A Systematic Literature Review of Software Product Line Management Tools

Juliana Alves Pereira[1], Kattiana Constantino[1], Eduardo Figueiredo[1]

[1] Computer Science Department, Federal University of Minas Gerais, Belo Horizonte, Brazil
{juliana.pereira, kattiana, figueiredo}@dcc.ufmg.br

Abstract. Software Product Line (SPL) management is a key activity for software product line engineering. The idea behind SPL management is to focus on artifacts that are shared in order to support software reuse and adaptation. Gains are expected in terms of time to market, consistency across products, costs reduction, better flexibility, and better management of change requirements. In this context, there are many available options of SPL variability management tools. This paper presents and discusses the findings from a Systematic Literature Review (SLR) of SPL management tools. Our research method aimed at analyzing the available literature on SPL management tools and the involved experts in the field. This review provides insights (i) to support companies interested to choose a tool for SPL variability management that best fits their needs; (ii) to point out attributes and requirements relevant to those interested in developing new tools; and (iii) to help the improvement of the tools already available. As a direct result of this SLR, we identify gaps, such as the lack of industrial support during product configuration.

Keywords: Systematic Literature Review. Software Product Lines. Variability Management. Tools.

1 Introduction

The growing need for developing larger and more complex software systems demands better support for reusable software artifacts [46]. In order to address these demands, software product line (SPL) has been increasingly adopted in software industry [15, 60, 52]. SPL is a set of software systems that share a common and variable set of features satisfying the specific needs of a particular market segment [46]. It is built around a set of common software components with points of variability that allow different product configurations [15, 60]. SPL adoption brings significant improvements to the software development process [46, 48]. Experience already shows that SPL can allow companies to realize order-of-magnitude improvements in time to market, cost, productivity, quality, and flexibility [15]. Large companies, such as Hewlett-Packard, Nokia, Motorola, and Dell have already adopted SPL practices [52].

An important concept of an SPL is the feature model. Feature models are used to represent the common and variable features in SPL [30]. A feature represents an increment in functionality or a system property relevant to some stakeholders [30]. It may refer to functional or non-functional requirements, architecture decisions, or

adfa, p. 1, 2011.

design patterns [4]. The potential benefits of SPLs are achieved through a software architecture designed to increase the reuse of features in several SPL products.

In practice, developing an SPL involves feature modeling to represent different viewpoints, sub-systems, or concerns of the software products [5]. Therefore, it is necessary to have tool support to aid the companies during the SPL variability management. Supporting tools provide the companies a guide for the development of SPLs, as well as, a development and maintenance environment of the SPL. However, the choice of one tool that best meets the companies SPL development goals is far from trivial. In particular, this is a critical activity due to a sharp increase in the number of SPL management tools available in the last years. Furthermore, tool support should assist the complete SPL development process, and not just some activities. In this context, this paper contributes with a systematic literature review (SLR) aiming to identify and classify tools that support the SPL management, including the stages from conception until products derivation.

SLR is one secondary study method that has gotten much attention lately in software engineering [32]. An SLR reduces researchers' bias through pre-defined data forms and criteria that limit the room for interpretation [64]. Briefly, an SLR goes through existing primary reports, reviews them in-depth, and describes their methodology and results [45]. Therefore, our SLR represents a significant step forward in the state-of-the-art by deeply examining many relevant tools for feature modeling and management. The general propose of this study is to give a visual summary, by categorizing existing tools. Results are extracted from searching for evidences in journals and conference proceedings since 2000, by the fact that visibility provided by SPL in recent years has produced a higher concentration of research [37].

This study is based on a systematic search of publications from various data sources and it follows a pre-defined protocol during the whole process. Our results contributes specifically with relevant information (i) to support practitioners choosing appropriate tools that best fits their needs in a specific context of SPL, (ii) to point out attributes and requirements relevant to those interested in developing new tools, and (iii) to help with the improvement/extension of existing tools. We expect that both researchers and practitioners can benefit from the results of this SLR.

The rest of this paper is organized as follow. Section 2 presents the steps carried out in this SLR, presenting the research protocol, conduction, and process of data extraction. Section 3 presents the summary and analysis of observed data in the selected studies, answering three research questions. Section 4 presents the threats to validity related to this SLR and how they were addressed prior of the study to minimize their impact. Finally, Section 5 concludes the paper and provides directions for future work.

2 Literature Systematic Review (SLR)

This study has been carried out according to the guideline for SLR proposed by Kitchenham and Charters [32]. We have adapted and applied such techniques in order to identify and classify existing tools to support SPL management. The guidelines [32] are structured according a three-step process for Planning, Conducting, and Reporting the review. Fig. 1 depicts an overview of our research process comprising each step and its stages sequence. The execution of the overall process involves iteration, feedback, and refinement of the defined process [32]. The SLR steps and protocol are detailed below.

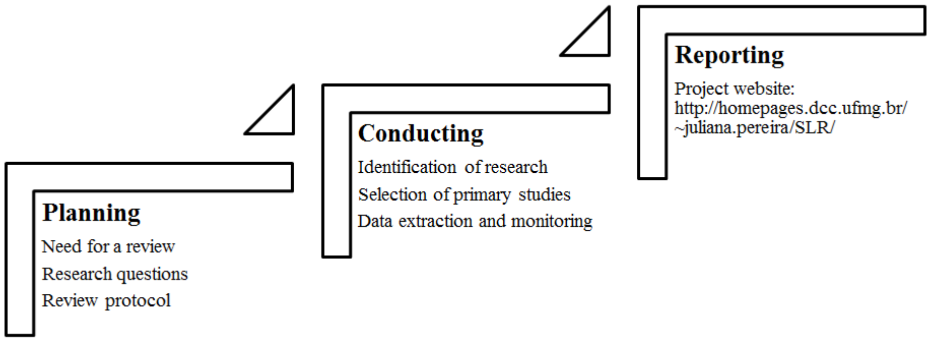

Fig. 1. Overview research process.

