ISSN: 4621 - 4803
Volume 12, Number 2, 2025
Innovative Journal of Science and Technology Research

ADVANCING SOFTWARE PRODUCT LINES: A SYSTEMATIC REVIEW OF TOOLS, TECHNIQUES AND METHODOLOGIES

Muhammad Garba; Muhammad Nura Malami; Hassan Suru Umar; Musa Sulaiman Argungu
Department of Computer Science, Abdullahi Fodio University of Science and Technology, Aliero, Kebbi State, Nigeria
Corresponding Author: Garbamga@gmail.com
Contributing Authors: Nuragwn44@gmail.com; Suruhassan@yahoo.com; Sm279@gmail.com

Abstract: Software Product Line (SPL) refers to a set of software systems that share a common core while allowing for systematic variations to meet different requirements. This study conducted a systematic literature review (SLR) to identify and categorizes various tools and techniques into model-driven and component-based approaches, offering insight into their application in SPL variability management. Through comprehensive analysis of SLR, the research identified the existing SPL tools, and later select and deselect from those identified data based on designed inclusion and exclusion criteria. These findings contribute to the ongoing discourse on improving SPL methodologies and provide a foundation for future advancements in tool development.

Keywords: Variability Modelling, Feature Models, Visualization Techniques, Variants, Variation Points, Spl Techniques, Spl Tool.

INTRODUCTION

Background of the Study

Over the last two and a half decades, Software Product Lines Engineering (SPLE) has increasingly gained the attention of researchers and practitioners alike. This is due to the potential economic advantages and business competitiveness the SPLE process can bring. The benefits can range from cutting the development cost and increasing software quality, to enabling mass customization, market dominance, and reduced time to market (Preuveneers et al., 2016).

Product lines complexity in software is tremendously increasing, diversity is evolving, and the lead time is also decreasing (Marchezan et al., 2020). The easiest way of handling these challenges is the reuse of the existing product lines in the software in meeting user demands. For example, take mobile phone as an example of user demand; all mobile phones have the menus for making calls, storing contacts information, typing, sending, receiving and storing messages, etc. It will be very expensive and time consuming in developing different software for each kind of mobile phone. This brings about the concept of software product family, product population, and product lines. To integrate all software within hardware three different approaches were employed. These approaches are Integration-Oriented platform, Hierarchal and Composition-Oriented. The integration-oriented approach is a classical approach used for many years in industry but unable to meet the currently increasing usage challenges. Hierarchical and Composition- Oriented approaches are popularly now-a-days used to meet the industrial challenges (Ignaim & Fernandes, 2019).

The SPLE technique provides a systematic way to reuse software assets. These assets are the software artefacts or resources associated with your products. The artefacts include, but are not limited to requirements analysis, design specifications, software implementation, configuration, test plans, test cases, etc. The assets are then engineered to be shared across the entire product line, i.e., to be used in multiple products. Therefore, SPLE is a technique that optimizes the reuse of existing software assets by creating multiple applications that share many features, while still exhibiting certain differences. SPLE allows for the planned reuse of artefacts among the software systems under development (Abbas et al., 2020).

Software product lines are increasingly used in managing expectations or demands of custom-tailored software variants. They provide reuse of software paired with variability mechanism systematically, in the code to implement the whole product families instead of single software

products. Consequently, it’s used in both Academia and Industry (Reinhartz-Berger & Kemelman, 2020).
Variability implementation: There is a number of approaches exist in this to realize variability in code, which can be categorized into annotative and compositional approaches (Belarbi & Englebert, 2019). Its sometimes referred to as approaches for positive and negative variability (El-Sharkawy et al., 2019).
Software variability is the ability of a software system to exist in different variants. Among others, variants arise from a diversity of hardware, operating environments, and customer demands—referred to as variability drivers in the remainder. A variety of strategies and mechanisms to manage variability has been proposed (Castro et al., 2021; Rhein et al., 2018).
This work tend to primarily focused on study by identifying the existing tools, techniques and/or methodologies that manages variability in SPL, noting their weaknesses and strength. This will serve as basis for other researchers to derive interest in any of the identified weaknesses. This work is limited to answering two research questions (RQs).

RESEARCH QUESTIONS
The following research questions have been defined to help achieve the study aim and objectives.
i. What are the Tools developed to handle and manage variability in Software Product Lines (SPL)?
ii. What are the features/characteristics of these Tools?

