain knowledge on the types of FMs used in the papers. The source of the classification was an open categorization, considering the following levels:

- Extended FM: considers the need to extend FMs to include more information about features (so-called feature attributes) [14].
- FODA: based on the original model proposed in [20].
- Multiplicity FM: Some authors propose extending FODA feature models with UML-like multiplicities (so-called cardinalities). The new relationships introduced in this notation are feature cardinality and group cardinality [14].
- Orthogonal variability model (OVM): The core concepts of the OVM language are variation points and variants. Each variation point has to offer at least one variant [3].
- Other.

Forty-six papers (69.7%) used the FODA model. Six papers (9.1%) used the extended FM. Three papers (4.5%) used the multiplicity FM. The remaining eleven papers (16.7%) used another representation, including multi-view feature diagrams, feature transition systems, DSML-FM, among others. See details in Figure 11. Table 14 shows the papers according to the FM used for each proposal.

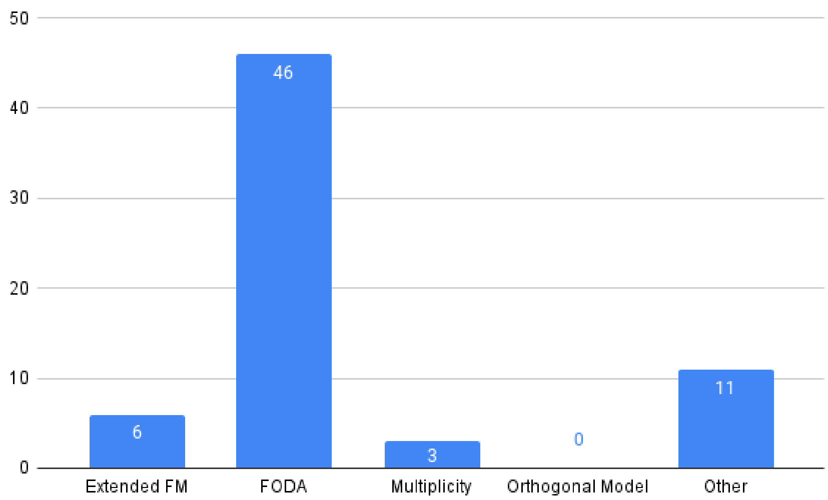


Figure 11. FMs used for each proposal.

Table 14. Selected papers and FMs used for the proposals.

Extended FM	FODA	Multiplicity	OVM	Other
SP1, SP2, SP9, SP64, SP65, SP69	SP3, SP4, SP5, SP6, SP7, SP8, SP10, SP11, SP12, SP13, SP15, SP18, SP19, SP22, SP23, SP24, SP25, SP26, SP27, SP29, SP30, SP31, SP32, SP33, SP35, SP36, SP37, SP38, SP39, SP40, SP41, SP42, SP43, SP49, SP50, SP52, SP53, SP54, SP55, SP57, SP58, SP60, SP66, SP67, SP70, SP72	SP20, SP28, SP45	–	SP21, SP44, SP46, SP47, SP56, SP59, SP61, SP62, SP63, SP68, SP71

5.1.6. RQ6: What Problems Does the Algorithm Solve?

This research question highlighted which problems had more solutions and which did not. The source of the classification was an open categorization, considering the following levels:

- Null FMs (NFM)
- Valid partial configuration (VC)
- Valid product (VP)
- Other.
- Unable to decide (UTD).

In reviewing the categories declared by the selected papers, we faced similar problems to those faced in RQ2. Forty-three papers (app. 61%) were included in the category of *Other*. This fact did not allow for a more detailed analysis of the types of problem solved by the proposals. After a detailed review of the papers, a new categorization was proposed.

We had to consider another aspect—cases in which one study aimed to solve more than one problem, so that the same paper contributed to more than one category. For example, the authors of SP3 aimed to solve the following types of problems: *Void FM, Valid product, All products, Number products, Commonality, and Variability factor*. The new categorization is presented in Table 15, including the selected papers on each category.

Table 15. Problems solved by the proposals.

Category	#Papers (%)	Selected Papers
Valid partial configuration	11 (17%)	SP1, SP6, SP12, SP22, SP36, SP37, SP43, SP46, SP64, SP67, SP71
Anomaly detection	10 (15%)	SP1, SP4, SP6, SP8, SP23, SP27, SP31, SP46, SP59, SP64
Void FM	7 (11%)	SP3, SP12, SP23, SP30, SP31, SP46, SP59
Synthesizing feature models	7 (11%)	SP15, SP18, SP44, SP45, SP49, SP54, SP60
Valid product	6 (9%)	SP3, SP25, SP28, SP30, SP31, SP59
Core Features	6 (9%)	SP6, SP29, SP30, SP46, SP59, SP64
Feature model relations	6 (9%)	SP12, SP13, SP29, SP59, SP61, SP62
All products	5 (8%)	SP3, SP23, SP30, SP59, SP64
SPL testing	5 (8%)	SP5, SP20, SP24, SP26, SP32
Optimization	5 (8%)	SP30, SP50, SP55, SP57, SP66
Number of products	4 (6%)	SP3, SP12, SP30, SP59
Commonality	4 (6%)	SP3, SP27, SP59, SP64
Model checking	4 (6%)	SP19, SP41, SP65, SP70
Variability factor	4 (6%)	SP3, SP40, SP59, SP64
Filter	4 (6%)	SP52, SP53, SP59, SP64
Dependency analysis	3 (5%)	SP23, SP43, SP56
Variant features	3 (5%)	SP29, SP30, SP59
Explanations	2 (3%)	SP8, SP71
Multi-step configuration	2 (3%)	SP38, SP64
Atomic set	2 (3%)	SP46, SP59
Change impact analysis	2 (3%)	SP29, SP47
Other	14 (21%)	SP7, SP9, SP11, SP13, SP27, SP29, SP33, SP39, SP40, SP45, SP46, SP58, SP59, SP69
UTD	8 (12%)	SP2, SP10, SP21, SP35, SP42, SP63, SP68, SP72

5.2. Answers to PQs

5.2.1. PQ1: Where Was the Paper Published?

To help researchers, we identified which journals or conferences were the most interested in this field by checking where these papers were published. The source's classification was a closed categorization, considering the levels: Conference and Journal.

Forty-six papers (69.7%) were published in some conferences. The remaining twenty papers (30.3%) were published in a journal. See details in Figure 12.

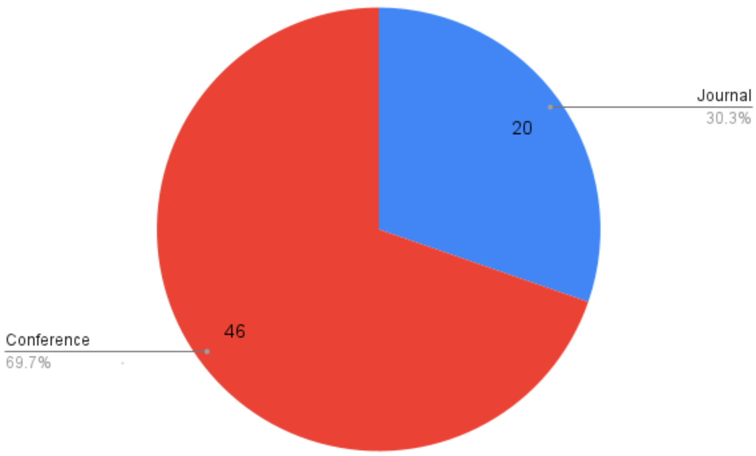


Figure 12. Number of papers according to the sources in which they were published.

Moreover, we identified the publisher of each paper. Forty papers (60.6%) were published by ACM, eighteen (27.3%) were published by Springer, seven papers (10.6%) were published by IEEE, and only one was published by Wiley (1.5%). See details in Figure 13.

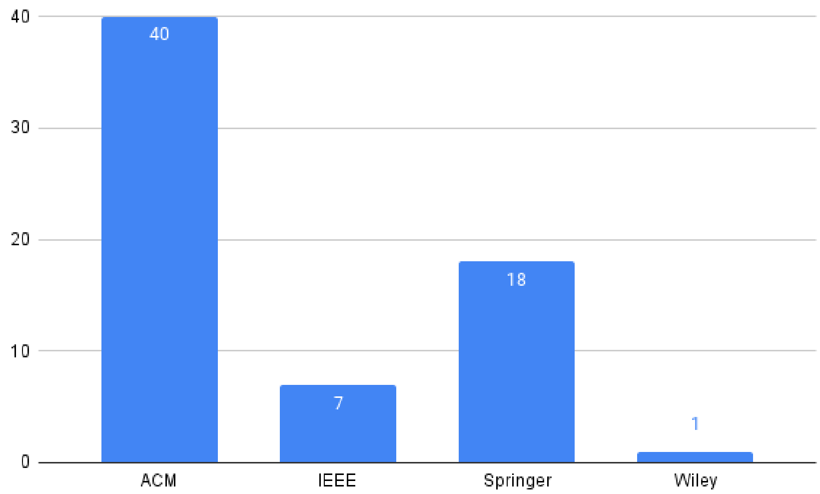


Figure 13. Number of papers according to publisher.