2.1 Planning the Review

This step has the goal of developing a protocol that specifies the plan that the SLR will follow to identify, assess, and collate evidence [32]. To plan an SLR includes several actions:

Identification of the need for a review. The need for an SLR originates from increase in the number of SPL management tools made available. In this context, the choice of one tool that best fits the researchers and practitioners needs in a specific context of SPL is far from trivial. Therefore, this SLR aims to give an overall picture of the tools available in the literature for SPL management, in order to find out how they support the variability management process.

Specifying the research questions. The focus of the research questions (RQs) is to identify and analyze SPL management tools. The RQs are formulated with the help of Population, Intervention, Comparison, Outcome, and Context (PICOC) criteria [28]. The population is composed of studies that cite SPL management tools. The intervention includes search for indications that the SPL conception, development, and maintenance can be fully supported. The comparison goal is to find evidences on the characteristics and functionalities that differ each tool. The outcome of this study represents how tools support the variability management process. It supports practitioners and researchers not only choosing appropriate tools, but also in the development of new tools and improvement of existing tools. The context is within the domains of SPL management with a focus on tools and their application domain.

The main research question (RQ) of this study is: **How do the available tools for SPL support the variability management process?** To answer this question, a set of sub-questions was derived from the main RQ to identify and analyze the relevant literature discussions. More specifically, we investigate three RQs:

- *RQ1*. **How many SPL management tools have been cited in the literature since 2000?**

- *RQ2*. **What are the main characteristics of the tools?**
- *RQ3*. **What are the main functionalities of the tools?**

To address *RQ1*, we identified tools that are cited in the literature since 2000. With respect to *RQ2*, we identify the types of tools that have been developments and where the tools were developed. Through these descriptions, it is possible to map the current adoption of tools for SPL. Finally, with respect to *RQ3*, we are concerned with how the tool supports each stage of the development process, since the SPL conception, to its development, and maintenance. In particular, what SPL topics, contributions, and novelty they constitute to. Hence, it is possible to map how tools are supporting the SPL management process and if the process is not fully supported, i.e., if there are gaps in the existing tools, or if there is a need of developing functionalities that are not available in existing tools. Therefore, this SLR was conducted to identify, analyze, and interpret all available evidence related to specific RQs.

Developing a review protocol. We conducted an SLR in journals and conferences proceedings published from January 1st 2000 to December 31th 2013, by the fact that visibility provided by SPL in recent years has produced a higher concentration of research [37]. Three researchers were involved in this process and all of them continuously discussed and refined the RQs, search strings, inclusion, and exclusion criteria.

2.2 Conducting the Review

Conducting the review means executing the protocol planned in the previous phase. The conduction of this SLR includes several actions:

Identification of research. Based on the RQs, the keywords were extracted and used to search the primary study sources. The search string used was constructed using the strategy proposed by Chen et al. [13]. Note that, we check preliminarily the keywords in all relevant papers already known [35, 40, 55]. Based on this strategy, this study relies on the following search string.

("tool") AND ("product line" OR "product family" OR "system family") AND ("management" OR "modeling" OR "configuration")

The term "*tool*" ensures that the research is conducted in order to find tools. In addition, the terms *"product line"*, *"product family"* or *"system family"* restrict the search by SPL tools. Finally, the terms *"management"*, *"modeling"* or *"configuration"* refers to the main functions for the development of SPL tools. Note that management is a generic term that includes both the modeling management and the configuration management in SPL. The primary studies were identified by applying the search strings to three scientific databases, namely ACM Digital Library[1], IEEE Xplore[2], and ScienceDirect[3]. These libraries were chosen because they are some of the most relevant ones in the software engineering literature [58]. The search was performed using the specific syntax of each database and considering only the title,

---

1 http://dl.acm.org/
2 http://ieeexplore.ieee.org/
3 http://www.sciencedirect.com/

abstract, and keywords. The results in each digital library are detailed in the Web supplementary material [2]. In addition to the search in digital libraries, the references of the primary studies were also read in order to identify other relevant primary studies (this technique is called "snowballing") [64].

Selection of primary studies. The basis for the selection of primary studies is the inclusion and exclusion criteria [32]. The following two inclusion criteria (IC) were used to include studies that are relevant to answer the RQs.

- *IC1*. The publications should be "journal" or "conference" and only papers written in English were considered.
- *IC2*. We included only primary studies that present tools to support the SPL management process (including one or more phases of the development cycle and maintenance of SPLs, from conception until products derivation). Therefore, the title, abstract, and keywords should explicitly mentions that the focus of the paper contributes with tools to support SPL variability management.

The following three exclusion criteria (EC) were used to exclude studies that we do not consider relevant to answer the RQs.

- *EC1*. We exclude technical reports presenting lessons learned, theses/dissertations, studies that describe events, and studies that are indexes or programming.
- *EC2*. We exclude duplicate papers. If a primary study is published in more than one venue, for example, if a conference paper is extended in a journal version, only the latter instance should be counted as a primary study. The journal version is preferred since it is expected to be the most complete report.
- *EC3*. Papers that do not focus on SPL management tools. Approaches, methods, and other techniques for SPL by itself should be excluded.