Related Works
A lot of work has been conducted in the area which resulted in many approaches including various techniques, methods, and tools. Typical Examples of these include early methods is Kahraman & Cleophas, (2022), the paper details the application of the Family Evaluation Framework (FEF) for evaluating software product lines in the avionics domain. It encompasses the methodology used, challenges faced, benefits observed, and the customization of interview questions to align with the organization's needs. Additionally, it offers examples of analysis, provides a list of questions for assessment, and presents a comprehensive account of using FEF in evaluating software product lines in large organizational settings. Tool is a web-based tool that supports the modeling of feature models for dynamic software product lines (DSPLs) and evaluates these models from maintainability measures with their respective thresholds. It also includes an adaptation mechanism for dynamic reconfiguration based on MAPE-K, a repository of feature models, and user authentication. The paper describes the tool's architecture, features, and functionality in detail (Lima et al., 2023).
Based on Abbas et al., (2018) Feature Oriented Reuse Method (FORM) a method that searches and captures common and different features of an application and using the analysis results to develop domain architecture and components. According to Ommen & Rock (2019) Visualization of Variability in complex Dev. Structures addresses the issue of visualizing large product lines and discusses different techniques which can be used towards an efficient visualization of variability. Furthermore, the study presents alternative methods to visualize complex constraints and group cardinalities. These techniques were illustrated with the help of a prototypical implementation. (Acher et al., 2018) presented an overview of research that applies visualization techniques in SPLE activities. The work is aimed at researchers and practitioners in the field of software engineering who are interested in the application of visualization techniques to SPLE. (Gharsellaoui et al., 2021) proposes a new UML profile named SPL-UML that can model the variabilities in the context of software product line (SPL) design. The proposed profile treats the functional, static, and dynamic aspects and defines the concept of recommendation at the use case, class, and sequence diagrams. The paper describes the coherence between the proposed extensions, which is ensured by OCL (Object Constraint Language) constraints. The proposed profile includes several meta-classes, such as Class, Feature, Comment, Variation Point, Actor, UseCase, Package, Interface, and Interaction, and defines several concepts, such as variant, alternative, requires, excludes, mandatory, recommended, and variation

point. The paper provides examples of how the proposed profile can be used to model SPL design and improve the traceability between the feature model and design.
(Eilers et al., 2019) presents a proposal for extending the Stereotype-based Management of Variability (SM arty) approach to manage variabilities in UML sequence diagrams and an experimental validation that provides evidence of the effectiveness of this extension. (Mazanek & Rutetzki, 2011) The paper discusses why model comparison tools are crucial for the judge and which of their features are most important. It further, describe how a first model comparison tool has been integrated into their system and how the judge could be used to compare comparison tools. (De Oliveira & De Almeida, 2016) This paper presents the results of an exploratory study aiming to support SPL engineers choosing the feature modelling tool that best fits their needs. This exploratory study compares and analyses two feature modelling tools, namely FeatureIDE and SPLOT, based on data from 56 participants that used the analyzed tools.

IV RESEARCH METHODOLOGY
In the first part, a Systematic Literature Review (SLR) approach will be used to identify gaps in the body of knowledge and answer RQ1/RQ2. An SLR is a formal and rigorous way used to carefully examine, evaluate and interpret an identified research evidence based on research questions or a particular research area (Sauer & Seuring, 2023).
The aim was to systematically review the reported literature on variability management tools in software product lines (known as primary studies). The process of SLR involved three main phases which are:
1) Planning the Review: - which has three stages:
- Identifying the need for a review (importance).
- Indicating the research question(s).
- Developing and evaluating a review protocol.

2) Conducting the Review: - which consists of four stages:
- Identification of primary studies.
- Selection of primary studies based on clear criteria.
- Assessing the quality of primary studies.
- Data extraction and synthesis.

3) Reporting the Review: - which involves:
- Writing and formatting the main report and
- Evaluating/drawing conclusions based on the findings

Data sources for this work combine both the manual search and electronic database (journals). Table 2 contained some electronic and manual data sources used to access recently done work. After the search strategy section, is the information selection criteria. This section is talking about the information to be include or excluded from the sourced data. In the E section, the work is trying to test the Quality of the selected data so we named the section as Quality Assessment section. Lastly, the study conducted Data Synthesis and Extraction in Section 3.5. The two sections were successfully conducted by a collaborative work with the course supervisors.

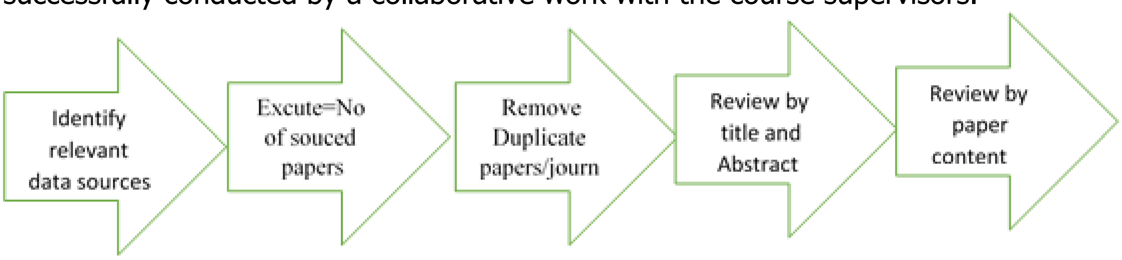

Figure 1: Study Selection Process

A. Search Strategy:

The Data search initially starts with the Electronic sources. Despite it is not possible to apply or use a general search term on all electronic resources, a search term has been design that caters all to help us answer some Questions in the Research Questions section of chapter one.

With the help of (Hierons et al., 2020; Setyautami & Hähnle, 2021) a search term was designed below.

≪Variability AND (Product Line∗ OR Software Product Lines OR Software Product Family OR Software Product Families OR Product Family OR Product Families∗ OR Systems Family OR Family of Systems) AND (Variability OR Variability Management OR Variant OR Variation Point OR Feature Model OR Feature Modeling or Feature Modelling) AND (Tool OR Tools OR Approach, Approaches, Method∗ OR Methods)≫

The search string was designed based on the following protocols that will help answer Q1 in the research questions earlier mentioned.