We also identified the journal or conference where each paper was published. For papers published in journals see Figure 14 and, for papers published in conferences see Figure 15.

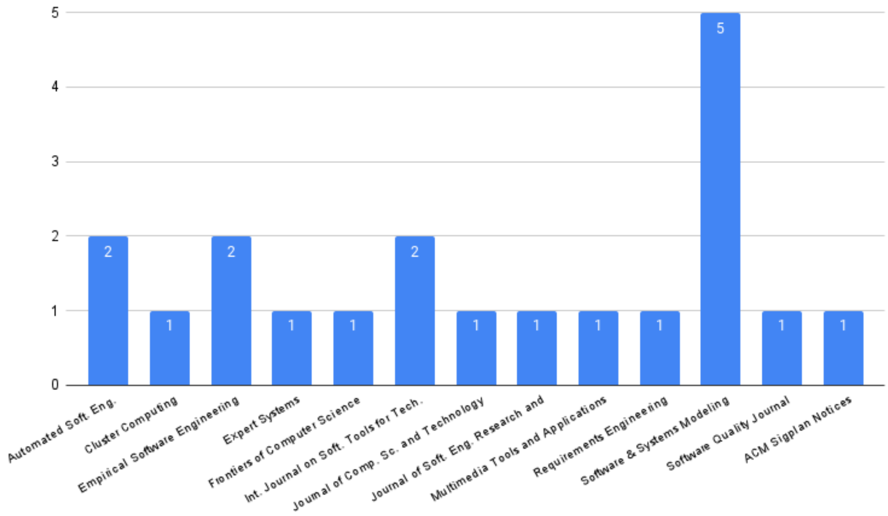


Figure 14. Papers published in journals.

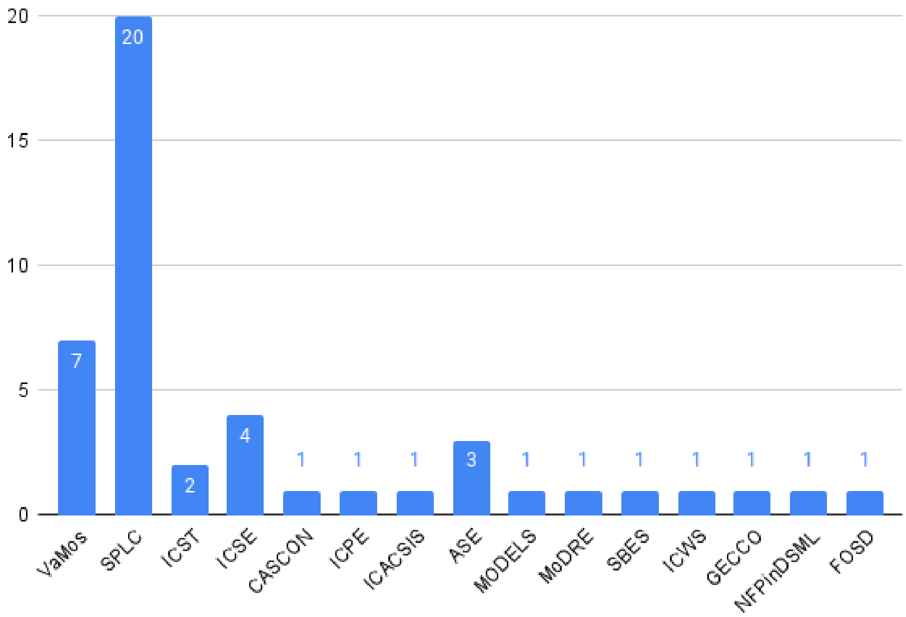


Figure 15. Papers published in conferences.

We now present a more detailed analysis of the journals and conferences. First, in the case of the journals, twenty papers came from 13 journals, which were classified according to their indexing (WoS or Scopus). Figure 16 shows the distribution of the journals according to their indexing. Figure 17 shows the distribution of JCR quartiles for those journals indexed in WoS.

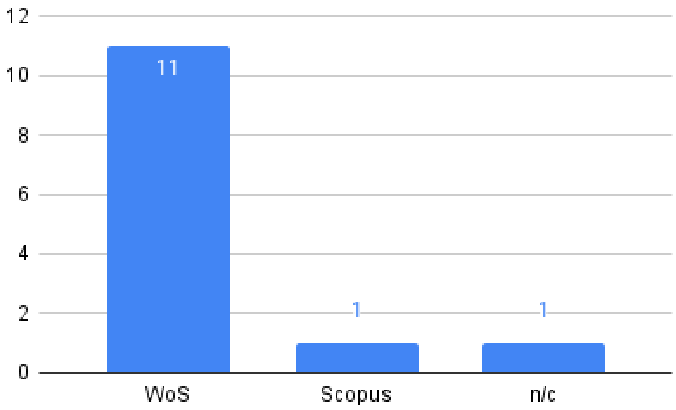


Figure 16. Journals according to their indexing.

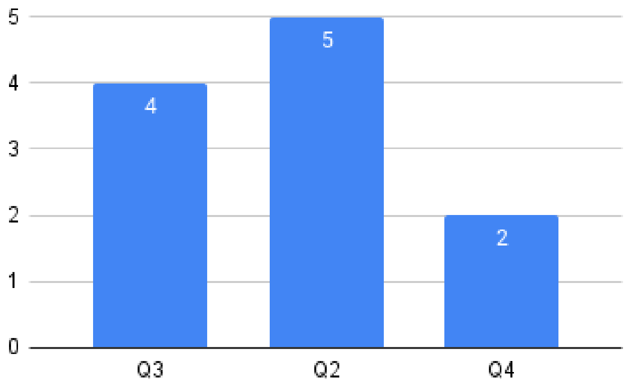


Figure 17. Journals according to their JCR quartil indexation.

Second, in the case of conferences, forty-six papers came from 15 conferences, which were classified according to their ranking (CORE or Qualis). Figure 18 shows the distribution of papers, according to Qualis ranking.

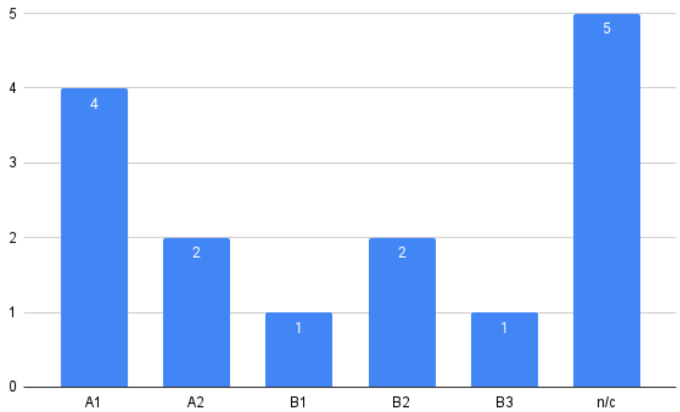


Figure 18. Conferences according to their indexing.

5.2.2. PQ2: What Was the Year of Publication of Each Paper?

To highlight how the proposals have progressed through the years, we identified the year in which each paper was published. See details in Figure 19. To see each of the papers published in each year, see Figure 20.

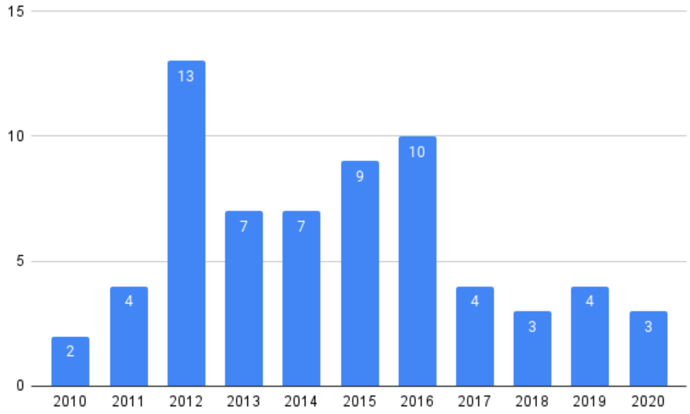


Figure 19. Number of papers published by year.

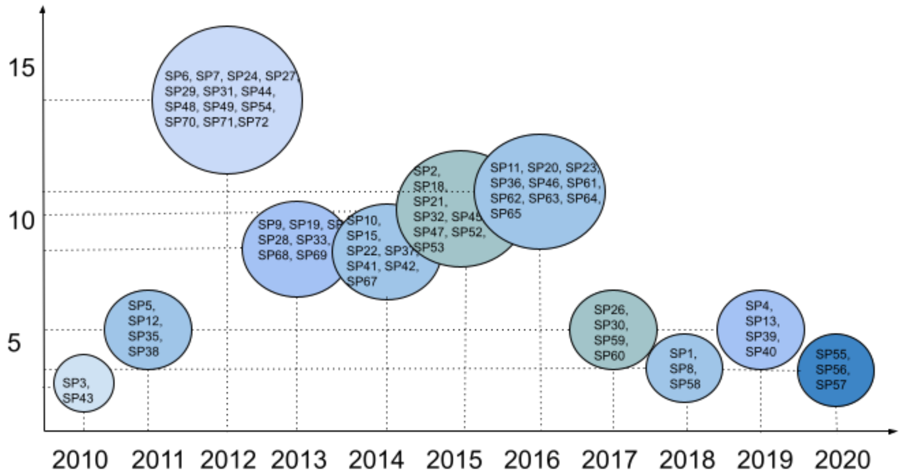


Figure 20. Papers published by year.

