Information and Software Technology 124 (2020) 106293

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Software product line applied to the internet of things: A systematic literature review

Ricardo Theis Geraldi, Sheila Reinehr, Andreia Malucelli*

Graduate Program in Computer Science (PPGIa), Pontifícia Universidade Católica do Paraná (PUCPR), Curitiba, Brazil

ARTICLE INFO

Keywords:
Internet of things
Software product line
Variability management
Product family engineering
Families of systems

ABSTRACT

Context: Internet of Things (IoT) is a promising paradigm due to the growing number of devices that may be connected, defined as "things". Managing these "things" is still considered a challenge. One way to overcome this challenge may be by adopting the software product line (SPL) paradigm and the variability management (VM) activity. SPL engineering consists of mechanisms that provide identification, representation, and traceability, which may be helpful to "things" management supported by VM organizational and technical activities.
Objective: This research aims to investigate how SPL engineering has been applied along with the IoT paradigm, as well as how VM is being carried out.
Method: A systematic literature review (SLR) was conducted considering papers available until March 2019. This systematic review identified 1039 papers. After eliminating the duplicated titles and the ones not related to the review, 112 papers remained. The number of papers was narrowed to 56 after applying the exclusion criteria.
Results: The results provide evidence on the diversity of proposed SPLs used to specify approaches for managing IoT systems. However, most SPLs and research developed for IoT lack a systematic and detailed specification to ensure their quality, as well as tailoring guidelines for further use.

1. Introduction

Among the systematically non-opportunistic strategies of reuse artifacts in software engineering, the software product line (SPL) paradigm is gaining attention and has been consolidated in the academy and industry [1,2]. SPL differs from the traditional software development process because it considers a family of products instead of each product individually [2] Several organizations[1] such as Boeing, Bosch Group, Cummins, Nokia, Philips, Siemens, Toshiba, and the US Naval Research Laboratory have adopted the SPL paradigm to obtain better quality and productivity in the software development cycle to encompass a set of specific products for a particularly competitive market [3,4]. Some benefits obtained by these organizations that may be emphasized are quality improvement, reduction of development costs, production time, risks, and improved delivery quality of a given product to the customer to meet their needs and expectations, both short-term and long-term [5].

Such benefits are attributed to the correct planning and application of technical and organizational management activities. One of the main SPL activities is variability management (VM) [7,8], which provides the concepts of customization and derivation of products using approaches developed in the literature [7–9]. Variability consists of how a product can be differentiated from others of the same family based on the common and variables features in an SPL [1–7].

We believe that such SPL engineering characteristics supported by VM can be a way to meet the heterogeneous device and systems management gaps of the Internet of Things (IoT) paradigm, which is also in agreement with the literature. Owing to the growth of "things" in specific IoT systems in complex and dynamic smart environments or networks, the common and variables features of VM can be adopted to manage them. Basically, IoT defines connections based on multiple objects ("things") as devices (e.g., sensors, radio frequency identification (RFID), and microcontrollers as Arduino and Raspberry Pi) and equipment (e.g., home refrigerators or medical hardware), which encompass characteristics such as identification, detection, and interaction [10,11].

Lee and Lee [12] believe that managing these things can also impact new costs and investments. In addition to these impacts on IoT, difficulties may arise in applying SPL engineering in IoT systems: (i) investing resources (time, people, and financial) to understand the reuse strategy in adapting SPL engineering in the IoT; (ii) the time spent learning the basic concepts of SPL engineering; (iii) understand how to map the features of IoT systems and represent them in specific SPL models (feature model or multi-perspective models); (iv) which processes and types of IoT artifacts can be represented using SPL; (v) understand SPL tools

* Corresponding author.
E-mail addresses: ricardo.geraldi@ppgia.pucpr.br (R.T. Geraldi), sheila.reinehr@pucpr.br (S. Reinehr), malu@ppgia.pucpr.br (A. Malucelli).

[1] SPLC Hall of Fame site: http://splc.net/hall-of-fame/.

https://doi.org/10.1016/j.infsof.2020.106293
Received 18 August 2019; Received in revised form 23 February 2020; Accepted 24 February 2020
Available online 21 March 2020
0950-5849/

for managing things; and (vi) how to configure or reconfigure functionalities and/or things (e.g., sensors and actuators) to derive custom IoT products. However, we believe that by adopting SPL in an organized and systematic manner, it is possible to improve IoT systems productivity and management related to the development of quality and customized products, reducing long-term difficulties.

A way to minimize these impacts and difficulties can be by the definition of architectural elements supported by approaches to the development of IoT systems [13] and the possible adaptation of the SPL engineering framework of Pohl et al. [5]. In this sense, Buyya and Dastjerdi [14] pointed out the importance of a reference architecture to organize and understand this complex scenario to guarantee the evolution of IoT. For instance, the IoT-i project in Europe defined strategies and techniques for developing IoT solutions according to interest in different sectors. Thus, a second project called IoT-A[2] specified an IoT architecture reference model using the building blocks and requirements collected in the IoT-i project [14].