After papers inclusion, during the tools selection, we apply the following two EC.

- *EC4*. Tools that are currently discontinued.
- *EC5*. Tools without executable and/or documentation describing its functionalities available. Moreover, the tools with written documentation that does not have usable description about their functionalities were excluded because it would not be possible to understand how the tool works.

Data extraction and monitoring. As in the study of Petersen et al. [45], the reviewers first read abstracts and look for keywords and concepts that reflect the contribution of the paper. When abstracts are insufficient to allow meaningful keywords to be chosen, reviewers choose to also scan the introduction or conclusion sections of the paper. As in the study of Kitchenham et al. [32], each journal and conference paper was reviewed by one of three different researchers (i.e. Pereira, Constantino and Figueiredo). Pereira coordinated the allocation of researchers to tasks based on the availability of each researcher and their ability to access the specific journals and conference proceedings. The researcher responsible for searching the specific journal or conference applied the detailed inclusion and exclusion criteria to the relevant papers. When the reviewers decided that a paper is not relevant, they provided a short rationale why paper should not be included in the study (for instance, because the paper does not cite a tool that support the SPL management process). In addition, another researcher checked any papers included and excluded at this stage. This step was done in order to check that all relevant papers were selected.

Once the list of primary studies is decided, the data from the tools cited by the papers are extracted. The phase of data extraction aims to summarize the data from the selected studies for further analysis. All available tool documentation served as data sources, such as tutorials, technical reports, industry magazines, dissertations/theses, websites, as well as the communication with authors (e.g., emails exchanged). The data extracted from each tool selected were: (i) date of data extraction; (ii) references to primary studies; (iii) tool name; (iv) main references of the tool; (v) release year; (vi) website tool (if available); (vii) main tools characteristics (graphical user interface, prototype, online, plugin, free for use, open-source, user guide, example solutions, and where the tool was developed); and (viii) main functionalities of each tool.

The data extraction phase involved all authors of this paper. We use Google's search engine to collect the data from the tools. The researcher Pereira extracted the data and the other researchers checked the extraction. Note that, during the data extraction the exclusion criteria (*EC4* and *EC5*) were verified. As in Kitchenham [32] when there was a disagreement, we discussed the issues until we reach an agreement. In this step, we used an Excel table to document the data extraction process. It was done in order to assess the quality of the extraction procedure. The data extraction can be found in the Web supplementary material [2].

2.3 Reporting the Review

The reporting step follows to publish the detailed results in the project website [2] and to write this paper. Its goal is to make it clear to others how the search was, and how they can find the same results. The project website also provides detailed information about the results and the search protocol.

3 Results and Analyses

The primary studies identified by the inclusion criteria were selected individually and summarized in the results. After the selection and data extraction of the tools cited by these papers, we discuss the answers to our RQs.

3.1 SPL Management Tools

Table 1 summarizes the number of papers found in our SLR. Note that, for this analysis, relevant studies (total included) more than doubled after 2007. In the first stage, after applying the inclusion and exclusion criteria, 46 papers were included and 103 papers were excluded (4 are duplicated papers). In the second stage, for each reference of primary studies included, we analyze the title and we included more 6 papers [1, 6, 25, 26, 29, 36] that present additional tools to support the SPL management process (technique called "snowballing"). This technique was necessary in order to have a more complete set of tools. At the end of the search phase, 52 papers were included. Specifically, we included 16 papers of the ACM Digital Library (from 48 papers returned), 19 papers of IEEE Xplore (from 71 papers returned), and 11 papers of Science Direct (from 30 papers returned).

Table 1. Number searched for years 2000-2013

Year	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	All
Total	1	4	1	3	7	4	6	13	13	9	15	22	30	21	149
Total Selected	0	0	1	1	2	1	3	4	6	2	4	7	6	9	46
Total Snowballing	0	0	0	0	1	0	1	0	1	1	1	0	0	1	6
Total Included	0	0	1	1	3	1	4	4	7	3	5	7	6	10	52

After the papers inclusion process, 60 potentially relevant tools were selected for extracting and analyzing the data. Another search was performed on Google search engine with the particular information of every tool cited by the papers, in order to find out more documentation about these tools. There are two major artifacts analyzed in this review. The first is concerned with the executable, through which the reviewers could test the functionalities of the tool; and the second involves the written documentation found, i.e. websites, tutorials, technical reports, papers, industry magazines and dissertations/theses for data extraction.

During data extraction, 19 relevant tools were excluded as a result of applying the detailed exclusion criteria *EC4* and *EC5*. Two tools was excluded because the project is currently discontinued (*EC4*), and the remaining tools were not possible to characterize how they work due to the lack of information (*EC5*). The 41 tools included are listed in Table 2 in chronological order. Data in this table show that the number of released tools increased in the years 2005, 2007, 2009 and 2012.

3.2 Main Characteristics of the Tools

After a careful selection of documentation about the included tools, we identify their main characteristics. Table 2 shows which characteristics each tool implements. The following characteristics were considered: graphical user interface (GUI), prototype (PRT), online (ONL), plugin (PLG), free for use (FRE), open-source (OPS), user guide available (USG), example solutions available (EXS), and where the tool was developed (DEV). We use "n/a" for information not available. These characteristic was selected inspired by [13, 36, 59].