6. Discussion

Next, we discuss the results presented above. Section 6.1 offers an interpretation to address the RQs and PQs. Section 6.2 discusses the relationships between some RQs and the different categories identified. Section 6.3 presents a bibliometric analysis. Section 6.4 discusses the main threats to the validity of this systematic mapping. Finally, Section 6.5 summarizes the primary relationships between the responses to some RQs.

6.1. Interpreting Answers to RQs and PQs

According to [28], there has been a gap in automatic analysis in recent years. This gap presents potential areas of application of the algorithms to be proposed. Within these topics are product configuration, testing and evaluation, reverse engineering, and variability-intensive systems analysis. The issues indicated by the authors coincide with the results collected in the answers to our RQs.

6.1.1. Interpreting Answers to RQs

According to the evidence gathered to answer RQ1, we can state that around 70% of the algorithms were used in the domain engineering stage. The remaining 30% were used jointly between the *application engineering* stage and *both* stages. The significant presence of the *domain engineering* stage can be justified because FMs help to manage product family variability in the early stages of SPL development.

According to the evidence gathered to answer RQ2, we can state that our initial classification scheme was incorrect. In part, we believe this may be due to our previous work, which may have added a biasing factor [45–49]. Second, considering the new classification scheme, the most used technologies were *Transformations (app. 18%), Modeling language (app. 14%), and Algorithms and Frameworks (app. 11% for each)*. The *Other* category also reached 14%. We believe that due to this high percentage, and due to the diversity of technologies included, these results should be further reviewed to understand the impact and scope of these diverse technologies, such as NLP and Pietri networks.

Based on the evidence gathered to answer RQ3, we can state that around 76% of the proposals originated in academia, whereas less than 20% originated in industry. Only 6% of the proposals emerged from a joint effort between academia and industry. This could be because the industry is unaware of the benefits and advantages of using reasoning algorithms for their FMs. On the other hand, the industry may not have experienced the supposed gains when testing or may not be interested in including such algorithms as part of their processes. All of these judgments can be considered hypotheses of which the verification is beyond the scope of this study, and they represent an open question that requires further research.

Based on the evidence gathered to answer RQ4, we can state that there is a high level of validation for selected papers *(Evalution Research, Validation Research, and Solution Proposal categories)*, which is very interesting to contrast with the response of RQ3, which indicates that the proposals mainly originated from *Academia* (app. 76%). This percentage could be a signal that proposals at the academic level are becoming much more concerned with validations and not simply leaving their proposals as a theoretical exercise.

Based on the evidence gathered to answer RQ5, we can state that the FODA model was still the most widely used model to represent the variability of SPLs. The *Other* category was more prevalent than the three other categories combined *(extended FM, multiplicity, and orthogonal model)*. This result could indicate that in addition to the use of FMs, researchers are developing ad hoc solutions to their problem domains.

Based on the evidence gathered to answer RQ6, we can state that the types of problems most frequently solved include *valid partial configuration, anomaly detection, void FMs, and synthesizing FMs*. Even though we reviewed the *Other* category in detail, the number of papers is still relevant. The criteria we finally used considered all types of solved problems that were mentioned only once, including twenty-eight solved problems such as the *automatic analysis of performance, feature traceability, conformance faults, depth of tree, variability safety, and unique features*, among others. A case in point for the *Other* category is the paper SP40. This paper claims to solve twelve types of problems—(*maintainability, index of the feature model, depth of tree, number of features, number of leaf features, FM cognitive complexity, graph density, configuration flexibility, number of mandatory features, feature extensibility, number of valid configurations, unique cyclic-dependant features,and multiple cyclic-dependant features*).

6.1.2. Interpreting Answers to PQs

According to the evidence gathered to answer PQ1, we can state that more than 70% of the papers were published at conferences. The most relevant conferences seemed to be SPLC (20 papers), VaMoS (seven papers), ICSE (four papers), ASE (three papers), and ICST (two papers). The rest of the conferences only registered one paper each. On the other hand, 30% of the papers were published in a journal. The most relevant journals seemed to be *Software and Systems Modeling* (five papers), *Automated Software Engineering*, *Empirical Software Engineering*, and the *International Journal on Software Tools for Technology Transfer* (two papers each). The rest of the journals register only one paper each. Finally, the most relevant publishers were ACM and Springer.

Based on the evidence gathered to answer PQ2, we can state that from the year 2010 to 2016, there was a steady increase in the number of papers published. Then, from 2016 until 2020, the number of papers did not exceed four per year. One interpretation of the data could be that the topic of reasoning algorithms for FMs reached its peak. Another cause could be that research led to more specific subtopics that we have not captured in this work. Whether or not these ideas are confirmed requires much more extensive bibliometric research.

6.2. Relationships between RQs

Figure 21 shows a Sankey diagram that represents the relationships that existed between the stages (RQ1), origins (RQ3), validation levels(RQ4), and FMs (RQ5) used so far. The Sankey diagram places a visual emphasis on the transfers or flows within a system. In this case, we refer to how each selected paper responds to each RQ. Next, we present a detailed explanation of these relationships between RQs.

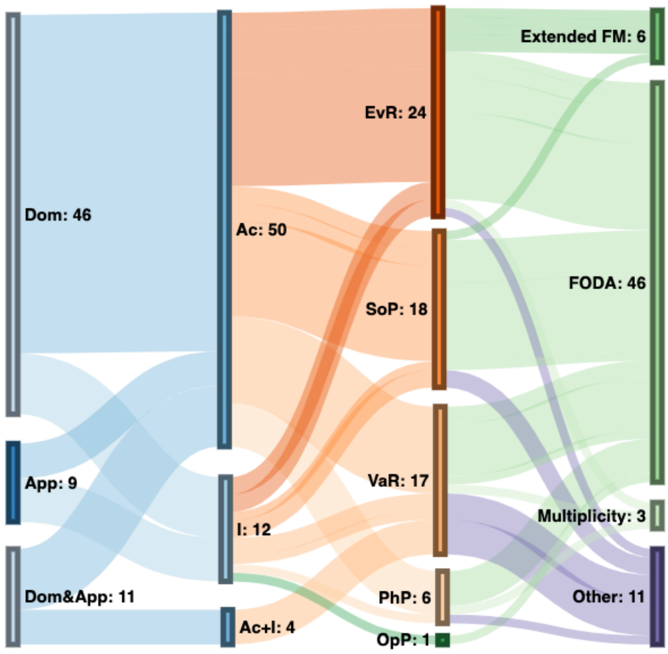


Figure 21. Relationships between stages (RQ1), origins (RQ3), validation levels(RQ4), and FMs (RQ5).

The first vertical axis represents the stages defined for RQ1, which considers domain engineering (Dom), application engineering (App), and both stages (Dom&App). The second vertical axis represents the origin of proposals defined for RQ3, comprising the academy (Ac), industry (I), and both origins (Ac + I). The third vertical axis represents the validation levelfor each proposal, comprising the categories of evaluation research (EvR), solution proposal (SoP), validation research (VaR), philosophical paper (PhP), and opinion paper (OpP). The fourth and last vertical axis represents the FM representation used.

The relationship between stages (RQ1) and origins (RQ3) show that 46 papers were associated with the domain engineering category. Thirty-nine were developed in academia, and seven were developed in industry. For the nine papers associated with the application engineering category, four were developed in academia and five in industry. Finally, for the 11 papers associated with both stages, seven were developed by academia and industry, and four were developed jointly by academia and industry.

The relationship between the origin (RQ3) and validation level(RQ4) shows that 50 papers were associated with the academy category. Twenty of these papers were categorized as evaluation research, fifteen papers were validated as solution proposals, ten papers were classified as validation research, and five papers were validated using the philosophical paper category. Of the twelve papers associated with the industry category, four were validated in the category of evaluation research, three papers were validated as solution proposal, three papers were categorized as validation research, one paper was validated using the philosophical paper category, and one paper was part of the opinion paper category. Finally, the four papers associated with a joint effort (academy + industry) were in the category of validation research.

The relationship between the validation level (RQ4) and FM (RQ5) shows that 24 papers were associated with the evaluation research category. Five of these papers used extended FM, seventeen papers used FODA, one paper used multiplicity FM, and one paper used a modeling approach classified as "other". Of the eighteen papers associated with the solution proposal category, one used extended FM, fifteen used FODA, and two used another modeling notation approach. Of the six papers associated with the philosophical paper category, four used FODA, one paper used multiplicity FM, and one used another modeling notation approach. Finally, one paper associated with the opinion paper category used FODA as its modeling approach.

6.3. Bibliometric Analysis

We conducted a bibliometric analysis of the selected papers to gain knowledge on the most relevant terms and authors and their relationships.