Motivated by these perceptions, this study aims to investigate how SPL engineering has been applied in the IoT paradigm, as well as how VM is carried out, and identify gaps in the primary studies considering their associations with further investigations to answer three research questions (RQs):

- RQ1: How are software product lines (SPL) being applied in the context of the Internet of Things (IoT) systems? This RQ aims to analyze and identify smart application areas/domains in which SPLs are applied in IoT; in which types of IoT systems one or several SPLs have been adopted or adapted; and which SPLs and IoT systems have been applied in the academic and/or industrial environment.
- RQ2: How is the variability management (VM) of SPL carried out in IoT systems? This RQ aims to explore VM activity applied to IoT systems; how VM activity is adopted to IoT devices or systems; the existence of processes using the VM activity; the existence of models (e.g., feature model and/or IoT-based feature model - same meaning with another nomenclature); VM resolution time types; and the existence of VM mechanisms, relationships, constraints to perform VM.
- RQ3: Which approaches, frameworks or platforms use SPL in IoT systems? This RQ aims to investigate and identify possible approaches, frameworks, or platforms that apply SPL concepts across different IoT systems domains; and presents their technology and application in the academy and/or industrial environments.

These RQs investigate how SPL is applied in the context of the IoT paradigm, considering how the VM of things is carried out, and the possible existence of approaches, frameworks, or platforms in this scenario. These RQs have emerged from the need to search for possibilities or solutions to better manage such interconnected things. Based on these RQs, the five main contributions of this study are:

- A systematic literature review (SLR) was conducted and retrieved studies up to 12 March 2019.
- 1039 studies were analyzed based on defined search strings and filtered through inclusion and exclusion criteria, allowing us to reach a final set of 56 selected primary studies.
- Relevant primary studies have been identified, allowing to discuss new research opportunities between SPL and IoT.
- The problem space has not been solved in IoT systems management, but positive evidence is present in most selected primary studies.
- No secondary studies such as systematic literature review or systematic mapping related to SPL and IoT were identified at the time of carrying out this SLR. Thus, we expect to provide an initial body of knowledge for scientific and industrial domains.

The remainder of this paper is structured as follows: Section 2 summarizes the background of the study. Section 3 describes the protocol and the research method used for conducting this SLR. Section 4 presents the results obtained as well as gaps identified in the studies related to the SPL and IoT paradigms. Section 5 reports the answers and discussions of the research questions based on the final set of selected primary studies. Section 6 presents the main threats to the validity of this review. Finally, we present conclusions and future work in Section 7.

[2] IoT-A Project site: https://cordis.europa.eu/project/rcn/95713_en.html.

2. Background

Software product line is a paradigm that allows the customization of software product families through the sharing of common and variable features based on central artifacts (core assets) [2]. The SPL engineering framework proposed by Pohl et al. [5] defines abstractions regarding the reuse level of the artifacts of an SPL for product customization through two processes: domain engineering (DE) and application engineering (AE). In DE, the identification of common and variables aspects in SPL is defined. AE encompasses the reuse of DE artifacts for the derivation of specific applications by solving or choosing their variabilities.

In this scenario, organizational and technical activities such as variability management are essential to the success of the SPL paradigm in product customization [15]. Variability refers to how the products of an SPL can be differentiated from each other by means of the constraints defined in variation points and variants [16]. The VM is performed by selecting one or more variants associated with a variation point, to be solved for specific products considering variant constraints [1–4]. A variation point can occur at different levels of abstraction of generic artifacts from an SPL, providing a solution to the variabilities from such variants in different locations. A variant can be chosen from the elements for the resolution of such variation points or variabilities. The constraints define the relationships that are allowed among variants, considering the resolution of the variation points or variabilities [2].

In addition, systematic reviews of Chen and Ali Babar's [6] and Galster et al.'s [7] works present several approaches to VM. According to this literature, one common way of representing an SPL, including VM, can be through the feature-oriented domain analysis (FODA) method or generic feature models according to Kang et al. [17]. The FODA method was explored and adapted in a few studies involving the IoT paradigm. For instance, Ortiz et al. [18] modeled an *indoor smart environmental and guidance systems* product family based on FODA. Yang and Zhang [19] applied VM to feature modeling using FODA aims to extract the features in *ventilator embedded SPL*. Marimuthu and Chandrasekaran [20] proposed a domain analysis framework using FODA to understand *Energy-Aware Self-Adaptive Systems*.

As mentioned, some studies encompass researches with a focus on SPL, VM, and IoT paradigms (e.g., using FODA). Thus, the adoption of SPL engineering mechanisms can be a way to manage specific smart objects (e.g., ventilator control and energy-aware mobile) or things (e.g., sensors and actuators) of the IoT paradigm owing to the complexity of interconnected things in IoT systems [18–20]. To understand this scenario, the main definitions of IoT are described as follows.

Miorandi et al. [21] presents an investigation of the IoT paradigm state-of-the-art, encompasses three main characteristics — identification, communication, and interaction between heterogeneous smart objects (e.g., sensors or actuators or RFID with unique name and address, physical feature, and a thing with trigger and communication actions) or things (e.g., sensors and actuators) digitally or physically in smart environments. In the identification characteristic, the interconnected things are considered specific. The communication characteristic is described to enable the detection and actions of the interaction of such things. The interaction characteristic allows the communication between interconnected things in different networks affected by a dynamic complex context, which requires to be managed. This context is influenced by device heterogeneity, scalability, ubiquitous wireless data, energy consumption, localization and tracking features, self-organization, semantic management, and embedded security [21].

In Buyya and Dastjerdi's work [14] among the essential concepts presented regarding IoT, the things are directly related to interaction in the physical world by means of the Internet or heterogeneous smart devices (e.g., several devices and humans), involving concerns related to the connectivity of sensors in different IoT environments. The things (e.g., several smart devices, including sensors and human-in-the-loop) in IoT can also be defined as smart objects in smart networks (e.g., communication infrastructure with standards to allow physically objects communication) that interact and communicate with a specific market, involving humans to provide a better quality of life.