In this analysis, we identified that only one tool (Pacogen) has no graphical user interface. Most tools are neither prototypes (81%) nor online (83%). Furthermore, we found out that 51% of the tools are plugins. Plugins provide the extension of tools already established and known. Additionally, 49% are free for use and only 34% are open-source projects. However, we highlight that the increased development of free and open source tools could increase the community's interest in adopting the concept of SPL, and promote knowledge sharing and extension. A drawback in this case is that several developers do not make the source code available.

Table 2. Characteristics each SPL management tool supports.

Tool [Ref.]	Year	GUI	PRT	ONL	PLG	FRE	OPS	USG	EXS	DEV
LISA toolkit [25]	2013	●			●	n/a	n/a	n/a	n/a	B
ISMT4SPL [43]	2012	●				n/a	n/a	n/a	n/a	A
Sysiphus-IVMM [56]	2012	●			●	●	●	n/a	n/a	A
VariaMos [39]	2012	●			●	●	n/a	●	n/a	A
VULCAN [36]	2012	●			●	●	●	n/a	n/a	A
Pacogen [27]	2011					●	●	n/a	n/a	A
Invar [19]	2010	●	●	●		●	n/a	n/a	●	A
FMT [35]	2009	●			●	●	n/a	●	n/a	A
Hephaestus [6]	2009	●				●	●	n/a	n/a	A
Hydra [47]	2009	●			●	●	●	●	n/a	B
SPLOT [40]	2009	●		●		●	●	n/a	●	A
S^2T^2 [7]	2009	●			●	●	n/a	n/a	●	B
FeatureMapper [26]	2008	●			●	●	●	●	n/a	A
MoSPL [54]	2008	●	●			n/a	n/a	n/a	n/a	A
VISIT-FC [12]	2008	●	●			n/a	n/a	n/a	n/a	A
DecisionKing [17]	2007	●			●	n/a	n/a	n/a	n/a	B
DOPLER [18]	2007	●			●	n/a	n/a	n/a	●	B
FaMa [3]	2007	●			●	●	●	●	●	A
GenArch [14]	2007	●			●	●	●	n/a	n/a	A
COVAMOF-VS [50]	2006	●		●	●			n/a	n/a	A
REMAP-tool [49]	2006	●	●		●			n/a	●	B
YaM [29]	2006	●		●		n/a	n/a	n/a	n/a	B
AORA [61]	2005	●				●	●	n/a	●	A
DOORS Extension [9]	2005	●	●	●				n/a	n/a	A
FeatureIDE [55]	2005	●			●	●	●	●	●	A
Kumbang [41]	2005	●			●	●	●	●	n/a	B
PLUSS toolkit [21]	2005	●						n/a	n/a	B
VARMOD [62]	2005	●			●	n/a	n/a	●	●	A
XFeature [63]	2005	●			●	●	●	●	●	B
DREAM [42]	2004	●				n/a	n/a	n/a	n/a	A
ASADAL [31]	2003	●				●	n/a	n/a	n/a	A
Pure::Variants [51]	2003	●			●			●	●	I
Captain Feature [11]	2002	●				●	●	●	n/a	A
DECIMAL [16]	2002	●				n/a	n/a	n/a	n/a	A
Odyssey [8]	2002	●				●		●	●	A
GEARS [33]	2001	●			●			n/a	n/a	I
WeCoTin [1]	2000	●	●	●				n/a	n/a	I
Holmes [53]	1999	●	●			n/a	n/a	n/a	n/a	A
DARE [23]	1998	●	●					n/a	n/a	I
Metadoc FM [57]	1998	●		●	●			●	●	I
001 [34]	1993	●						n/a	n/a	A

User guide (USG) and example solutions (EXS) are not available for most tools analyzed. User guide and example solutions are mainly important for users that do not have any kind of previous training before the tool usage, because these documents show how the user can start from. They enable novice users to use the tool according to the planned process and the result to be correctly interpreted. Little documentation available and lack of supporting examples are some of the reasons that may end up discouraging the adoption and wide use of these tools since it would requires users to study and guess the tool behavior. Finally, we extract information about where the tool was developed (last column in Table 2). We found out that 63,4% of the selected tools were developed exclusively in the academic environment (A), while only 12,2% were developed exclusively in the industry (I), and the remaining 24,4% tools were developed in both academic and industrial environments (B).

3.3 Main Functionalities of the Tools

Our next analyses in this section extend an existing classification of research approaches by Lisboa et al. [37], summarized in Table 3. This classification schema present a practical approach for analysis based on a well-defined set of guidelines and metrics. We extend the previous classification [37] by analyzing new functionalities, such as source code generation and support for integration. Based on Table 3, Table 4 shows which functionalities each tool offers support to. The numbers refer to the functionalities described in Table 3 and the columns separate each group of functionalities: *Planning*, *Modeling*, *Validation*, and *Product Configuration and Integration (PCI)*, respectively. The analysis of the results focuses on presenting the frequencies of publications for each functionality. This result facilitates the identification of which functionalities have been emphasized in past research and, thus, to identify gaps and possibilities for future research. In addition, it can help to discover which tool best satisfies the need of a user in a given context.

Table 3. Explanation functionalities evaluated extended from Lisboa et al. [37].