- Use logical terminologies such as "OR" and "AND" to link alternative spellings and join the synonyms, words, or phrases.
- Derive main terms from topic being search and research questions.
- Determine and include synonyms, related terms and alternative spellings for major terms
- Include other relevant terms where there is a possibility to identify material about the search topic.
- Check the keywords in all relevant papers researchers already knew and initial searches on the relevant databases

Other search strings have also been employed to help us obtain more data. As mentioned earlier, a manual data search was employed from different Software Product Lines data store such as the workshops and conferences.

Electronic data sources include 7 digital library as shown in Table 2. The manual search covered 5 conferences and workshops altogether as also shown in Table 2 below

Table 2 List of Electronic Data Sources

SN	Digital Library	Manual Library
1	ACM Digital Library	Variability Modeling Workshop
2	IEEE Xplore	International Workshop on Visualizatio SPLE (VisPLE)
3	Science Direct	Software Product Lines Conference (SPL
4	Microsoft Academic Search	Workshop on Variability Modeling Software-Intensive System (VaMoS)
5	IEEE Computer Society Digital Lib	Software Product Lines Conference
6	Google Scholar	
7	Web of Science	

B. Data Selection Criteria:

This section is comprised of two steps; Inclusion Criteria (IC) parts and the Exclusion Criteria part (EC). The Inclusion part deals with deals with some qualities a research work has to meet for it to be considered part of the work to be included. These include four major qualities led down.

The second part, Exclusion Criteria comprises of five led down qualities that if a research work consist any of the features, the work is not considered in this research work.

C. Data Inclusion Criteria:
A list of some qualities that a research need to meet for it to be considered. These qualities are:
1. A study must be scientific, peer reviewed and should not be PowerPoint presentation, short or extended abstract.
2. If there exist numerous research on same area, the most recent report is considered.
3. English written report. The report should not be in any language other than English
4. The research primarily talked about variability management tool, technique or methodology in SPL.

D. Data Exclusion Criteria:
Some qualities that disqualify a paper to be considered in this research work.
1. The paper does not primarily discussed about variability management in SPL.
2. The paper focused about commercial variability management in SPL.
3. If the report is a compilation report of different works.
4. The research work published earlier than 2015.
5. A PowerPoint presentation or short report.

In conducting Selection/Exclusion Criteria, a total number of 568 work was gathered from different data sources as mentioned in Table 3. After the initial screening of works' abstract 425 were screened out based on non SPL related study criteria. Out of the 143 selected studies a review were conducted leaving 75 screening out 68. After further and careful study, the study selection reduced to 28. Figure 1 depicts selection processes. The 28 selected primary studies are shown in table 5.

E. Quality Assessment Section
The selected studies were weighted based on some quality selection criteria. A selection criteria by (Yadav et al., 2021)were used in the process. The selection process tend to give a score card for each quality. Where each were given a score rate of 1(for Yes) 0(No). However, where the rate is neither Yes or No, a score rate of 0.5 will be given. The score rate questions was designed to identify the rationale; description of the context area; work limitations; etc. as comprehensively shown in Table 3 below.

Table 3: Quality Assessment Criteria

Question	Value (1:Yes; 0.5: Partial; 0:No)
Q1. Is there a rationale for why the study was undertaken?	
Q2 Is there a description of the context (e.g. industry, laboratory setting, products used, etc) in which the research was carried out?	
Q3. Did the paper present enough details about the variability tool to enable us conduct the required analysis?	
Q4. Did the paper present an evaluation of the tool? If yes, did it include feedback from end users?	
Q5. Are the substantive claims in the paper supported by reliable evidence?	

Q6. Do the authors compare and evaluate their own results against related work?	
Q7. Do the authors discuss the credibility of their own findings?	
Q8. Are the limitations of the study discussed explicitly?	

The questions in Table 3 above will provide more flexibility in answering some of the questions that may be difficult to judge as to "Yes" or "No" from the information provided in the primary study. Once score is allocated to each question, an aggregate mark was then given to each study. The data will also be helpful in the accuracy of answering question RQ1.

F. Data Extraction:

Table 4: Data Extraction Questions

S/No	Data Extraction Question	Description
Q1	Paper title	Paper given name by the author(s)
Q2	Year of publication	The year paper is made available on a journal source
Q3	Type of publication	E.g. journal, conference, workshop, etc.
Q4	Publication outlet	e.g. conference name, etc.
Q5	Paper brief description	Short note about the paper contents
Q6	Research rationale, challenges or problems as reported in the paper (to be categorized)	A rationale or problem faced as reported in the paper
Q7	Research context (e.g., industry, academic, product etc.)	The context upon which research was conducted e.g. industrial, academic, commercial etc.
Q8	Tool performance and stability	How stable and stability the tool was designed
Q9	Visualization technique?	Does the tool/research conducted address visualization technique?
Q10	Textual notation?	Does the tool/research conducted address textual notation?
Q11	Usability?	Does the tool/research conducted address usability issue(s)?
Q12	Tool environment/platform?	What is the environment/platform was the tool implemented e.g. pure::variants
Q13	Integration?	Do the address issues support integration feature with other tools/platform?
Q14	Scalability?	Is the designed model scalable?
Q15	Relevance (Research or Practice)?	What is the relevance of the research? Is it practically implemented work or research based?
Q16	Research limitations as reported in the paper?	Snapshot of the research limitations