Whitmore et al. [11] present future directions and discuss trends and topics for IoT considering the characteristics and concepts defined by Miorandi et al. [21] and Buyya and Dastjerdi [14]. Thus, Whitmore et al. [11] define six categories (technology, applications, challenges, business models, future directions, and overview) to classify the studies of literature in the context of IoT, presenting the control of change technology and the way in which automation is performed in IoT [11]. Furthermore, Gubbi et al. [13] describe future directions in the integration of users and application areas encompassing sensing, analytics, and visualization tools of the things (e.g., RFID and wireless sensors) in IoT at any instant.

3. Systematic literature review protocol

This SLR was conducted through a strategy consisting of six main stages: (i) definition of research questions (Section 3.1), (ii) research method for primary studies (Section 3.2), (iii) definition of data sources, keywords, and search query (Section 3.3), (iv) selection of studies by applying the inclusion and exclusion criteria (Section 3.4), (v) study quality assessment (Section 3.5), (vi) extraction and aggregation of data (Section 3.6), and (vii) the classification scheme of selected primary studies (Section 3.7).

3.1. Research questions

The research questions (RQs) defined are as follows:

- RQ1: How are software product lines (SPL) being applied in the context of the Internet of Things (IoT) systems?
- RQ2: How is the variability management (VM) of SPL carried out in IoT systems?
- RQ3: Which approaches, frameworks or platforms use SPL in IoT systems?

The main purpose of these RQs is to retrieve, identify, and analyze scientific papers on SPL applied for IoT in the academic and industrial environments. The RQ1 aims to explore in the literature smart application areas/domains applied in IoT systems and how SPLs have been adapted in this context. The RQ2 and RQ3 help to complement the answers for RQ1 providing additional information on how SPL is been applied in IoT considering VM activity, and which approaches/frameworks/platforms use SPL in IoT.

RQ1 presents a relation between SPL and IoT application areas through a qualitative analysis including (i) gaps synthesis for each primary study to answer RQ1; (ii) identify if IoT problem space has been solved in the SPL domain and application engineering; and (iii) identify academic and smart application areas/domains in which SPL in IoT occurs.

RQ2 includes studies gaps synthesis (Section 4.2) and a qualitative analysis to investigate (i) VM activity in IoT systems or things; (ii) which management process apply VM activity; (iii) which representation models (e.g., feature model and/or IoT-based feature model - same representation with another name) apply VM activity; (iv) VM resolution time types (e.g., runtime), mechanisms (e.g., variation points), relationships (e.g., mandatory, AND, OR), and constraints (e.g., dependency, requires).

RQ3 presents approaches, frameworks, and/or platforms that apply SPL concepts to different IoT system domains, and presents the developed technology as well as its application in the academy and/or industry. We believe this RQ provides an initial overview of what has been developed in both areas to support researchers in conducting new relevant researches.

3.2. Research method

The applied research method is based on guidelines proposed by Kitchenham and Charters [22], Kitchenham et al. [23], and Petersen et al. [24]. Kitchenham and Charters [22] defined detailed guidelines to conduct systematic literature reviews rigorously. This guideline addresses three main stages involving planning, conducting, and reporting the review. Kitchenham and Charters [22] guideline was created based on concepts of medical researches, social science, and evidence-based practices. In another work, Kitchenham et al. [23] encompass several evidence-based practices for conducting and reporting systematic reviews in software engineering researches as planning, searching, study selection, extracting study data, analysis, synthesis, and report. Petersen et al. [24] present similar stages to conduct secondary studies focusing on systematic mapping studies.

Complementary studies were used as a basis to provide directions in our research method stages [25–28]. Thus, the research method is divided into stages called filters, activities, and tasks. The main stage of this research method consists of filters. A filter is defined as a phase that allows planning and executing activities and tasks of retrieval, inclusion and exclusion, and selection of relevant studies to this systematic review. A filter contains inputs and outputs related to activities allowing perform tasks.

The activities are inputs for tasks. Tasks are outputs performed based on the inputs of these activities. An activity receives inputs as outputs of tasks. These outputs help to define the next inputs for the following tasks, which will be performed throughout the process of this research method.

Each filter receives as inputs a set of different activities and tasks. Filter #1 activities are "Select databases" and "Define keywords". In the activity, "Select databases" was defined as digital databases as a first item before conducting this systematic review. The "Select databases" activity contains the selected databases for the "Data source list" task. The activity "Define keywords" receives as input the task "Data source list", used as a list of digital databases in which studies will be retrieved. Thus, the "Define keywords" defines the input terms that allow the execution of the task "Search strings". The output of this task contains search strings adapted for performing and retrieving studies on six search mechanisms (e.g., IEEE Xplore).

Filter #2 receives as input retrieved studies based on search strings executed on the digital databases defined in Filter #1. After that, in Filter #2 the inclusion and exclusion criteria are applied to all studies retrieved in Filter #1. Filter #2 has one activity and one task. The "Review abstracts and conflicts" activity is the first output of Filter #2 that allows us to review and establish which studies will be selected based on the inclusion and exclusion criteria. The output of the "Review abstracts and conflicts" activity is the input to the "Updated list of papers" task. This task allows creating as output an updated list of studies that will be fully read in the main input for Filter #3.