Functionalities	Explanation
Planning	*It is responsible for collecting the data needed to define domain scope*
(1) Pre-analysis documentation	Stores and retrieves the information
(2) Matrix of the domain	It is represented using rows and columns (features and applications)
(3) Scope definition	Identifies the features that should be part of the reuse infrastructure
Modeling	*It represents the domain scope (commonality, variability and constraint)*
(4) Domain representation	Represents the defined scope
(5) Variability	Represents the variability a feature can have (optional, alternative and or)
(6) Mandatory features	Represent the features that will always be in the products
(7) Composition rule	Create restrictions for representing and relating the features
(8) Relationship types	Provides different types of relationships between the features
(9) Feature attributes	Permits the inclusion of specific information for each feature
(10) Source code generator	Responsible for generating source code based on model
Validation	*This group refers to functionalities responsible to validate the domain*
(11) Domain documentation	Provides documentation about the domain
(12) Manage requirements	Provides support for inclusion of requirements or use cases in the tool
(13) Relationship	Relates the existing features of a domain to the requirements
(14) Reports	Generates reports about the information available in the domain
(15) Consistency check	Verifies if the generated domain follows the composition rules created
Product Configuration and Integration	*Product configuration is built from a common set of reusable asset, and Integration allows the interoperability between other applications.*
(16) Product derivation	Identifies the features that belong to a product
(17) Import/export	Provides the function of Import/export from/to other applications

Table 4. Functionalities each SPL management tool supports.

Tool	Planning			Modeling							Validation					PCI	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
LISA toolkit	•		•	•	•	•	•	•	•	•	•	•	•				•
ISMT4SPL	•		•	•	•	•		•		•		•	•		•	•	
Sysiphus-IVMM	•	•	•	•	•	•	•					•	•		•	•	
VariaMos				•	•	•	•							•	•	•	•
VULCAN				•	•	•	•	•	•	•					•	•	•
Pacogen														•	•		•
Invar											•				•	•	•
FMT				•	•	•	•		•						•	•	•
Hephaestus											•	•	•		•	•	•
Hydra				•	•	•	•		•						•	•	•
SPLOT				•	•	•	•		•		•			•	•	•	•
S^2T^2				•	•	•	•							•	•	•	
FeatureMapper				•	•	•	•		•		•	•	•		•	•	•
MoSPL				•	•	•	•								•	•	•
VISIT-FC				•	•	•	•								•	•	
DecisionKing	•		•	•	•	•	•		•		•	•	•		•		•
DOPLER	•		•	•	•	•	•		•		•	•	•		•	•	•
FaMa				•	•	•	•	•	•					•	•	•	•
GenArch				•	•	•	•		•		•				•	•	•
COVAMOF-VS				•	•	•	•								•	•	•
REMAP-tool	•		•	•	•	•	•				•	•	•		•	•	•
YaM			•													•	
AORA	•	•	•	•	•	•	•				•	•	•		•	•	•
DOORS Extens.				•	•	•	•					•	•		•		•
FeatureIDE				•	•	•	•	•		•				•	•	•	•
Kumbang				•	•	•	•		•						•	•	
PLUSS toolkit				•	•	•	•						•	•	•	•	
VARMOD				•	•	•	•		•		•	•	•		•		•
XFeature				•	•	•	•	•	•	•	•				•	•	•
DREAM		•	•	•	•	•		•					•	•			•
ASADAL				•	•	•	•	•		•					•	•	•
Pure::Variants				•	•	•	•	•	•	•	•			•	•	•	•
Captain Feature				•	•	•	•								•	•	
DECIMAL				•	•	•	•								•	•	
Odyssey				•	•	•					•	•	•				
GEARS				•	•	•	•	•		•				•	•	•	•
WeCoTin				•	•	•	•		•						•	•	•
Holmes	•	•	•	•	•	•	•				•				•	•	•
DARE	•	•	•	•	•	•					•						
Metadoc FM				•	•	•	•				•	•	•		•	•	•
001				•	•	•	•	•		•	•			•	•	•	

In Table 4, there is an evident lack of tools to support all stages of SPL development and evolution. In this classification, the majority of the analyzed tools have similar functionalities. Most of the tools available both commercially and freely support the first four functionalities in group *Modeling*, the last functionality in the *Validation* group, and the two functionalities in the *Product Configuration and Integration* group. Therefore, this result identified that the largest gap in the analyzed tools is support the *Planning* group. In the *Planning* group, the identified functionalities were available only in eleven tools, and only four tools implement all functionalities in this group. The results of this systematic review reveal that 73% of the selected tools stress the need for planning support. A much smaller number of tools (27%) describe concrete support. This seems to indicate that, despite increasing interest and importance, planning is not yet the main focus of the product line research community. Particularly, in *Modeling* group, thirty-five tools (85%) support at least four of the functionalities of this group. Regarding the *Validation* group, only one tool (*YaM*) does not support any functionality. Note that seventeen tools (42%) support at least three of the functionalities of this group. In the *Product Configuration and Integration* group, thirty-three tools (81%) support the product derivation functionality, and the import/export from/to other applications functionality is exploited for twenty-nine tools (71%). Integration is a desirable functionality since it allows the interoperability between other applications. The lack of this functionality could hinder the adoption of the tool, as it hampers their integration with other existing tools. Therefore, although the results highlight the lack of tools to fully support SPL management; e.g., much of the analyzed tools do not support the *Planning* group. On the other hand, we found that the tools offer interoperability between other applications. This fact maximizes the reuse within the solution itself and externally, for instance, allowing users to migrate from on technology (e.g., conditional compilation [22]) to another (e.g., feature oriented programming [24]).

4 Threats to Validity

A key issue when performing of the SLR is the validity of the results. Questions we need to answer include: was the study designed and performed in a sound and controlled manner? To which domain can the results generalize? This section presents the different validity threats related to SLR. We presented how the threats were addressed to minimize the likelihood of their realization and impact. We discussed the SLR validity with respect to the four groups of common threats to validity: internal validity, external validity, construct validity, and conclusion validity [64].