To provide a first impression of the topical content of the selected papers, Figure 22 shows a simple weighted word cloud generated from the titles of the selected papers, including the fifty most relevant terms. As we can observe, the word cloud trending topics are aligned with the subjects of interest of our SMS (e.g., analysis, automated, configuration, reasoning, and variability, among others), thus supporting the appropriateness of their inclusion in our study. As shown below in Figures 23–25, we present a couple of maps showing the most relevant terms and authors. We used the VOSviewer tool to build these maps.

Figure 22. Weighted word cloud derived from titles of selected papers.

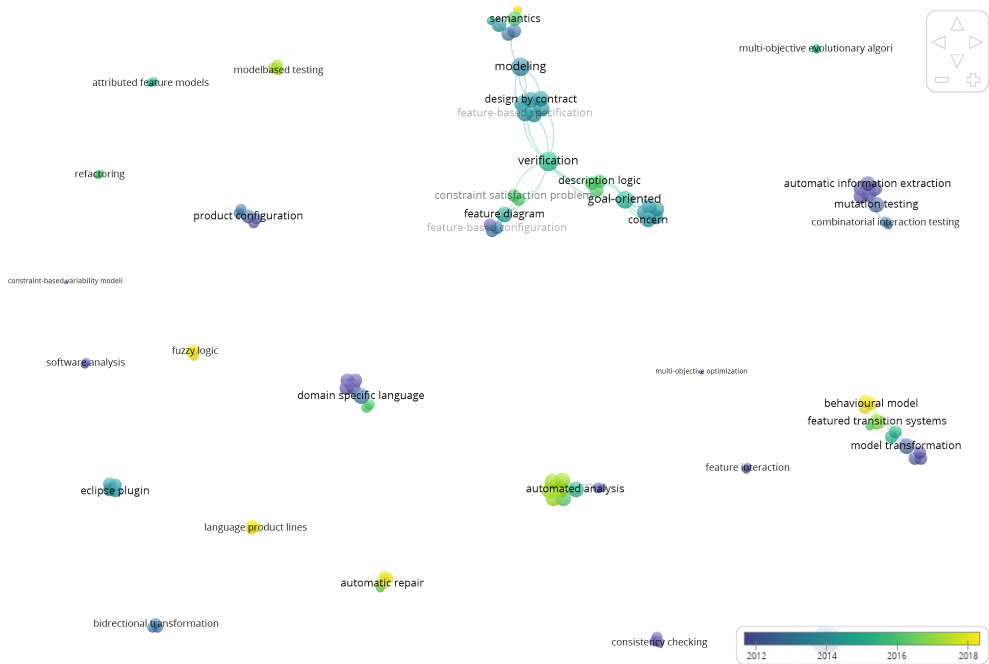


Figure 23. Relationship between most relevant terms.

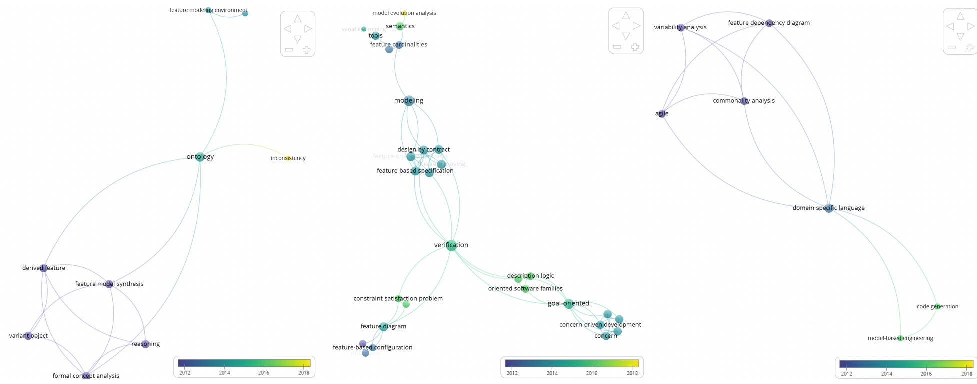


Figure 24. Details of three clusters from relevant terms.

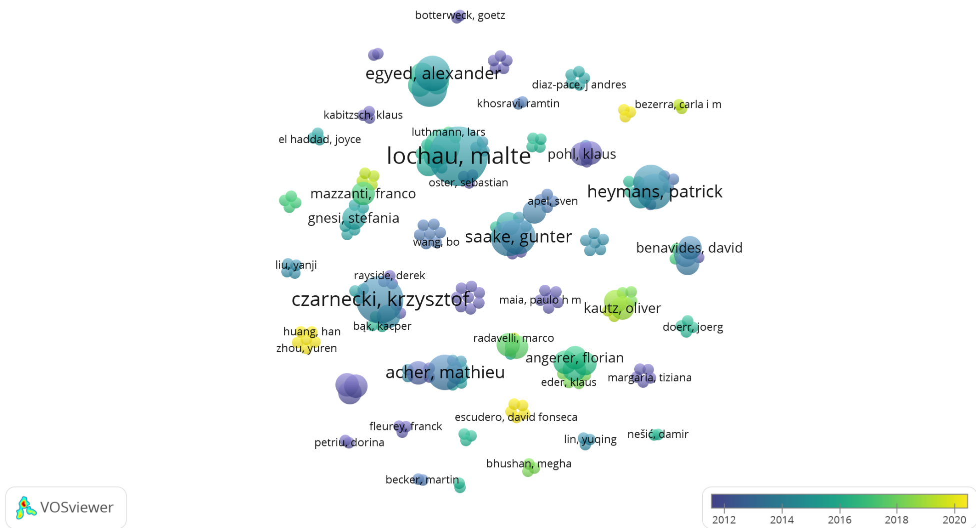


Figure 25. Relationships between the most relevant authors and their teams.

6.3.1. Most Relevant Terms

Figure 23 shows the relationship between the most relevant terms for automated analysis in the feature modeling domain, determined based on the keywords of the selected papers. The circle size corresponds to the relevance of each term, and their colors show the evolution of terms over time. In the figure, it is possible to observe 20 clusters, including 137 terms. We built a thesaurus to focus on more specific methodological and technological concepts. Furthermore, we unified the terms and all their variants under a single term (e.g., terms such as feature, feature model analysis, and feature interaction). Finally, less frequently used and less relevant terms were discarded from the analysis to focus exclusively on the most relevant ones.

Due to the size of the previous figure, it is difficult to observe details of the relationships between the different terms in the clusters. For this reason, in Figure 24 we present an enlarged view of three clusters that allows us to observe the most relevant terms and the evolution of these relationships over time. The cluster on the left shows that the terms between 2010 and 2013 dealt primarily with formalization and reasoning about FMs. Then, between 2014 and 2016, this drifted into the work related to ontologies, and finally, from 2018 onwards, was taken over bythe analysis of model inconsistencies. The central cluster shows that the terms between 2010 and 2013 were related to FM cardinality, configuration, and modeling. Then, between 2014 and 2015, work with formal specifications, and goal orientation can be observed. Between 2016 and 2017, we observed work based on logical descriptions, constraint satisfaction, semantic aspects, and model checking. Finally, from 2018 onwards, the analysis of the evolution of models for FM appears. The cluster on the right shows that the terms present between 2010 and 2013 were fundamentally related to analyzing the commonality and variability present in the FMs. Then, between 2014 and 2015 the work with DSLs appears, and finally from 2016 onwards, the literature incorporated research on automatic code generation and model-based development.

6.3.2. Most Relevant Authors and Teams

Figure 25 shows the relationships between these most relevant authors. The presented map considered the 100 most relevant authors publishing papers in the *feature modeling* and *automated analysis* domain. The size of the circles corresponds to each author's number of published papers, and their color shows the evolution of these collaborations over time. The clusters show the groups of authors working together.

6.4. Threats to Validity

The secondary studies suffered from some well-known limitations and threats to their validity that we discuss below [50]. We also discuss mitigation strategies to minimize their impact on this study.

6.4.1. Descriptive Validity

This validity criterion seeks to ensure that observations are objectively and accurately described. The associated mitigation actions were as follows.

- We structured the information to be collected by means of several forms of data extraction (for RQs and PQs) to support the uniform recording of data and to ensure the objectivity of the data extraction process.
- Moreover, all the researchers participated in an initial meeting, intending to unify concepts and criteria, answer any questions, and demonstrate (using examples) how to conduct the process.

6.4.2. Theoretical Validity

This validity criterion depends on the ability to obtain information that it is intended to be captured. The associated mitigation actions were as follows.

- We started with a search string tailored for the six most popular digital libraries in online computer science databases.
- We defined a set of exclusion criteria to ensure the objectivity of the selection process.
- The selection of articles written in English and the discarding of studies in other languages could have a minimal effect on this criterion.

6.4.3. Generalizability

This validity criterion is concerned with the ability to generalize the results of the entire domain. The associated mitigation actions were as follows.

- We ensured that our set of RQs was general enough to identify and classify the findings on aspect-oriented software development methodologies regardless of specific cases, the type of industry, etc.

6.4.4. Interpretive Validity

This validity criterion is achieved when the conclusions are reasonable, given the data. The associated mitigation actions were as follows.