In Filter #3 the final set of selected studies will be fully read according to the input task received from Filter #2. This input contains the final list of updated studies that are the input for the first Filter #3 activity called "Extract and aggregate data". In this activity, the metadata and information of the selected studies are organized. The next "Classify papers" activity receives as input the studies metadata and aims to classify the studies based on the method presented in Section 3.7. After the execution of these last activities the final task of the Filter #3 called "Results and discussion of papers" aims to present the research questions answered and discussed as final results for this SLR.

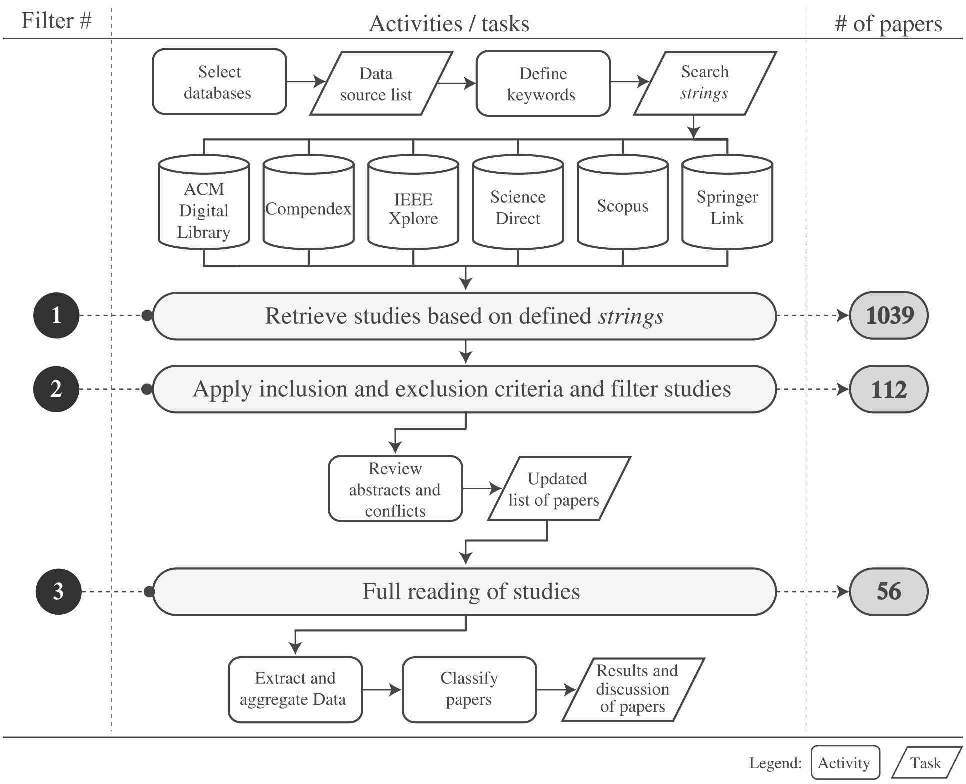

Fig. 1. Research method stages (based on Geraldi and OliveiraJr [25], Mota Silveira Neto et al. [26], Barney et al. [27], and Mohabbati et al. [28]).

Fig. 1 presents the research method stages with the number of filtered studies and the detailed activities/tasks that allowed the selection of such studies in each filter.

The following procedures detail the research stages (Fig. 1) through activities/tasks contained in the filters: (i) selection of digital databases (Section 3.3); (ii) keyword definition, as well as, a search string and generic query adapted in each data source to retrieve the studies (Section 3.3); and (iii) application of defined search strings in data sources; (iv) duplicated and conflicted studies were reviewed to generate an updated list of primary studies (Section 3.4); (v) study quality assessment (Section 3.5), and (vi) extraction and aggregation of data (Section 3.6) present the data related to selected primary studies that were extracted and synthesized as well as classified (Section 3.7) corresponding to the paradigms of SPL and IoT. The final set of primary studies were analyzed and classified.

3.3. Data sources, keywords, and search strings

After the search (Fig. 1), 1039 studies based on Filter #1 were retrieved. A preliminary selection of such studies was performed by reading title, abstract, introduction, and conclusion sections. Thus, through the application of the inclusion and exclusion criteria, 112 studies were obtained according to Filter #2. Thus, 56 studies were selected for full reading based on RQ1, RQ2, and RQ3, according to Filter #3. Table 1 presents the filters associated with the number of studies in each data source.

The following digital databases were used to search and filter studies in this SLR: ACM Digital Library, Compendex, IEEE Xplore, ScienceDirect, Scopus, and Springer Link. The next step was to define keywords to specify a generic search query (see Table 2). The query, as well as the keywords, were adapted iteratively to search in each selected digital database.

Table 1
Number of studies by data sources and filters.

Data Sources	Filter #1	Filter #2	Filter #3
ACM Digital Library	11	5	3
Compendex	31	3	2
IEEE Xplore	289	34	17
ScienceDirect	111	13	4
Scopus	382	31	17
Springer Link	215	26	13
Total	1039	112	56
Duplicated studies	140	–	–

3.4. Inclusion and exclusion criteria

We present the inclusion and exclusion criteria to support the selection of the relevant studies in the scope of this SLR. Thus, we also defined the following inclusion and exclusion criteria for each RQ.