External validity. External validity concerns the ability to generalize the results to other environments, such as to industry practices [64]. A major external validity to this study was during the identified primary studies. The search for the tools was conducted in three relevant scientific databases in order to capture as much as possible the available tools and avoid all sorts of bias. However, the quality of search engines could have influenced the completeness of the identified primary studies. That means our search may have missed those studies whose authors would have used other terms to specify the SPL management tool or would not have used the keywords that we used for searches in the title, abstract, and keywords of their papers.

Internal validity. Internal validity concerns the question whether the effect is caused by the independent variables (e.g. reviewers) or by other factors [64]. In this sense, a limitation of this study concerns the reliability. The reliability has been addressed as far as possible by involving three researchers, and by having a protocol which was piloted and hence evaluated. If the study is replicated by another set of researchers, it is possible that some studies that were removed in this review could be included and other studies could be excluded. However, in general we believe that the internal validity of the SLR is high given the use of a systematic procedure, consultation with the researchers in the field, involvement, and discussion between three researchers.

Construct validity. Construct validity reflects to what extent the operational measures that are studied really represent what the researcher have in mind and what is investigated according to the RQs [64]. The three reviewers of this study are researchers in the software engineering field, focused in SPL, and none of the tools was written/developed by us. Therefore, we are not aware of any bias we may have introduced during the analyses. However, from the reviewers perspective, a construct validity threat could be biased judgment. In this study, the decision of which studies to include or to exclude and how to categorize the studies could be biased and thus pose a threat. A possible threat in such review is to exclude some relevant tool. To minimize this threat both the processes of inclusion and exclusion were piloted by the three reviewers. Furthermore, potentially relevant studies that were excluded were documented. Therefore, we believe that we do not have omitted any relevant tool.

Conclusion validity. Threats to conclusion validity are related with issues that affect the ability to draw the correct conclusions from the study [64]. From the reviewers' perspective, a potential threat to conclusion validity is the reliability of the data extraction categories from the tools, since not all information was obvious to answer the RQ and some data had to be interpreted. Therefore, in order to ensure the validity, multiple sources of data were analyzed, i.e. papers, websites, technical reports, industry magazines manuals, and executable. Furthermore, in the event of a disagreement between the two primary reviewers, the third reviewer acted as an arbitrator to ensure agreement was reached.

5 Conclusion and Future Work

In industry, the development of large software systems requires an enormous effort for modeling and configuring multiple products. To face this problem, several tools have been proposed and used for representing and managing the variations of a system. There are many available options of variability management tools. Therefore, this paper aims to identify and evaluate existing tools in the literature to support the SPL management. To this end, we conducted a SLR, where we follow a guide of systematic review proposed by Kitchenham [32]. To substantiate the findings of the paper, the working group has set up a website [2] where interested people can verify detailed results of the SLR.

Although analysis of existing SPL management tools has been performed in previous studies [5, 10, 20, 59, 44], the purpose of these studies in a general way was to facilitate tool selection in the context of SPL. However, they were aimed at studying only tools very specifics or a small group of tools. Our study differs from others by the fact of being an SLR. In addition, with regard to SLR presented by Lisboa et al. [37], the results confirm that our study analyzes a relatively number higher of tool. Finally, our resulted have impact due to the SLR period, search string, processes of inclusion/exclusion and data extraction.

Our SLR identified 41 tools existing in the literature that provide support to at least one SPL management phase. It provides an overview of tools where it is possible to see which characteristics and functionalities each tool implemented. The contribution of this research covers both the academic and the professional segment. In the academic environment, this research helps to highlight the lack of complete tools in this area, identifying gaps in current research to provide a background for new research activities. In the professional context, this research helps to find possible tools that will assist SPL developers or even help them manage existing SPL.

Through the results obtained, it is clear that the little documentation available and the complexity and/or the unavailability of existing support tools are some of the reasons that may end up discouraging the adoption and wide use of these tools in organizations and academia. Therefore, we intend to use our knowledge to conduct courses in both academia and industry, in order to encourage adoption of these tools. In addition, there are still gaps in the complete process support in all the tools we investigated. Although most of the tools offer interoperability between other applications, such gaps in functionality can make it difficult for industry to adopt a tool, because it would hinder the use of several tools and traceability of information among them. Therefore, our review suggests that there are opportunities for the extension of the existing tools. One of the gaps that we identify with the SLR is that manual method for product configuration adopted by the tools may not be sufficient to support industries during SPL managing. The manual process makes product configuration into a complex, time-consuming, and error-prone task. Therefore, we are working on an extension of a tool with the new functionality [38]. Moreover, as future work, we aim to evaluate (i) notations used in each tool; (ii) tools that can be used together in a complementary way; (iii) some criteria to analyze the tools comparatively, for example, GUI complexity and visualization support; and (vi) characteristics and functionalities relevant for the practitioners and researchers.

Acknowledgements

This work was partially supported by CNPq (grant Universal 485907/2013-5) and FAPEMIG (grants APQ-02532-12 and PPM-00382-14).