- Both of the two researchers validated the conclusions.
- One researcher with experience in the problem domain helped us with the interpretation of data.

6.4.5. Repeatability

This validity criterion ensures that the research process is detailed enough that its results can be exhaustively repeated. The associated mitigation actions were as follows.

- We designed a detailed protocol to allow others to repeat the process that we have followed.
- The protocol was published online [49], so other researchers can replicate the process and, hopefully, corroborate the results.

6.5. Advances and Limitations

Figures 26 and 27 show the frequencies of publications in each selected category. The analysis was focused on presenting the frequencies of publications for each category. This analysis allowed us to see which categories have been emphasized in past research. Thus, we were able to identify gaps and possibilities for future research. These maps are two x-y scatterplots with bubbles in category intersections. The bubble size is proportional to the number of articles in the pair of categories corresponding to the bubble coordinates. We agree with Petersen that a bubble plot is a powerful tool, providing a quick overview of a field, and thus providing a map [35,51].

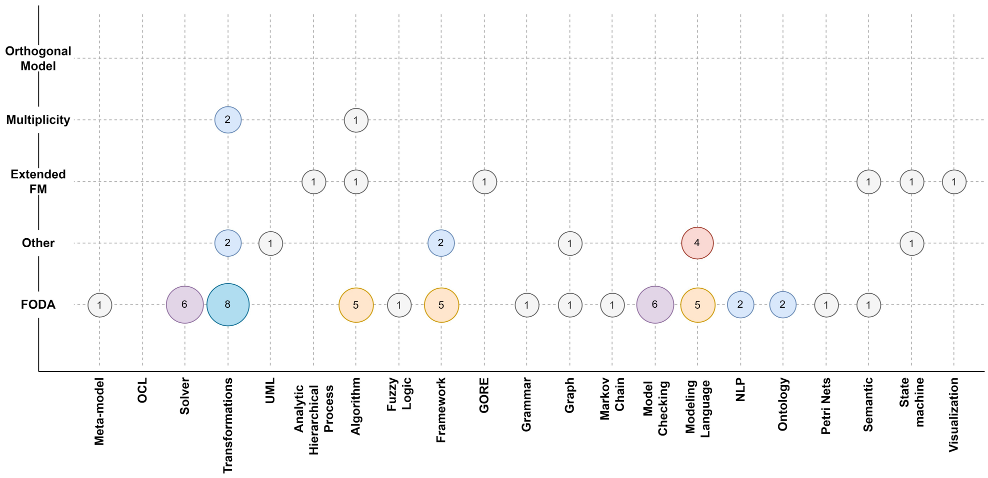


Figure 26. Summary of FM representation vs technologies implemented using reasoning algorithms.

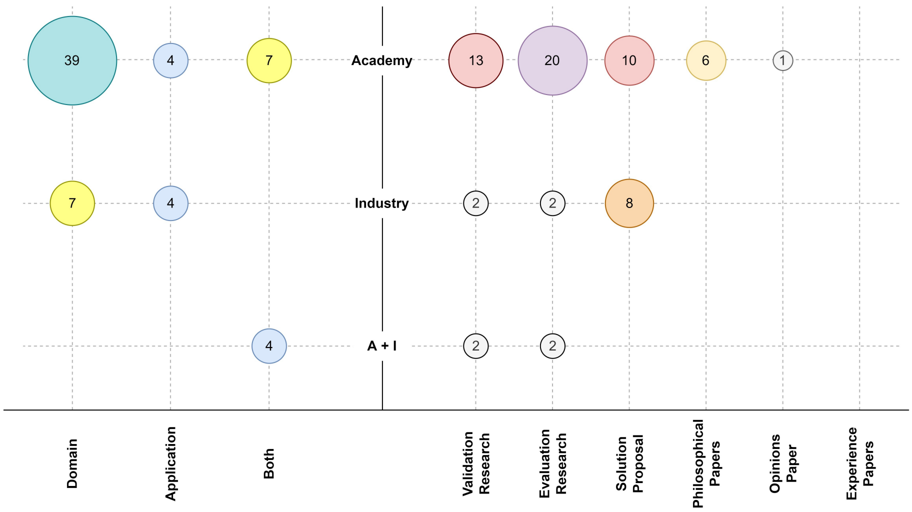


Figure 27. Summary of SLP stages, origins, and validations of proposals.

From Figure 26, the most active research areas were related to the FODA model, implementing solutions based on transformations, solvers, model checking, algorithms, frameworks, and modeling languages. On the other hand, it was possible to observe areas that were little-explored, such as meta-models, UML, GORE, Petrinetworks, and visualization techniques. The particular case of OCL stands out for not reporting evidence of use. The FM proposals with the least evidence of solution implementation were FMs with multiplicity and orthogonal models.

In Figure 27, we can see that the most active research areas were related to proposals developed in academia that considered the domain stage and to proposals in the categoriesof evaluation research, validation research, and solution proposals. It is interesting to note that the published proposals seemed to be devoted more to presenting solutions than to providing discussions or opinions regarding the work conducted both in academia and in industry. Finally, we recommended that the links between academia and industry be strengthened.

7. Conclusions

In this paper, we have presented an SMS about the use of reasoning algorithms in FMs for SPLs from 2010 to 2020. We selected 66 papers that met the inclusion and exclusion criteria. Six RQs were defined to synthesize FM proposals using reasoning algorithms. Two PQs were specified to show bibliographic and demographic characteristics.

We found that reasoning algorithms used in feature modeling were created to correct a series of problems: null FMs, valid partial configuration, and valid product problems. These algorithms were implemented using various technologies such as metamodels, UML, frameworks, graphs, check models, natural language processing, ontologies, and solvers. The most commonly used proposals involved transformations based on the use of models or other representations as inputs and transforming them into other outputs to run some analysis.On the other hand, we observed different ways of representing FMs, such as the FM extended, FODA, OVM, and multiply FM approaches. FODA was the most widely used and known by the research community.

We observed a significant presence of domain stage studies and proposals coming from academia. There was no absolute majority or trend for the problems, validation methods, or technologies present in the proposals. Given the difficulties involved in responding to RQ2 and RQ6, these require further review. We can state that the most critical conference was undoubtedly the SPLC, and the most relevant journal was *Software & Systems Modeling*. Regarding the temporal distribution of the papers, the period with the most publications was the period from 2010 to 2016. The most pertinent publisher was ACM.

We plan to define (or adapt) a set of automated analyses of FMs in our future work. In this work, we will consider upgrading our initial proposal [42,49]. This proposal considered the use of reasoning algorithms to improve the performance over large FMs and to streamline variability management in SPLs. Moreover, a replication and an updated study can also be considered, using the snow-balling technique to update the work of secondary studies [52].

Author Contributions: Conceptualization, S.S.; methodology, S.S. and A.C.; validation, S.S. and A.C.; formal analysis, S.S.; investigation, S.S.; resources, S.S. and A.C.; data curation, S.S.; writing—original draft preparation, S.S.; writing—review and editing, S.S. and A.C.;visualization, A.C.; supervision, S.S. and A.C.; project administration, S.S.; funding acquisition, S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Universidad de La Frontera, Vicerrectoría de Investigación y Postgrado. Dr. Samuel Sepúlveda thanks to research project DIUFRO DI20-0060.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Special thanks to Marcelo Esperguel for the initial ideas and discussions and Sebastian Pardo and Jonathan Jara for their useful technical support on this work.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

In Table A1 we present a summary of the related work. This summary considers the goal, RQs, time span, the numbers of papers included, and the main results of each one.

Table A1. Summary of related work.

REF.	Goal	RQs	Time Span and #Papers	Results
[14]	To provide a comprehensive literature review on the automated analysis of feature models 20 years after their invention.	RQ1: What operations of analysis on feature models have been proposed? RQ2: What kind of automated support has been proposed, and how is it performed? RQ3: What are the challenges to be faced in the future?	1990–2010 #53 papers	A catalog with 30 analysis operations identified in the literature, classifying the existing proposalsand providing automated support for them according to their underlying logical paradigms.
[28]	To provide an overview of the evolution of the automated analysis of FMs since 2010 by performing a systematic mapping study.	RQ1: Where are the papers published? RQ2: Who are the authors and institutions that conduct research on AAFM? RQ3: What are the areas in which AAFM has been applied? RQ4: What kind of publications are used to address the challenges? RQ5: When were the papers published? RQ6: What are the interrelationships among the papers?	2010–2017 #423 papers	Six different variability facets in which AAFM was applied were used to define the trends. The resultsproved the maturity in the number of journals published over the years, as well as the diversity of conferences and workshops in which papers were published.
[31]	To provide a short overview of the history and the importance of variability modeling and analysis over 30 years.	N/A (not a secondary study)	1990–2020 N/A	Variability modelling and analysis has progressed in the last three decades. One of their conclusions was that the discipline progressed faster and better when formal approaches were considered by the researchers.
[32]	To provide key research issues related to FM defects in SPLs since 1990 by performing a systematic literature review.	RQ1: What is the classification of FM defects? RQ2: What are the types of FM defects and relationships that cause these defects? RQ3: What corrective explanations have been proposed and implemented for defect removal in FMs? RQ4: What are the future challenges in the field of FM defects?	1990–2015 #77 papers	The authors derived a typology of FM defects according to their level of importance. Information on the identification of defects and explanations are provided with a formalization. Furthermore, corrective explanations are presented, incorporating various techniques used to fix defects, along with their implementation.