3.4.1. Inclusion criteria

We established the following six Inclusion criteria (Ic) to this SLR:

- Ic1: Studies published until 12 March 2019;
- Ic2: Consider relevant secondary studies retrieved as systematic literature reviews or mapping studies that could contain relevant primary studies to this SLR;
- Ic3: Studies that address both SPL engineering concepts and/or models to IoT systems.
- Analyze studies and answer to RQs to include:
- Ic4: *RQ1*. Verify the existence and the application or adaptation of SPL in IoT systems;
- Ic5: *RQ2*. Analyze the VM carried out and discussed in the primary studies; and
- Ic6: *RQ3*. Identify which approaches, frameworks, and/or platforms are proposed in the primary studies considering this context.

Table 2
Keywords and strings defined.

Generic search query
("internet of things" OR "iot") AND ("software product line" OR "SPL" OR "product family engineering" OR "pfe" OR "software product family" OR "families of systems" OR "variability management")

3.4.2. Exclusion criteria

The following further eight Exclusion criteria (Ic8) were established:

- Ec1: Exclude the primary studies that do not answer the RQs;
- Ec2: Studies that do not define concepts and/or models regarding SPL engineering, including feature models, UML models, and/or other models;
- Ec3: Studies not encompassing SPL engineering applied in IoT systems;
- Ec4: Duplicated studies and/or those retrieved in more than one data source;
- Ec5: Philosophical studies;
- Ec6: Studies reported or presented as keynote;
- Ec7: Studies in languages other than English; and
- Ec8: Unavailable studies at the time of the search (e.g., URL unavailable) in digital databases.

3.5. Study quality assessment

Several research types of primary studies were selected, encompassing experiments, case studies, expert opinion studies, and others (motivational examples, motivational running examples, or prototype). To classify and evaluate these studies, we adopted a quality instrument based on a generic checklist based on Kitchenham and Brereton [29]. This instrument aims to evaluate the study's quality from the perspective of questions with different quality criteria. The main criteria defined to evaluate the quality of these studies involve the reporting of each study through a screening process considering the research type, research method, and the objectives of these studies. The following instrument questions (*checklist*) were fully adopted as defined by Kitchenham and Brereton [29].

1 "Is the paper based on research (or is it a discussion paper based on expert opinion)? Yes/No."
2 "What research method was used: Experiment, Quasi-Experiment, Lessons learned, Case study, Opinion Survey, Tertiary Study, Other (specify)? Note: This is to be based on our reading of the paper, not the method claimed by the author of the paper."
3 "Is there a clear statement of the aims of the study? Yes/ Partly/No. Score as 1, 0.5, 0. Interpolation is permitted."

The quality data of each study were identified and are shown in Appendix C. This process occurred in conjunction with the data extraction and aggregation from such studies. Second Kitchenham and Brereton [29] we also decided not to exclude studies based on the score defined in the third question of this study quality assessment. The score helped in identifying the clarity of the objectives of each study but did not help in identifying and classifying the research method type adopted or adapted in each study. Thus, we observed the following situations:

- In criteria 1, only three studies were identified related to experts' opinions with lessons learned. These studies helped to understand and corroborate with a possible application to SPL and IoT, for instance, in smart homes or context-aware systems variability.
- In criteria 2, most studies conducted small evaluations. Some studies used the terms "Case study" or "Experiment" indiscriminately. However, fourteen case studies were conducted in industry or "real world" environments. Thirteen studies were classified as "Experiment". In this evaluation type, the authors presented an initial prototype evaluated by changing parameters in design-time or runtime. We attempt to classify the remaining of the studies as "Other", which mostly provided motivational or executable examples supported by some kind of early prototype developed.
- In criteria 3, few studies have not clearly defined their objectives. Most selected studies have been published in high-quality journals or conferences, so we believe this is a consequence of the scientific rigor established by these publication venues.

3.6. Data extraction and aggregation

This section presents the extraction stage, summarizing the main metadata related to the final set of selected primary studies, including: (i) Data source: see list in Table 1, (ii) Authors, (iii) Title, (iv) Publication year: the studies were retrieved from the data sources up to 12 March 2019, (v) Publication type: conference papers and journal articles, and (vi) Publication venue (acronym). The final metadata set of the selected primary studies (Filter #3) may be viewed in Table 6.

In addition to the extraction stage, considering the aggregation of data and the application of the inclusion and exclusion criteria, the following data of the selected primary studies were identified (Section 4) and synthesized (Section 5): (i) application and adaptation of SPL in IoT, (ii) how VM is carried out, and (iii) approaches platform, and/or frameworks.

The Zotero[3] and Mendeley[4] tools were used to organize and filter the bibliographic references supported by Microsoft Excel spreadsheets.[5] Thus, the supplementary material of this SLR was created; it contained the URL of selected studies by source (as well as *.bibtex* files). The query string is available at https://doi.org/10.5281/zenodo.3576170.

3.7. Classification scheme

The selected studies were classified considering six categories based on Wieringa et al.'s work [30] (i) Validation research, which focused on academic research, encompassing experiments in the laboratory or simulations. (ii) Evaluation research, which involved industrial research to verify problems in practice, e.g., evaluate the behavior of a new approach in the industry. (iii) Solution proposal, which focused on describing solutions to solve a specific problem by means of new approaches/techniques. (iv) Philosophical papers, which explored new ways or concepts in researches. (v) Opinion papers, which presented opinion aspects (positive or negative) regarding certain approaches or case studies. (vi) Experience papers, which reported the personal experience of the researcher in practice.

4. Results

4.1. Overview of selected primary studies

In the following sections, we present demographic data, the distribution, and the data sources with respect to the data extracted from each selected primary study to provide the analyses of frequency, venue, and trends of these publications.