References

1. Asikainen, T. et al.: Using a Configurator for Modelling and Configuring Software Product Lines based on Feature Models. In: Workshop on Software Variability Management for Product Derivation, Software Product Line Conference (SPLC), pp. 24–35 (2004)
2. A Systematic Literature Review of Software Product Line Management Tools, `http://homepages.dcc.ufmg.br/~juliana.pereira/SLR`
3. Benavides, D. et al.: Fama: Tooling a Framework for the Automated Analysis of Feature Models. In: 1st International Workshop on Variability Modelling of Software Intensive Systems (VaMoS), pp. 129–134 (2007)
4. Bernardo, M. et al.: Architecting Families of Software Systems with Process Algebras. In: ACM Transactions on Software Engineering and Methodology, 11(4):386–426 (2002)
5. Beuche, D. et al.: Variability Management with Feature Models. Journal Science of Computer Programming, 53(3):333–352 (2004)

6. Bonifácio, R. et al.: Hephaestus: A Tool for Managing SPL Variabilities. In: Brazilian Symposium on Components, Architectures and Reuse Software (SBCARS), pp. 26–34 (2009)
7. Botterweck, G. et al.: A Design of a Configurable Feature Model Configurator. In: 3rd International Workshop on Variability Modelling of Software Intensive Systems (VaMoS), pp. 165–168 (2009)
8. Braga, R. et al.: Odyssey: A Reuse Environment based on Domain Models. In: IEEE Symposium on Application-Specific Systems and Software Engineering and Technology (ASSET), pp. 50–57 (1999)
9. Buhne, S. et al.: Modelling Requirements Variability across Product Lines. In: 13th International Conference on Requirements Engineering (RE), pp. 41–50 (2005)
10. Capilla, R. et al.: An Analysis of Variability Modeling and Management Tools for Product Line Development. In: Software and Service Variability Management Workshop - Concepts, Models, and Tools, pp. 32–47 (2007)
11. Captain Feature Tool, `http://sourceforge.net/projects/captainfeature`
12. Cawley, C. et al.: Interactive Visualisation to Support Product Configuration in Software Product Lines. In: 2nd International Workshop on Variability Modeling of Software-Intensive Systems (VaMoS), pp. 7–16 (2008)
13. Chen, L., Babar, M. A.: A Systematic Review of Evaluation of Variability Management Approaches in Software Product Lines. Journal Information and Software Technology, 53(4):344–362 (2011)
14. Cirilo, E. et al.: A product Derivation Tool based on Model-Driven Techniques and Annotations. Journal of Universal Computer Science, 14(8):1344–1367 (2008)
15. Clements, P., Northrop, L.: Software Product Lines: Practices and Patterns. Addison-Wesley (2001)
16. Dehlinger, J. et al.: Decimal and PLFaultCAT: From Product-Line Requirements to Product-Line Member Software Fault Trees. In: 29th International Conference on Software Engineering (ICSE), pp. 49–50 (2007)
17. Dhungana, D. et al.: Decisionking: A Flexible and Extensible Tool for Integrated Variability Modeling. In: 1st International Workshop on Variability Modelling of Software-intensive Systems (VaMoS), pp. 119–128 (2007)
18. Dhungana, D. et al.: The Dopler Meta-Tool for Decision-Oriented Variability Modeling: A Multiple Case Study. Journal Automated Software Engineering, 18(1):77–114 (2011)
19. Dhungana, D. et al.: Integrating Heterogeneous Variability Modeling Approaches with Invar. In: 7th International Workshop on Variability Modelling of Software-intensive Systems (VaMoS), (2013)
20. Djebbi, O. et al.: Industry Survey of Product Lines Management Tools: Requirements, Qualities and Open Issues. In: 15th IEEE International Requirements Engineering Conference (IREC), pp. 301–306 (2007)
21. Eriksson, M. et al.: The Pluss Toolkit: Extending Telelogic Doors and IBM-Rational Rose to Support Product Line Use Case Modeling. In: 20th International Conference on Automated Software Engineering (ASE), pp. 300–304 (2005)
22. Figueiredo, E. et al.: Evolving Software Product Lines with Aspects: An Empirical Study on Design Stability. In: 30th International Conf. on Soft. Eng. (ICSE), pp. 261-270 (2008).
23. Frakes, W. B. et al.: Dare-cots: A Domain Analysis Support tool. In: International Conference of the Chilean Computer Science Society, pp. 73–77 (1997)
24. Gaia, F. et al.: A Quantitative and Qualitative Assessment of Aspectual Feature Modules for Evolving Software Product Lines. Science of Computer Programming (SCP), pp. 1–24 (2014)