Appendix B

In Table A2 we present a summary of tools developed to support AAFM. This summary comprises the bibliographic reference, tool name, resume, and a link to the tool's website.

Table A2. List of tools supporting AAFM.

REF.	Tool	Resume	URL (Last access)
[53]	S.P.L.O.T.	A web application that allows the creation of FMs and offers some model reasoning functionalities. This application uses a DB engine and SAT solver to perform various analyses.	http://www.splot-research.org/ (accessed on 5 June 2021)
[54]	FAMILIAR (FeAture Model scrIpt Language for manIpulation and Automatic Reasoning)	This is a DSL for working with FMs; among the functionalities it offers are exporting, importing, editing, configuration, composition, and decomposition of models.	https://github.com/FAMILIAR-project/familiar-language (accessed on 5 June 2021)
[37]	FaMa	This is an Eclipse plugin for modeling variability using FMs with multiplicity. In particular, through external reasoners, the application allows one to perform automated analysis on the created models.	https://www.isa.us.es/fama/?FaMa_Framework (accessed on 5 June 2021)
[55]	Feature IDE	This is is an open-source framework for feature-oriented software development based on Eclipse.	https://featureide.de/ (accessed on 5 June 2021)

Appendix C

In Table A3 we present some details on the selected papers. These details include the paper ID, title, authors, publication year, source, and publisher of each one.

Table A3. List of selected papers.

ID	Title-Authors-Year-Source-Publisher
SP1	Controlled and Extensible Variability of Concrete and Abstract Syntax with Independent Language Features. Butting, A.; Eikermann, R.; Kautz, O.; Rumpe, B.; Wortmann, A., 2018, VaMoS, ACM.
SP2	CMT and FDE: Tools to Bridge the Gap between Natural Language Documents and Feature Diagrams. Ferrari, A.; Spagnolo, G.; Gnesi, S.; Dell'Orletta, F., 2015, SPLC, ACM.
SP3	Automated Test Data Generation on the Analyses of Feature Models: A Metamorphic Testing Approach. S. Segura; R. M. Hierons; D. Benavides; A. Ruiz-Cortés, 2010, ICST.
SP4	Static Analysis of Featured Transition Systems. Beek, M. H.; Damiani, F.; Lienhardt, M.; Mazzanti, F.; Paolini, L., 2019, SPLC, ACM.
SP5	Pairwise Feature-Interaction Testing for SPLs: Potentials and Limitations. Oster, S.; Zink, M.; Lochau, M.; Grechanik, M., 2011, SPLC, ACM.
SP6	An Algorithm for Generating T-Wise Covering Arrays from Large Feature Models. Johansen, M.F.; Haugen, O.; Fleurey, F., 2012, SPLC, ACM.
SP7	User-Friendly Approach for Handling Performance Parameters during Predictive Software Performance Engineering. Tawhid, R.; Petriu, D., 2012, ICPE, ACM.
SP8	Improving quality of software product line by analysing inconsistencies in feature models using an ontological rule-based approach. Bhushan, M.; Goel, S.; Kumar, A., 2018, Expert Systems, Wiley.
SP9	Mining Complex Feature Correlations from Software Product Line Configurations. Zhang, B,; Becker, M., 2013. VaMoS, ACM.
SP10	Handling Complex Configurations in Software Product Lines: A Tooled Approach. Urli, S.; Blay-Fornarino, M.; Collet, P., 2014, SPLC, ACM.
SP11	Automatic Detection and Removal of Conformance Faults in Feature Models. P. Arcaini; A. Gargantini; P. Vavassori, 2016, ICST, IEEE.
SP12	Managing Feature Models with Familiar: A Demonstration of the Language and Its Tool Support. Acher, M.; Collet, P.; Lahire, P.; France, R.B., 2011, VaMoS, ACM.
SP13	Semantic Evolution Analysis of Feature Models. Drave, I.; Kautz, O.; Michael, J.; Rumpe, B., 2019, SPLC, ACM.
SP15	WebFML: Synthesizing Feature Models Everywhere. Bécan, G.; Ben Nasr, S.; Acher, M.; Baudry, B., 2014, SPLC, ACM.
SP18	Synthesis of Attributed Feature Models from Product Descriptions. Bécan, G.; Behjati, R.; Gotlieb, A.; Acher, M., 2015, SPLC, ACM.

Table A3. *Cont.*

ID	Title-Authors-Year-Source-Publisher
SP19	Beyond Boolean Product-Line Model Checking: Dealing with Feature Attributes and Multi-Features. Cordy, M.; Schobbens, P-Y.; Heymans, P.; Legay, A., 2013, ICSE, ACM.
SP20	Featured Model-Based Mutation Analysis. Devroey, X.; Perrouin, G.; Papadakis, M.; Legay, A.; Schobbens, P-Y.; Heymans, P., 2016, ICSE, ACM.
SP21	SAT-Based Analysis of Large Real-World Feature Models is Easy. Liang, J.H.; Ganesh, V.; Czarnecki, K.; Raman, V., 2015, SPLC, ACM.
SP22	Automated Verification of Feature Model Configuration Processes Based on Workflow Petri Nets. Mennicke, S.; Lochau, M.; Schroeter, J.; Winkelmann, T., 2014, SPLC, ACM.
SP23	Multi-View Modeling and Automated Analysis of Product Line Variability in Systems Engineering. Nešić, D.; Nyberg, M., 2016, SPLC, ACM.
SP24	Grammar-Based Test Generation for Software Product Line Feature Models. Bagheri, E.; Ensan, F.; Gasevic, D., 2012, CASCON, ACM.
SP25	Strategies for Product-Line Verification: Case Studies and Experiments. Apel, S.; Rhein, A. von; Wendler, P.; Größlinger, A.; Beyer, D., 2013, ICSE, ACM.
SP26	Modeling and Testing Product Lines with Unbounded Parametric Real-Time Constraints. Luthmann, L.; Stephan, A.; Bürdek, J.; Lochau, M., 2017, SPLC, ACM.
SP27	Towards Fixing Inconsistencies in Models with Variability. Lopez-Herrejon, R.E.; Egyed, A., 2012, VaMoS, ACM.
SP28	Discrete Time Markov Chain Families: Modeling and Verification of Probabilistic Software Product Lines. Varshosaz, M.; Khosravi, R., 2013, SPLC, ACM.
SP29	Squid: An Extensible Infrastructure for Analyzing Software Product Line Implementations. Vianna, A.; Pinto, F.; Sena, D.; Kulesza, U.; Coelho, R.; Santos, J.; Lima, J.; Lima, G., 2012, SPLC, ACM.
SP30	Extending the automated feature model analysis capability of the abstract behavioral specification. Achda, A. C. ; Azurat, A.; Muschevici, R.; Setyautami, M. R. A., 2017, ICACSIS, IEEE.
SP31	Safe Adaptation in Context-Aware Feature Models. Marinho, F.; Maia, P.; Andrade, R.; Vidal, V.; Costa, P.; Werner, C., 2012, FOSD, ACM.
SP32	Fault-Based Product-Line Testing: Effective Sample Generation Based on Feature-Diagram Mutation. Reuling, D.; Bürdek, J.; Rotärmel, S.; Lochau, M.; Kelter, U., 2015, SPLC, ACM.
SP33	Measuring the structural complexity of feature models. Pohl, R.; Stricker, V.; Pohl, K., 2013, ASE, IEEE.
SP35	A performance comparison of contemporary algorithmic approaches for automated analysis operations on feature models. Pohl, R.; Lauenroth, K.; Pohl, K., 2011, ASE, IEEE.
SP36	Multi-Variability Modeling and Realization for Software Derivation in Industrial Automation Management. Fang, M.; Leyh, G.; Doerr, J.; Elsner, C., 2016, MODELS, ACM.
SP37	Combined propagation-based reasoning with goal and feature models. Yanji, L.; Yukun, S.; Xinshang, Y.; Mussbacher, G., 2014, MoDRE, IEEE.
SP38	Multi-Dimensional Variability Modeling. Rosenmüller, M.; Siegmund, N.; Thüm, T.; Saake, G., 2011, VaMoS, ACM.
SP39	A Process for Fault-Driven Repair of Constraints Among Features. Arcaini, P.; Gargantini, A.; Radavelli, M., 2019, SPLC, ACM.
SP40	Development of the Maintainability Index for SPLs Feature Models Using Fuzzy Logic. de Oliveira, D.; Bezerra, C., 2019, SBES, ACM.
SP41	Potential Synergies of Theorem Proving and Model Checking for Software Product Lines. Thüm, T.; Meinicke, J.; Benduhn, F.; Hentschel, M.; von Rhein, A.; Saake, G., 2014, SPLC, ACM.
SP42	Low-Level Variability Support for Web-Based Software Product Lines. Machado, I.; Santos, A,; Cavalcanti, Y,; Trzan, E.; de Souza, M.; de Almeida, E., 2014, VaMoS, ACM.
SP43	A Feature-Oriented Approach for Web Service Customization. Nguyen, T.; Colman, A., 2010, ICWS, IEEE.
SP44	Domain Specific Feature Modeling for Software Product Lines. Hofman, P.; Stenzel, T.; Pohley, T.; Kircher, M.; Bermann, A., 2012, SPLC, ACM.
SP45	Extracting Variability-Safe Feature Models from Source Code Dependencies in System Variants. Assunçao, W.; Lopez-Herrejon, R.; Linsbauer, L.; Vergilio, S.; Egyed, A., 2015, GECCO, ACM.
SP46	Feature-Model Interfaces: The Highway to Compositional Analyses of Highly-Configurable Systems. Schröter, R.; Krieter, S.; Thüm, T.; Benduhn, F.; Saake, G., 2016, ICSE, ACM.
SP47	Configuration-Aware Change Impact Analysis. Angerer, F.; Grimmer, A.; Prähofer, H.; Grünbacher, P., 2015, ASE, ACM.
SP49	Efficient Synthesis of Feature Models. Andersen, N.; Czarnecki, K.; She, S.; Wąsowski, A., 2012, SPLC, ACM.
SP50	Modelling and Multi-Objective Optimization of Quality Attributes in Variability-Rich Software. Olaechea, R.; Stewart, S.; Czarnecki, K.; Rayside, D., 2012, NFPinDSML, ACM.
SP52	Using FMC for Family-Based Analysis of Software Product Lines. ter Beek, M.; Fantechi, A.; Gnesi, S.; Mazzanti, F., 2015, SPLC, ACM.
SP53	Managing the Variability in the Transactional Services Selection. Gamez, N.; El Haddad, J.; Fuentes, L., 2015, VaMoS, ACM.