Analyzing the results presented in Fig. 1 complemented by Fig. 2, it was possible to observe that 56 studies were selected, among which

[3] Zotero site: https://www.zotero.org.
[4] Mendeley site: https://www.mendeley.com.
[5] Microsoft Excel site: https://microsoft.office.com/excel.

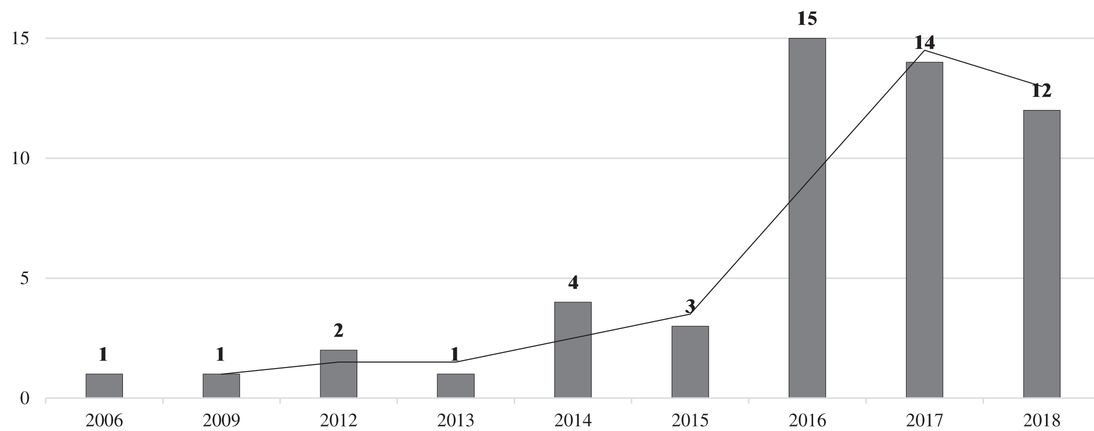

Fig. 2. Demographic distribution of selected primary studies over the years 2006–2018 (Filter #3).

three secondary studies indirectly contributed to answering questions RQ1, RQ2, and RQ3. Wohlin [31] presents guidelines to help researches to identify relevant studies in systematic literature studies using the snowballing approach. Backward and forward snowballing provides means to look title and abstract of the papers referenced and cited in secondary studies of the literature [31]. Thus, such search type is related to the snowballing approach [31] applied in Krupitzer et al. [32], Sánchez Guinea et al. [33] and DeFranco et al. [34]. One primary study was selected from each one of these secondary studies.

Krupitzer et al.'s [32] survey was performed in the context of taxonomies for self-adaptive applications; Sánchez Guinea et al.'s [33] SLR identified primary studies with different approaches to specifying the development cycle phases for ubiquitous applications, and DeFranco et al.'s [34] systematic mapping study encompassed the non-functional requirements based on medical device software. Therefore, we decided to include these secondary studies owing to the identification and selection of the following primary studies using the snowballing approach: (i) Lee and Kang's [35] study was selected based on the survey of Krupitzer et al.'s work [32] (ii) Baresi et al.'s [36] study was selected based on the SLR of Sánchez Guinea et al. [33], and (iii) Pessoa et al.'s [37] study was selected from the systematic mapping of DeFranco et al.'s work [34]. Thus, from the 56 selected studies, the above-mentioned secondary studies were removed. Therefore, 53 selected primary studies were analyzed in the following sections.

4.1.1. Demographics and distribution of selected studies

Fig. 2 illustrates the demographic data showing that the majority of studies were conducted between 2016 and 2017, which generated the largest number of published studies (29.55%). This evidence shows that research on the IoT and SPL paradigms may be on the rise. However, observing the data for the year 2018, it was found that the number of selected studies (12.23%) remained slightly below average when compared to the years 2016 and 2017 up to 12 March 2019, the closing date of this SLR.

Analyzing the studies distribution by publication venues in Table 3, we observe that the Software Product Line Conference (SPLC) provides the largest number of studies (5), followed by the Asia-Pacific Software Engineering Conference (APSEC) (2), International Symposium on Business Modeling and Software Design (BMSD) (2), and the IEEE International Conference on Internet of Things (iThings) (2). Considering journals, we observe that IEEE Access (2), IEEE Computer (2), Information and Software Technology (INFSOF) (2), and MDPI Sensors (2) contain an equal number of studies (2). The 53 primary studies selected appeared in 42 different publication venues, including 31 conferences and 11 journals.

In relation to the distribution of selected primary studies by data source through the analysis of Fig. 3, Tables 1, and 6 we observe that IEEE Xplore has the largest number of studies (17.32%), including 13 studies published in conferences and 4 in journals. A similar number of studies appeared in Scopus (17.32%), including 9 studies published in conferences and 8 in journals. Next in the sequence is Springer Link (12.22%) followed by a smaller number of but relevant studies that appeared in ACM Digital Library (3.6%) followed by ScienceDirect (2.4%) and Compendex (2.4%).

4.1.2. Classification method applied in selected studies

Regarding the classification of the selected primary studies defined in Section 3.7, Figs. 4 and 5 present the classification of the type of research (Filter #3) according to Wieringa et al.'s [30] method.