25. Groher, I., Weinreich, R.: Supporting Variability Management in Architecture Design and Implementation. In: 46th Hawaii International Conference on System Sciences (HICSS), pp. 4995–5004 (2013)
26. Heidenreich, F. et al.: FeatureMapper: Mapping Features to Models. In: International Conference on Software Engineering (ICSE), pp. 943–944 (2008)
27. Hervieu, A. et al.: Pacogen: Automatic Generation of Pairwise Test Configurations from feature models. In: 22nd International Symposium on Software Reliability Engineering (ISSRE), pp. 120–129 (2011)
28. Higgins J. et al.: Cochrane Handbook for Systematic Reviews of Interventions, vol. 5, Wiley Online Library (2008)
29. Jain, A., Biesiadecki, J.: Yam: A Framework for Rapid Software Development. In: 2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT), pp. 182–194 (2006)
30. Kang, K. et al.: Feature-Oriented Domain Analysis (FODA) Feasibility Study. Available at: `http://www.sei.cmu.edu/ reports/90tr021.pdf/` (1990)
31. Kim, K. et al.: Asadal: A Tool System for Co-Development of Software and Test Environment based on Product Line Engineering. In: 28th International Conference on Software Engineering (ICSE), pp. 783–786 (2006)
32. Kitchenham, B. et al.: Systematic Literature Reviews in Software Engineering: A systematic Literature Review. Journal Information and Software Technology, 51(1):7–15 (2009)
33. Krueger, C.: Biglever Software Gears and the 3-tiered SPL Methodology. In: 22nd Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA), pp. 844–845 (2007)
34. Krut, J.R.W.: Integrating 001 Tool Support in the Feature-Oriented Domain Analysis methodology. Technical Report, Software Engineering Institute (SEI). Paper available at: `http://repository.cmu.edu/cgi/viewcontent.cgi?article=1166&context=sei` (1993)
35. Laguna, M., Hernández, C.: A Software Product Line Approach for Ecommerce Systems. In: 7th International Conference on e-Business Engineering (ICEBE), pp. 230–235 (2010)
36. Lee H. et al.: VULCAN: Architecture-Model-Based Workbench for Product Line Engineering. In: 16th International Software Product Line Conference (SPLC), pp. 260–264 (2012)
37. Lisboa, L.B. et al.: A Systematic Review of Domain Analysis Tools. Journal Information and Software Technology, 52(1):1–13 (2010)
38. Machado, L. et al.: SPLConfig: Product Configuration in Software Product Line. In: Brazilian Congress on Software (CBSoft), Tools Session, pp. 1–8 (2014)
39. Mazo, R. et al.: Variamos: A Tool for Product Line Driven Systems Engineering with a Constraint Based Approach. In: 24th International Conference on Advanced Information Systems Engineering (CAiSE), pp. 1–8 (2012)
40. Mendonça, M. et al.: S.P.L.O.T.: Software Product Lines Online Tools. In: 24th Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA), pp. 761–762 (2009)
41. Myllärniemi, V. et al.: Kumbang tools. In: 11th Software Product Line Conference (SPLC), pp. 135–136 (2007)
42. Park, J. et al.: Dream: Domain Requirement Asset Manager in Product Lines. In: International Symposium on Future Software Technology (ISFST) (2004)
43. Park, K. et al.: An Integrated Software Management Tool for Adopting Software Product Lines. In: 11th International Conference on Computation and Information Science (ICIS), pp. 553–558 (2012)

44. Pereira, J. et al.: Software Variability Management: An Exploratory Study with Two Feature Modeling Tools. In: Brazilian Symposium on Software Components, Architectures and Reuse (SBCARS), v. 1. p. 1–10 (2013)
45. Petersen, K. et al.: Systematic Mapping Studies in Software Engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering. (EASE), pp. 68–77 (2008)
46. Pohl, K. et al.: Software Product Line Engineering: Foundations, Principles and Techniques. Springer-Verlag (2005)
47. Salazar, J. R.: Herramienta para el Modelado y Configuración de Modelos de Características. PhD Thesis, Dpto. Lenguajes y Ciencias de la Comp., Universidad de Málaga (2009)
48. Santos, A. et al.: Test-based SPL Extraction: An Exploratory Study. In: 28th ACM Symposium on Applied Computing (SAC), Software Engineering Track, pp. 1031–1036 (2013).
49. Schmid, K. et al.: Requirements Management for Product Lines: Extending Professional Tools. In: 10th International Software Product Line Conference (SPLC), pp. 113–122 (2006)
50. Sinnema M. et al.: The COVAMOF Derivation Process. In: 9th International Conference on Reuse of Off-the-Shelf Components (ICSR), pp. 101–114 (2006)
51. Spinczyk, O. and Beuche, D.: Modeling and Building Software Product Lines with Eclipse. In: 19th conference on Object-oriented programming systems, languages, and applications (OOPSLA), pp. 18–19 (2004)
52. SPL Hall of Fame, http://splc.net/fame.html
53. Succi, G. et al.: Holmes: An Intelligent System to Support Software Product Line Development. In: 23rd International Conference on Software Engineering (ICSE), pp. 829–830 (2001)
54. Thao, C. et al.: Software Configuration Management for Product Derivation in Software Product Families. In: 15th International Conference and Workshop on the Engineering of Computer Based Systems (ECBS), pp. 265–274 (2008)
55. Thüm, T. et al.: FeatureIDE: An Extensible Framework for Feature-Oriented Software Development. Journal Science of Computer Programming, 79(0):70–85 (2014)
56. Thurimella, A.K., Bruegge, B.: Issue-Based Variability Management Information and Software Technology. Journal Information and Soft. Technology, 54(9):933–950 (2012)
57. Thurimella, A. K. and Janzen, D.: Metadoc Feature Modeler: A Plug-in for IBM Rational Doors. In: International Software Product Line Conference (SPLC), pp. 313–322 (2011)
58. Travassos, G.H., Biolchini, J.: Systematic Review Applied to Software Engineering. In: Brazilian Symposium on Software Engineering (SBES), Tutorials, p. 436 (2007)
59. Unphon, H.: A Comparison of Variability Modeling and Configuration Tools for Product Line Architecture. Paper available at: http://www.itu.dk/people/unphon/ technical_notes/CVC_v2008-06-30.pdf (2008)
60. Van der Linden, F. et al.: Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering, Springer-Verlag (2007)
61. Varela P. et al.: Aspect-Oriented Analysis for Software Product Lines Requirements Engineering. In Proceedings of the 2011 ACM Symposium on Applied Computing (2011)
62. Varmod-Prime Tool, http://paluno.uni-due.de/en/varmod-prime. Software Systems Engineering Research Group/ University of Duisburg-Essen.
63. XFeature Modeling Tool, http://www.pnp-software.com/XFeature
64. Wohlin, C. et al.: Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers (2012)