Out of the 564 selected studies, 28 were identified as shown in the below:

Table 5: Selected primary studies in the review

Paper ID	Paper Title	Year of Publication

MG7	DyMMer 2.0: A Tool for Dynamic Modeling and Evaluation of Feature Model	2021
NM8	A Tool for Modeling and Analysis of Relationships among Feature Model Views	2021
NM9	Variability Extraction and Modeling for Product Variants	2021
NM10	A systematic mapping study of information visualization for software product line engineering	2017
NM11	Variability Modeling with a SPL-UML Profile	2016
NM13	Variability Identification and Representation in Software Product Line UML Sequence Diagrams: Proposal and Empirical Study	2014
NM14	Automating the Extraction of Model-based Software Product Lines from Model Variants	2015
SA15	ArchSPL-MDD: An ADL-based Model-Driven Strategy for Automatic Variability	2015
SA16	A Self-Adaptation Mechanism for Variability Management in Dynamic Software Product Lines	2023
SA17	Towards a Semiautomatic Tool to Support the Integration of Feature Models	2019
SA18	Approaching Pattern Catalogues for Variability Realization in Model-Based Systems and Software Engineering.	2023
SA19	Visualization of Variability in Complex Development Structures.	2015
SA20	Software Variability Management: An Exploratory Study with Two Feature Modeling Tools	2017
SA21	Assessment of Product Derivation Tools in the Evolution of Software Product Lines: An Empirical Study.	2020
SN22	Variant management solution for large scale software product lines	2018
SN23	On the importance of model comparison tools for the automatic evaluation of the correctness of model transformations	2019
SN24	An elementary theory of product-line variation	2022
SN25	Product Line Bundles to Support Product Derivation in Multi Product Lines	2023
MG1	The state of adoption and the challenges of systematic variability management in industry	2020
MG2	An Infrastructure for Composing Build Systems of Software Product Line	2020
MG3	Approaching Pattern Catalogues for Variability Realization in Model-Based Systems and Software Engineering	2023
MG4	An approach to support variant management on safety analysis using CHESS error models	2020
MG5	Efficiently Managing the Impact of Hardware Variability on GPUs' Streaming Processors	2018
MG6	A Pattern-based Modeling Approach for Software Product Line Engineering	2022

SN26	Variability Management for Automated Production Systems Using Product Lines and Feature Models	2016
SN27	Managing Variability of Logistics Robot System: Experience at Hitach	2023
SN28	Online Resource Allocation With Machine Variability: A Bandit Perspectiv	2020
NM12	Seamless Variability Management With the Virtual Platform	2021

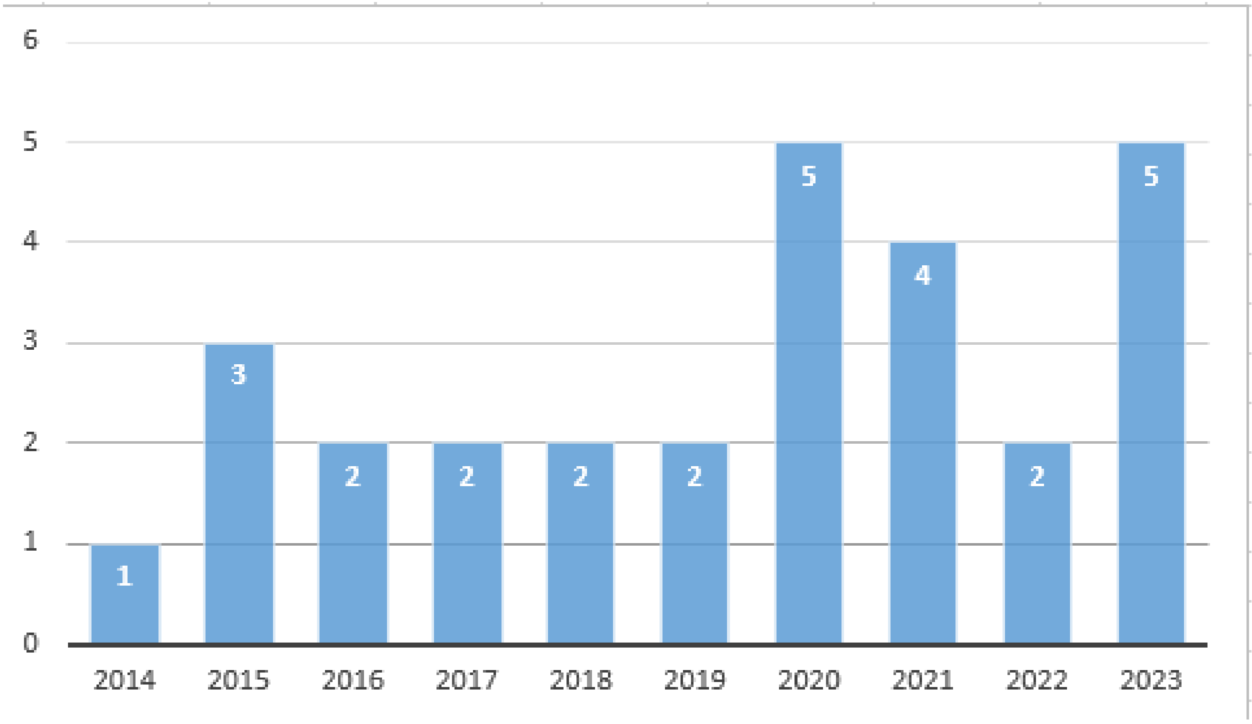

Figure 2: A Pie chart of the selected studies over time

A number of studies were rejected due to one or more reason(s) of the laid down ECs listed section 3.3.2 above listening them will make this report bulky.