Table A3. *Cont.*

ID	Title-Authors-Year-Source-Publisher
SP54	Reasoning of Feature Models from Derived Features. Ryssel, U.; Ploennigs, J.; Kabitzsch, K., 2012, SIGPLAN Notices, ACM.
SP55	A novel hybrid approach for feature selection in software product lines. Hitesh Y.; Charan K., 2020, Multimedia Tools and Applications, Springer.
SP56	Connecting domain-specific features to source code: towards the automatization of dashboard generation. Vázquez-Ingelmo, A.; García-Peñalvo, F.; Therón, R.; Filvà, D.; Escudero, D., 2020, Cluster Computing, Springer.
SP57	Going deeper with optimal software products selection using many-objective optimization and satisfiability solvers. Yi, X.; Xiaowei, Y.; Yuren, Z.; Zibin, Z; Miqing, L.; Han, H., 2020, Empirical Software Engineering, Springer.
SP58	Multi-purpose, multi-level feature modeling of large-scale industrial software systems. Rabiser, D.; Prähofer, H.; Grünbacher, P.; Petruzelka, M.; Eder, K.; Angerer, F.; Kromoser, M.; Grimmer, A., 2018, Software & Systems Modeling, Springer.
SP59	FLAME: a formal framework for the automated analysis of software product lines validated by automated specification testing. Durán, A.; Benavides, D.; Segura, S.; Trinidad, P.; Ruiz-Cortés, A., 2017, Software & Systems Modeling, Springer.
SP60	Multi-objective reverse engineering of variability-safe feature models based on code dependencies of system variants. Assunção, W.; Lopez-Herrejon, R.; Linsbauer, L.; Vergilio, S,; Egyed, A., 2017, Empirical Software Engineering, Springer.
SP61	Reasoning about product-line evolution using complex feature model differences. Bürdek, J.; Kehrer, T.; Lochau, M.; Reuling, D.; Kelter, U.;, Schürr, A., 2016, Automated Software Engineering, Springer.
SP62	A Feature Model Based Framework for Refactoring Software Product Line Architecture. Tanhaei, M.; Habibi, J.; Mirian-Hosseinabadi, S.-H., 2016, Journal of Computer Science and Technology, Springer.
SP63	Clafer: unifying class and feature modeling. Bąk, K.; Diskin, Z.; Antkiewicz, M.; Czarnecki, K.; Wąsowski, A., 2016, Software & Systems Modeling, Springer.
SP64	Attribute-based variability in feature models. Ahmet, Serkan, Karataş; Halit, O., 2016, Requirements Engineering, Springer.
SP65	Goal-oriented modeling and verification of feature-oriented product lines. Mohsen, A.; Gerd, G.; Bardia, M.; Dragan, G., 2016, Software & Systems Modeling, Springer.
SP66	An approach based on feature models and quality criteria for adapting component-based systems. Sanchez, E.; Diaz-Pace, A.; Zunino, A.; Moisan, S.; Rigault, J.P., 2015, Journal of Software Engineering Research and Development, Springer.
SP67	Quality attribute modeling and quality aware product configuration in software product lines. Guoheng, Z.; Huilin, Y.; Yuqing, L., 2014, Software Quality Journal, Springer.
SP68	Supporting multiple perspectives in feature-based configuration. Hubaux, A.; Heymans, P.; Schobbens, P.Y.; Deridder, D.; Khalil Abbasi, E., 2013, Software & Systems Modeling, Springer.
SP69	Supporting feature model refinement with updatable view. Bo, Wang; Zhenjiang, HuQiang; SunHaiyan, Zhao; Yingfei, Xiong; Wei, Zhang; Hong, Mei, 2013, Frontiers of Computer Science, Springer.
SP70	A constraint-based variability modeling framework. Jörges, S.; Lamprecht,A.-L.; Tiziana, MargariaIna; Schaefer, B., 2012, International Journal on Software Tools for Technology Transfer, Springer.
SP71	Visualization of variability and configuration options. Pleuss, A.; Botterweck, G., 2012, International Journal on Software Tools for Technology Transfer, Springer.
SP72	Decision support for the software product line domain engineering lifecycle. Bagheri, E.; Ensan, F; Gasevic, D., 2012, Automated Software Engineering, Springer.

References

1. Van der Linden, F.J.; Schmid, K.; Rommes, E. *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007.
2. Clements, P.; Northrop, L. *Software Product Lines, Course Notes of Product Line Systems Program*; Software Engineering Institute, Carnegie Mellon University: Pittsburgh, PA, USA, 2003.
3. Pohl, K.; Böckle, G.; van Der Linden, F.J. *Software Product Line Engineering: Foundations, Principles and Techniques*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2005.
4. Ahmed, F.; Capretz, L.F. The software product line architecture: An empirical investigation of key process activities. *Inf. Softw. Technol.* **2008**, *50*, 1098–1113. [CrossRef]
5. Chen, L.; Babar, M.A. A systematic review of evaluation of variability management approaches in software product lines. *Inf. Softw. Technol.* **2011**, *53*, 344–362. [CrossRef]
6. Asikainen, T.; Mannisto, T.; Soininen, T. A unified conceptual foundation for feature modelling. In Proceedings of the 10th International Software Product Line Conference (SPLC'06), Baltimore, MD, USA, 21–24 August 2006; pp. 31–40. [CrossRef]
7. Raatikainen, M.; Tiihonen, J.; Männistö, T. Software product lines and variability modeling. *J. Syst. Softw.* **2019**, *149*, 485–510. [CrossRef]
8. Geraldi, R.T.; Reinehr, S.; Malucelli, A. Software product line applied to the Internet of Things: A systematic literature review. *Inf. Softw. Technol.* **2020**, *124*, 106293. [CrossRef]