Analyzing Fig. 4, validation research (19.36%), evaluation research (15.28%), and solution proposal (16.30%) occurred more often within 2016–2018. Few experience papers (3.6%) appeared distributed over the years. In addition, Fig. 5 illustrates the research-type classification considering the number of studies published by the data source. We observe in Figs. 4 and 5 that validation research and solution proposals are being explored by the academy, favoring the initial scientific maturity to contribute to the SPL and IoT paradigms. The evaluation research type has a significant number of relevant studies in the industry with approaches evaluated in practice.

4.2. Studies research topics

In this section, we summarize and categorize the main research topics by gaps identified in each selected primary study. Ten categories of research topics emerged based on the analysis of gaps in the studies. Our analysis shows directions for further investigations aiming to propose possible solutions for IoT systems. All categories are represented graphically with our respective gaps (underlined) that may be traced in the following paragraphs. Fig. 6 presents all categories.

4.2.1. Variability management/modeling

Fig. 7 represents the gaps of this category and their subcategories: 1.1 Combine Variability and 1.2 Dynamic VM and SPL Reconfigurations for IoT Systems.

VM activity is applied in the development and management phases of IoT systems and/or devices in different domains (e.g., health and transportation), according to Venčkauskas et al. [38], Sharif et al. [39], and Ayala et al. [40]. We believe that there is a lack of studies applying VM in other phases of IoT and it could be explored in new researches.

Venčkauskas et al. [38] identified as the main gap in the management of different devices in IoT systems in the area of personalized health monitoring (PHM). We suppose the development of PHM is dependent and still, needs VM approaches due to the number of variations (e.g., devices and applications).

Table 3
Distribution by publication venue and type of selected primary studies.

Publication venue	# Times	Acronym	Publication type
Software Product Line Conference	5	SPLC	Conference
Asia-Pacific Software Engineering Conference	2	APSEC	Conference
International Symposium on Business Modeling and Software Design	2	BMSD	Conference
IEEE International Conference on Internet of Things	2	iThings	Conference
First International EAI Conference on Emerging Technologies for Developing Countries	1	EAI/AFRICATEK	Conference
European Conference on Modelling Foundations and Applications	1	ECMFA	Conference
European Conference on Software Architecture	1	ECSA	Conference
Energy and Sustainability in Small Developing Economies	1	ES2DE	Conference
European Conference on Service-Oriented and Cloud Computing	1	ESOCC	Conference
IEEE International Conference on Future Internet of Things and Cloud	1	FiCloud	Conference
International Conference on Fuzzy Systems and Knowledge Discovery	1	FSKD	Conference
International Conference on Information and Software Technologies	1	ICIST	Conference
International Conference on Service-Oriented Computing	1	ICSOC	Conference
International Student Project Conference	1	ICT-ISPC	Conference
International Conference Integrated Formal Methods	1	IFM	Conference
International Conference on Innovative Computing Technology	1	INTECH	Conference
Internet of Things. IoT Infrastructures: Second International Summit	1	IoT 360º	Conference
International Conference on Information Reuse and Integration	1	IRI	Conference
IEEE International Symposium on Systems Engineering	1	ISSE	Conference
CEUR Workshop Proceedings	1	CEUR	Conference
German Conference on Multiagent System Technologies	1	MATES	Conference
International Conference on Mechatronic and Embedded Systems and Applications	1	MESA	Conference
International Conference on Model-Driven Engineering and Software Development	1	MODELSWARD	Conference
Trends in Practical Applications of Scalable Multi-Agent Systems	1	PAAMS	Conference
International Conference on Services Computing	1	SCC	Conference
Euromicro Conference on Software Engineering and Advanced Applications	1	SEAA	Conference
International Symposium on Software Engineering for Adaptive and Self-Managing Systems	1	SEAMS	Conference
International Conference on Ubiquitous Computing and Ambient Intelligence	1	UCAml	Conference
International Workshop on Variability Modelling of Software-Intensive Systems	1	VaMoS	Conference
Workshop on Social, Human, and Economic Aspects of Software	1	WASHES	Conference
Working IEEE/IFIP Conference on Software Architecture	1	WICSA	Conference
IEEE Access	2	–	Journal
IEEE Computer	2	–	Journal
Information and Software Technology	2	INFSOF	Journal
MDPI Sensors	2	–	Journal
Computer Science and Information Systems	1	ComSIS	Journal
IEEE Software	1	–	Journal
Journal of Theoretical and Applied Information Technology	1	JATIT	Journal
Security and Communication Networks	1	SEC	Journal
MDPI Symmetry	1	–	Journal
Springer Precision Agriculture	1	–	Journal
Telkomnika	1	–	Journal