Data Meta-Analysis:

This step presents the selected primary studies in respect to publication types, venues, trends, and overall characteristics. As earlier mentioned 28 papers scales through the Ics and Ecs out of the 568 primary studies.

Table 6: Distribution of primary studies on publication outlet

Paper ID	Type of Publication
MG7	Conference
NM8	Journal
NM9	Conference
NM10	Journal
NM11	Journal
NM13	Conference
NM14	Conference
SA15	Symposium/Conference
SA16	Symposium/Conference
SA17	Conference
SA18	Conference

SA19	Conference
SA20	Symposium/Conference
SA21	Conference
SN22	Conference
SN23	Journal
SN24	Journal
SN25	Journal
MG1	Journal
MG2	Journal
MG3	Conference
MG4	Conference
MG5	Journal
MG6	Conference
SN26	Journal
SN27	Conference
SN28	Journal
NM12	Conference

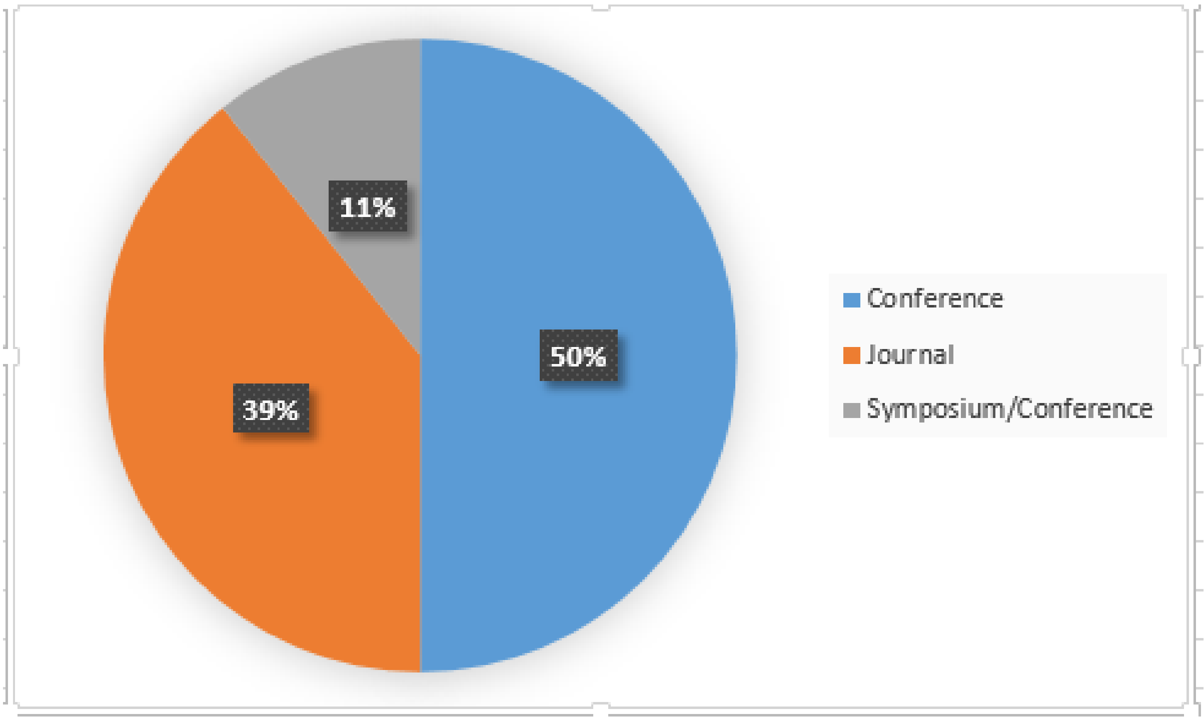

Figure 3: Percentage of selected primary studies on publication outlet

Figure 3 above shows the distinction of studies over time. The search did not consider any work earlier than 2014 and later than 2024 as mention in the ICs and ECs criteria. The chart presents that most of the relevant publications are conference based followed by the journal papers and symposium publications with least number of papers.

Figure 2 relates the distribution of primary studies over time having 2020 and 2023 with the highest number of relevant studies in the selected primary data source.

Table 7: List of works and features/characteristics

S/No	Name of author(s)	Year of Publication	Title	Strengths	Weaknesses
1	Lima	2021	A Tool for Dynamic Modeling and Evaluation of Feature Model	The paper primarily focuses on the extension of the DyMMer tool to a web version, incorporating new features for modeling feature models from Software Product Lines (SPLs) and Dynamic Software Product Lines (DSPLs), development of an adaptation mechanism for FM of DSPLs, repository of FMs, inclusion of thresholds for measures, and user authentication.	The tool does not support modeling of cross-tree contraints, which are important for representing complex relationships between features in a feature model. The tool's adaptation mechanism is based on MAPE-K, which is a rules-based approach that may not be suitable for all DSPLs. The tool's adaptation mechanism mechanism is based on a single context agent, and future work could explore the use of multiple context agents to improve the tool's effectiveness.
2	Kahraman	2021	A Tool for Modeling and Analysis of Relationships among Feature Model	The paper details the application of the Family Evaluation Framework (FEF) for evaluating software product lines in the avionics domain. It encompasses the methodology used, challenges faced, benefits observed, and the customization of interview questions to align with the organization's needs. Additionally, it offers examples of analysis, provides a list of questions for assessment, and presents a comprehensive account of using FEF in evaluating software product	The tool does not support the modeling limitations of cross-tree constraints. The tool does not support for the automatic generation of relationship models. The tool has not been evaluated in terms of its scalability The tool has not been compared to the other existing tools for modeling and analyzing relationships among feature model views.