9. Ji, W.; Berger, T.; Antkiewicz, M.; Czarnecki, K. Maintaining feature traceability with embedded annotations. In Proceedings of the 19th International Conference on Software Product Line, Nashville, TN, USA, 20–24 July 2015; pp. 61–70.
10. Krüger, J.; Gu, W.; Shen, H.; Mukelabai, M.; Hebig, R.; Berger, T. Towards a better understanding of software features and their characteristics: A case study of marlin. In Proceedings of the 12th International Workshop on Variability Modelling of Software-Intensive Systems, Madrid, Spain, 7–9 February 2018; pp. 105–112.
11. Bhushan, M.; Goel, S. Improving software product line using an ontological approach. *Sādhanā* **2016**, *41*, 1381–1391. [CrossRef]
12. Hähnle, R.; Schaefer, I. The quest for formal methods in software product line engineering. In *Software Technology: 10 Years of Innovation in IEEE Computer*; John Wiley & Sons: Hoboken, NJ, USA, 2018.
13. Heradio, R.; Perez-Morago, H.; Fernandez-Amoros, D.; Cabrerizo, F.J.; Herrera-Viedma, E. A bibliometric analysis of 20 years of research on software product lines. *Inf. Softw. Technol.* **2016**, *72*, 1–15. [CrossRef]
14. Benavides, D.; Segura, S.; Ruiz-Cortés, A. Automated analysis of feature models 20 years later: A literature review. *Inf. Syst.* **2010**, *35*, 615–636. [CrossRef]
15. Arboleda, H.; Royer, J.C. *Model-Driven and Software Product Line Engineering*; John Wiley & Sons: Hoboken, NJ, USA, 2012. [CrossRef]
16. Apel, S.; Batory, D.; Kästner, C.; Saake, G. Software product lines. In *Feature-Oriented Software Product Lines*; Springer: New York, NY, USA, 2013; pp. 3–15. [CrossRef]
17. Camacho, M.C.; Álvarez, F.; Collazos, C.; Leger, P.; Bermúdez, J.D.; Hurtado, J.A. A Collaborative Method for Scoping Software Product Lines: A Case Study in a Small Software Company. *Appl. Sci.* **2021**, *11*, 6820. [CrossRef]
18. Galster, M.; Weyns, D.; Tofan, D.; Michalik, B.; Avgeriou, P. Variability in software systems—A systematic literature review. *IEEE Trans. Softw. Eng.* **2013**, *40*, 282–306. [CrossRef]
19. Sinnema, M.; Deelstra, S. Industrial validation of COVAMOF. *J. Syst. Softw.* **2008**, *81*, 584–600. [CrossRef]
20. Kang, K.C.; Cohen, S.G.; Hess, J.A.; Novak, W.E.; Peterson, A.S. *Feature-Oriented Domain Analysis (FODA) Feasibility Study*; Technical Report CMU/SEI-90-TR-021; Carnegie-Mellon University: Pittsburgh, PA, USA; Software Engineering Institute: Pittsburgh, PA, USA, 1990.
21. Czarnecki, K.; Wasowski, A. Feature diagrams and logics: There and back again. In Proceedings of the 11th International Software Product Line Conference (SPLC 2007), Kyoto, Japan, 10–14 September 2007; pp. 23–34. [CrossRef]
22. Batory, D. Feature models, grammars, and propositional formulas. In Proceedings of the International Conference on Software Product Lines, Rennes, France, 26–29 September 2005; pp. 7–20. [CrossRef]
23. Benavides, D.; Trinidad, P.; Ruiz-Cortes, A. Automated reasoning on feature models. In Proceedings of the International Conference on Advanced Information Systems Engineering, Porto, Portugal, 13–17 June 2005; pp. 491–503. [CrossRef]
24. Benavides, D.; Galindo, J.A. Automated analysis of feature models: Current state and practices. In Proceedings of the 22nd International Systems and Software Product Line Conference, Gothenburg, Sweden, 10–14 September 2018; Volume 1, p. 298. [CrossRef]
25. Lettner, M.; Rodas, J.; Galindo, J.A.; Benavides, D. Automated analysis of two-layered feature models with feature attributes. *J. Comput. Lang.* **2019**, *51*, 154–172. [CrossRef]
26. Cook, S.A. The complexity of theorem-proving procedures. In Proceedings of the Third Annual ACM Symposium on Theory of Computing, Shaker Heights, OH, USA, 3–5 May 1971; pp. 151–158.
27. Bryant, R.E. Graph-based algorithms for boolean function manipulation. *IEEE Trans. Comput.* **1986**, *100*, 677–691. [CrossRef]
28. Galindo, J.; Benavides, D.; Trinidad Martín Arroyo, P.; Gutiérrez, A.; Ruiz, A. Automated analysis of feature models: Quo vadis? *Computing* **2019**, *101*, 387–433. [CrossRef]
29. Gacitúa, R.; Sepúlveda, S.; Mazo, R. FM-CF: A framework for classifying feature model building approaches. *J. Syst. Softw.* **2019**, *154*, 1–21. [CrossRef]
30. Galindo, J.A.; Benavides, D. A Python framework for the automated analysis of feature models: A first step to integrate community efforts. In Proceedings of the 24th ACM International Systems and Software Product Line Conference-Volume B, Montreal, QC, Canada, 19–23 October 2020; pp. 52–55. [CrossRef]
31. Benavides, D. Variability modelling and analysis during 30 years. In *From Software Engineering to Formal Methods and Tools, and Back*; Springer: Cham, Switzerland, 2019; pp. 365–373.
32. Bhushan, M.; Negi, A.; Samant, P.; Goel, S.; Kumar, A. A classification and systematic review of product line feature model defects. *Softw. Qual. J.* **2020**, *28*, 1507–1550. [CrossRef]
33. Segura, S.; Galindo, J.A.; Benavides, D.; Parejo, J.A.; Ruiz-Cortés, A. BeTTy: Benchmarking and testing on the automated analysis of feature models. In Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems, Leipzig, Germany, 25–27 January 2012; pp. 63–71.
34. Pohl, R.; Lauenroth, K.; Pohl, K. A performance comparison of contemporary algorithmic approaches for automated analysis operations on feature models. In Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011), Lawrence, KS, USA, 6–10 November 2011; pp. 313–322.
35. Petersen, K.; Vakkalanka, S.; Kuzniarz, L. Guidelines for conducting systematic mapping studies in software engineering: An update. *Inf. Softw. Technol.* **2015**, *64*, 1–18. [CrossRef]

36. Kitchenham, B.A.; Budgen, D.; Brereton, O.P. The value of mapping studies—A participant-observer case study. In Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering (EASE), Durham, UK, 12–13 April 2010; pp. 1–9.
37. Benavides, D.; Trinidad, P.; Ruiz-Cortés, A.; Segura, S. Fama. In *Systems and Software Variability Management*; Springer: Cham, Switzerland, 2013; pp. 163–171. [CrossRef]
38. Kitchenham, B.A.; Charters, S. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*; EBSE Technical Report EBSE-2007-01; Keele University: Newcastle, UK, 2007.
39. Wieringa, R.; Maiden, N.; Mead, N.; Rolland, C. Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Requir. Eng.* **2006**, *11*, 102–107. [CrossRef]
40. Brereton, P.; Kitchenham, B.A.; Budgen, D.; Turner, M.; Khalil, M. Lessons from applying the systematic literature review process within the software engineering domain. *J. Syst. Softw.* **2007**, *80*, 571–583. [CrossRef]
41. Petticrew, M.; Roberts, H. *Systematic Reviews in the Social Sciences: A Practical Guide*; John Wiley & Sons: New York, NY, USA, 2008.
42. Sepúlveda, S.; Esperguel, M. Systematic Mapping Protocol: Reasoning Algorithms on Feature Model. *arXiv* **2021**, arXiv:2103.16325.
43. Gwet, K. Inter-rater reliability: Dependency on trait prevalence and marginal homogeneity. *Stat. Methods Inter-Rater Reliab. Assess. Ser.* **2002**, *2*, 1–9.
44. Fleiss, J. *Statistical Methods for Rates and Proportions*; John Wiley & Sons: New York, NY, USA, 1981.
45. Esperguel, M.; Sepúlveda, S. Feature modeling tool: A proposal using ADOxx technology. In Proceedings of the 2016 XLII Latin American Computing Conference (CLEI), Valparaiso, Chile, 10–14 October 2016; pp. 1–9. [CrossRef]
46. Esperguel, M.; Sepúlveda, S.; Monsalve, E. FMxx: A proposal for the creation, management and review of feature models in software product lines. In Proceedings of the 36th International Conference of the Chilean Computer Science Society (SCCC), Arica, Chile, 16–20 October 2017; pp. 1–7. [CrossRef]
47. Esperguel, M.; Sepúlveda, S. From UML/OCL to ADOxx specifications: How to do it. In Proceedings of the 2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA), Concepcion, Chile, 17–19 October 2018; pp. 1–6. [CrossRef]
48. Sepúlveda, S.; Cravero, A.; Cachero, C. Requirements modeling languages for software product lines: A systematic literature review. *Inf. Softw. Technol.* **2016**, *69*, 16–36. [CrossRef]
49. Sepúlveda, S.; Bobadilla, A.; Espinoza, M.; Esparza, V. Solving errors detected in feature modeling languages: A proposal. In Proceedings of the International Conference on Information Technology & Systems, Libertad City, Ecuador, 4–6 February 2021; pp. 375–385.
50. Petersen, K.; Gencel, C. Worldviews, research methods, and their relationship to validity in empirical software engineering research. In Proceedings of the 2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement, Ankara, Turkey, 23–26 October 2013; pp. 81–89.
51. Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M. Systematic mapping studies in software engineering. In Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), Bari, Italy, 26–27 June 2008; pp. 1–10.
52. Wohlin, C.; Mendes, E.; Felizardo, K.R.; Kalinowski, M. Guidelines for the search strategy to update systematic literature reviews in software engineering. *Inf. Softw. Technol.* **2020**, *127*, 106366. [CrossRef]
53. Mendonca, M.; Branco, M.; Cowan, D. SPLOT: Software product lines online tools. In Proceedings of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications, Orlando, FL, USA, 25–29 October 2009; pp. 761–762. [CrossRef]
54. Acher, M.; Collet, P.; Lahire, P.; France, R.B. Familiar: A domain-specific language for large scale management of feature models. *Sci. Comput. Program.* **2013**, *78*, 657–681. [CrossRef]
55. Thüm, T.; Kästner, C.; Benduhn, F.; Meinicke, J.; Saake, G.; Leich, T. FeatureIDE: An extensible framework for feature-oriented software development. *Sci. Comput. Program.* **2014**, *79*, 70–85. [CrossRef]