				lines in large organizational settings.	
3	Roberto	2017	A systematic mapping study of information visualization for software product line engineering.	The paper presents an extended systematic mapping study on the subject of information visualization for software product line engineering (SPLE). The paper is aimed at researchers and practitioners in the field of software engineering who are interested in the application of visualization techniques to SPLE.	Difficulty in finding the implementation tools used: The authors found it challenging to identify the implementation tools used in the primary sources they analyzed. This limitation may have affected the completeness of their analysis. Provenance of the evaluation examples: The authors also found it challenging to determine the provenance of the evaluation examples used in the primary sources. This limitation may have affected the reliability of the empirical evaluations reported in the primary sources. Classification criteria: The authors faced the challenge of determining the classification criteria for the primary sources and reaching a consensus on the classification of each paper. This limitation may have affected the accuracy of their analysis.
4	Jihen	2016	Variability Modeling with a SPL-UML Profile	The paper proposes a new UML profile named SPL-UML that can model the variabilities in the context of software product line (SPL) design. The proposed profile treats the functional, static, and dynamic aspects and defines the concept of recommendation at	The proposed SPL-UML profile does not provide information about the integration capability or tool environment/platform used to develop or implement the profile. The proposed SPL-UML profile can generate some inconsistencies, such as if an optional class contains a mandatory method, the resulting model is incoherent.

				the use case, class, and sequence diagrams. The paper describes the coherence between the proposed extensions, which is ensured by OCL (Object Constraint Language) constraints. The proposed profile includes several meta-classes, such as Class, Feature, Comment, Variation Point, Actor, UseCase, Package, Interface, and Interaction, and defines several concepts, such as variant, alternative, requires, excludes, mandatory, recommended, and variation point.	
5	Anderson	2017	Variability Identification and Representation in Software Product Line UML Sequence Diagrams: Proposal and Empirical Study.	Variability Identification and Representation in Software Product Line UML Sequence Diagrams: Proposal and Empirical Study" presents a proposal for extending the Stereotype-based Management of Variability (SMarty) approach to manage variabilities in UML sequence diagrams and an experimental validation that provides evidence of the effectiveness of this extension.	Did not measured the performance rigorously based on prototyping and has been limited to product specific evaluation
6	Jabier	2020	Automating the Extraction of Model-based Software Product Lines	The paper presents an approach named MoVa2PL for automating the extraction of model-	The paper mentions that MoVa2PL does not consider constraints that could be defined using the Object Constraints Language (OCL).

			from Model Variants.	based software product lines from model variants of software systems. It describes the process of reverse engineering the source code of ArgoUML and its six variants related to diagram edition support as UML models, and then applying MoVa2PL to decompose the models into atomic model elements (AMEs) and identify interdependent AMEs that form blocks of features.	Also, the constraints discovery only checks requires and excludes constraints between pairs of features, and more complex structural constraints involving more than two features are not considered
7	Ana	2020	ArchSPL-MDD: An ADL-based Model-Driven Strategy for Automatic Variability.	The paper presented ArchSPL-MDD as a strategy based on the integration of MDD and SPL approaches for specifying an SPL with customized products. The proposed strategy comprises a process with activities used to transform and refine the models specified in LightPL-ACME ADL. Moreover, ArchSPL-MDD is generic and encompasses the creation and representation of the feature models and the system architecture by using LightPL-ACME for specifying the referenced architecture and the product family, product generation, and the product	The authors did not verify the quality of the code generated by the ArchSPL-MDD tool The applicability of the proposed strategy to other different domain was not verified and The applicability of the tool in other case studies of a different domain was also not ascertained

				source code. In addition, the authors developed the ArchSPL-MDD tool to support the proposed MDD process and the automatic product derivation. The ArchSPL-MDD strategy was evaluated in a controlled experiment using GingaForAll, an SPL for the Ginga digital TV middleware. The experiment also pointed out that the ArchSPL-MDD strategy shown to be better in terms of efficiency, complexity, and expressiveness when compared to GingaForAll an UML-based MDD strategy.	
8	Rafael	2023	A Self-Adaptation Mechanism for Variability Management in Dynamic Software Product Lines.	This paper proposes an adaptation mechanism for DSPL feature models that supports dynamic variability based on the MAPE-K model and the mechanism has been explicitly discussed. The authors implemented the mechanism in the DyMMer 2.0 modelling tool and tested its performance using various DSPL feature models. Additionally, they performed a preliminary evaluation with a proof-of-concept	• The study has limited knowledge base to share information that could aid in the execution of reconfiguration processes, potentially improving the construct validity of the mechanism. • The study did not consider the missing essential features in the adaptation engine interface in DyMMer 2.0, such as the ability to delete and edit agents and contexts and to visualize constraints between features and mapped states.

				study with an expert to assess its practical usage. The results of the study have shown the effectiveness and practicality of the proposed mechanism in managing variability in complex and dynamic environments.	
10	Juliana	2020	Software Variability Management: An Exploratory Study with Two Feature Modeling Tools.	This paper presents the results of an exploratory study aiming to support SPL engineers choosing the feature modelling tool that best fits their needs. This exploratory study compares and analyses two feature modelling tools, namely FeatureIDE and SPLOT, based on data from 56 participants that used the analyzed tools. The authors performed a four-dimension qualitative analysis with respect to common functionalities provided by feature modeling tools, these are: (i) Feature Model Editor, (ii) Automated Analysis of Feature Models, (iii) Product Configuration, and (iv) Tool Notation. Issues related to the interface of SPLOT only were observed. FeatureIDE, on the other hand, revealed some constraints	The Exploratory Study is for Two Feature Modeling Tools only (FeatureIDE and SPLOT) and hence the study does not work for other Variability Management tools. Additionally, SPLOT interface related issues when interacting with the tool to save changes.

				when creating feature models.	
11	Prahladavarandan	2018	An elementary theory of product-line variations.	Feature oriented domain analysis (FODA) diagrams are a formalism for modelling the variability in a software product-line, and are used as a tool for managing a product-line and planning its evolution. The study presented an elementary theory of variations in a product-line, leading up to a technique for extracting FODA diagrams from legacy product-lines. The theory is elementary in the sense that it is built using very simple approaches with a dedicated model or file to represent the CK specification can bring several benefits to the modularization and stability of a software product line.	Analysing and extracting such feature diagrams from already existing artifacts of a legacy software product-line were not achieved. The development presented in this paper is elementary in the sense that it is built on very simple mathematical structures and makes minimal assumptions on the structure of a product-line. Features of FODA tool are reconstructed from just simple mathematical structures.

DISCUSSION, AND CONCLUSION

A: Results:

This work is aimed at identifying or to answer two questions as mentioned in the RQs section. The questions are answered based on the deployment of Systematic Literature Review (SLR) steps. To simplify the work assessment and avoid research biasness the total no of sourced data are reviewed by 5 different analyzers, each with a researcher code as contained in table 5.

The total number of 568 source data was obtained from a number of data sources. Refer to table 2 for the list of digital and manual data sources. This is achieved by employing a designed search strategy, which is applicable to all digital libraries. The designed search strategy is the Search Strategy section.

A review guidelines (SLR review steps) quality assessment questions has been designed and shared among the review team members to aide answer the RQs (RQ1 and RQ2). These guidelines include the ICs, ECs, Quality assessment questions, the search strategy for sourcing data from the selected primary data sources.

After getting answer to RQ1 ie to identify Tools developed to handle variability by applying the designed search strategy on the data sources, the sourced papers (568) data was then analyzed

using ICs and ECs which dropped down to 143 screening out 425 that are non SPL related study criteria. After repeated and careful analysis, the study selection reduced to 28. These 28 turned to be our primary studies that met all the laid down ICs and ECs criteria. At this step we were able to answer RQ1. Having met the research objective one which is identifying the Tools developed to handle variability in SPL, we move further to answer and achieve our RQ2 and research objective two which is identifying the features or characteristics of the selected primary studies.
Table 5 contained the list of selected primary studies, their title, year of publication and paper ID. The paper ID is a combination of the analyzer code (the first two letters) and the number paper appears in the list of primary studies. Figure 2 depicts a bar chart presentation of the selected papers over time as well as the number of papers in each year. Table 6 shows distribution of primary studies based on outlets, journal/conference.
Figure 3 depicts a bar chart presentation and distinction of the papers on outlets of publication. Figure 2 relates the distribution of primary studies over time having 2020 and 2023 with the highest number of relevant studies in the selected primary data source.
Table 7 presents the list of works and their features/characteristics (strengths and weaknesses). This table is deployed to simply present the RQ2 and research objective ie to identify the features/characteristics of the identified tools in each research.

Conclusion

This systematic literature review provides a comprehensive overview of tools and techniques for handling variability in Software Product Lines. The findings offer valuable insights for researchers and practitioners, contributing to the advancement of SPL research and practices. The work can serve as basis to practitioners and researchers to conduct further researches on the available weaknesses in the identified tools in the table 7. The table tends to summarize the gaps in the studied tools for easier comprehension.
The work was able to identify 47 tools which best skipped all the laid down ICs/ECs in the Inclusion and Exclusion criteria.
Interested researchers can further extend their interest in tackling the identified gaps/weaknesses of any tool of interest. It’s also imperative to further the research questions by tackling other components, features or characteristics of those tools. This work is primarily based to only tackle two RQs; (1) identify the existing tools that handle variability management in SPL and (2) identify their features/characteristics. To achieve this, the work deployed SLR in conducting each paper study.

REFERENCE

Abbas, A., Farah Siddiqui, I., Lee, S. U. J., Kashif Bashir, A., Ejaz, W., & Qureshi, N. M. F. (2018). Multi-objective optimum solutions for IoT-Based feature models of software product line. *IEEE Access*, *6*(8), 12228–12239. https://doi.org/10.1109/ACCESS.2018.2806944

Abbas, M., Jongeling, R., Lindskog, C., Enoiu, E. P., Saadatmand, M., & Sundmark, D. (2020). *Product line adoption in industry*. 1–11. https://doi.org/10.1145/3382025.3414953

Acher, M., Lopez-Herrejon, R. E., & Rabiser, R. (2018). Teaching software product lines: A snapshot of current practices and challenges. *ACM International Conference Proceeding Series*, *1*(1), 249. https://doi.org/10.1145/3233027.3236393

Belarbi, M., & Englebert, V. (2019). Bespoke: A Methodology to design Software Factories. A preliminary approach. *Proceedings - International Conference on Research Challenges in Information Science*, *2019-May*. https://doi.org/10.1109/RCIS.2019.8876968

Castro, T., Teixeira, L., Alves, V., Apel, S., Cordy, M., & Gheyi, R. (2021). A Formal Framework of Software Product Line Analyses. *ACM Transactions on Software Engineering and Methodology*, *30*(3). https://doi.org/10.1145/3442389

De Oliveira, R. P., & De Almeida, E. S. (2016). Evaluating Lehman's Laws of Software Evolution for Software Product Lines. *IEEE Software*, *33*(3), 90–93. https://doi.org/10.1109/MS.2016.78

Eilers, M., Müller, P., & Hitz, S. (2019). Modular product programs. *ACM Transactions on Programming Languages and Systems*, *42*(1). https://doi.org/10.1145/3324783

El-Sharkawy, S., Yamagishi-Eichler, N., & Schmid, K. (2019). Metrics for analyzing variability and its implementation in software product lines: A systematic literature review. *Information and Software Technology*, *106*, 1–30. https://doi.org/10.1016/j.infsof.2018.08.015

Gharsellaoui, H., Maazoun, J., Bouassida, N., Ahmed, S. Ben, & Ben-Abdallah, H. (2021). A Software Product Line Design Based Approach for Real-time Scheduling of Reconfigurable Embedded Systems. *Computers in Human Behavior*, *115*. https://doi.org/10.1016/j.chb.2017.04.026

Hierons, R. M., Li, M., Liu, X., Parejo, J. A., Segura, S., & Yao, X. (2020). Many-objective test suite generation for software product lines. In *ACM Transactions on Software Engineering and Methodology* (Vol. 29, Issue 1). https://doi.org/10.1145/3361146

Ignaim, K., & Fernandes, J. M. (2019). An industrial case study for adopting software product lines in automotive industry an evolution-based approach for software product lines (EVOA-SPL). *ACM International Conference Proceeding Series*, *B*. https://doi.org/10.1145/3307630.3342409

Kahraman, G., & Cleophas, L. (2022). A tool for modeling and analysis of relationships among feature model views. *26th ACM International Systems and Software Product Line Conference, SPLC 2022 - Proceedings*, *B*, 103–109. https://doi.org/10.1145/3503229.3547076

Lima, R., Bezerra, C., & MacHado, I. (2023). A Self-Adaptation Mechanism for Variability Management in Dynamic Software Product Lines. *ACM International Conference Proceeding Series*, 51–60. https://doi.org/10.1145/3622748.3622754

Marchezan, L., Rodrigues, E., Bernardino, M., & Basso, F. P. (2020). PAxSPL: a feature retrieval process for SPL reengineering. *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, *Part F1642*(2019), 2–13. https://doi.org/10.1145/3382025.3414991

Mazanek, S., & Rutetzki, C. (2011). On the importance of model comparison tools for the automatic evaluation of the correctness of model transformations. *ACM International Conference Proceeding Series*, 12–15. https://doi.org/10.1145/2000410.2000413

Ommen, J., & Rock, G. (2015). Visualization of variability in complex development structures. *ACM International Conference Proceeding Series*, *21-23-Janu*, 109–116. https://doi.org/10.1145/2701319.2701324

Preuveneers, D., Heyman, T., Berbers, Y., & Joosen, W. (2016). Feature-based variability management for scalable enterprise applications: Experiences with an E-payment case. *Proceedings of the Annual Hawaii International Conference on System Sciences*, *2016-March*, 5793–5802. https://doi.org/10.1109/HICSS.2016.717

Reinhartz-Berger, I., & Kemelman, M. (2020). Extracting core requirements for software product lines. *Requirements Engineering*, *25*(1), 47–65. https://doi.org/10.1007/s00766-018-0307-0

Rhein, A. Von, Liebig, J., Janker, A., Kästner, C., & Apel, S. (2018). Variability-Aware Static Analysis at Scale. *ACM Transactions on Software Engineering and Methodology*, *27*(4), 1–33. https://doi.org/10.1145/3280986

Sauer, P. C., & Seuring, S. (2023). How to conduct systematic literature reviews in management research: a guide in 6 steps and 14 decisions. In *Review of Managerial Science* (Vol. 17, Issue 5). Springer Berlin Heidelberg. https://doi.org/10.1007/s11846-023-00668-3

Setyautami, M. R. A., & Hähnle, R. (2021). An Architectural Pattern to Realize Multi Software Product Lines in Java. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3442391.3442401

Yadav, H., Chhikara, R., & Kumari, A. C. (2021). A novel hybrid approach for feature selection in software product lines. *Multimedia Tools and Applications*, *80*(4), 4919–4942. https://doi.org/10.1007/s11042-020-09956-6