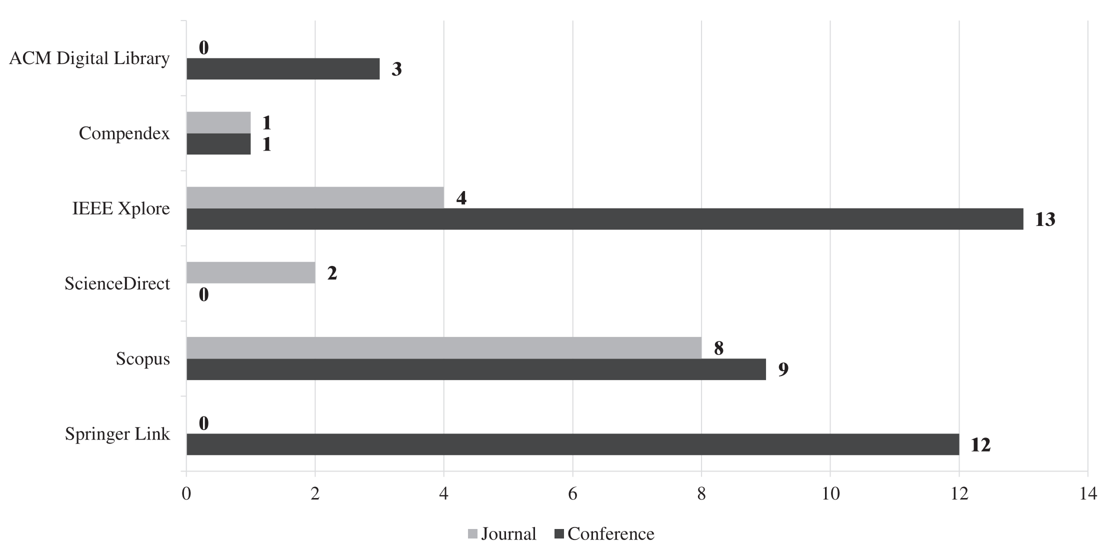

Fig. 3. Distribution of selected primary studies by data source in conferences and journals (Filter #3).

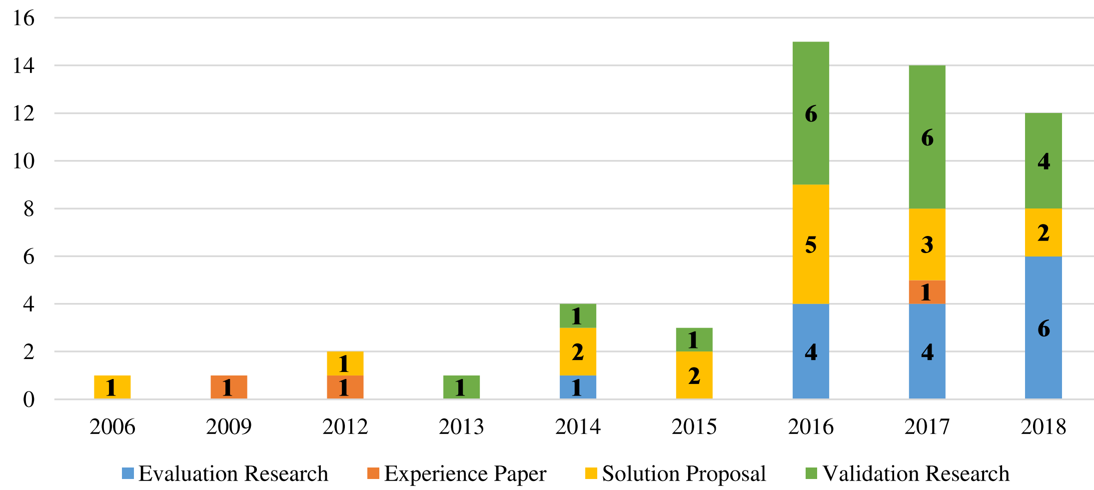

Fig. 4. Research-type classification of selected primary studies (Filter #3).

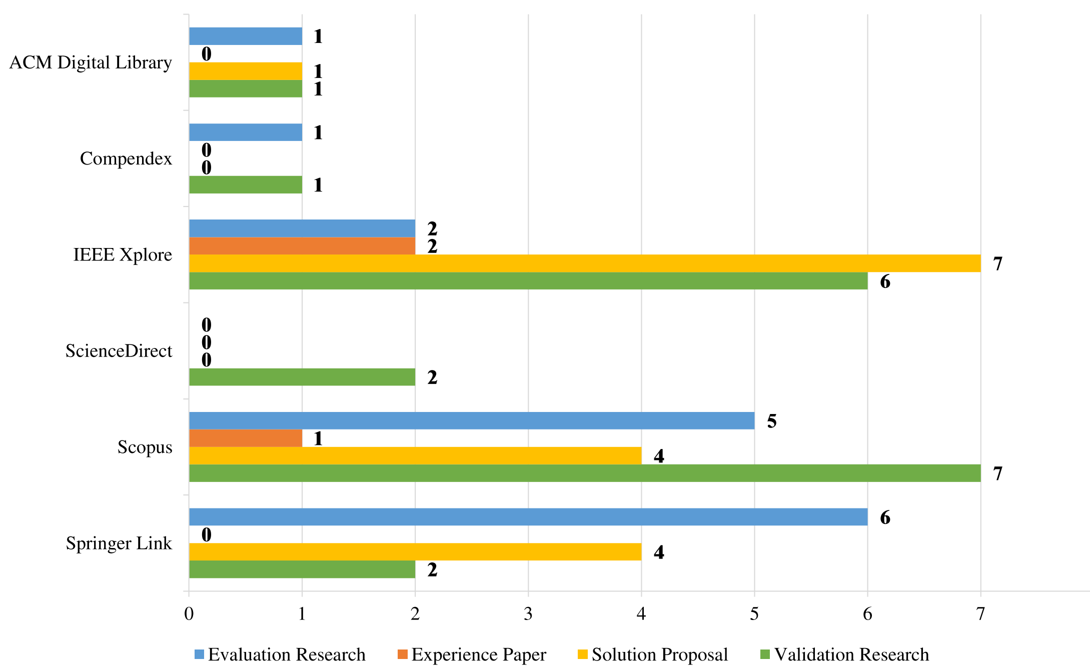

Fig. 5. Research-type classification related to the number of selected primary studies by data source (Filter #3).

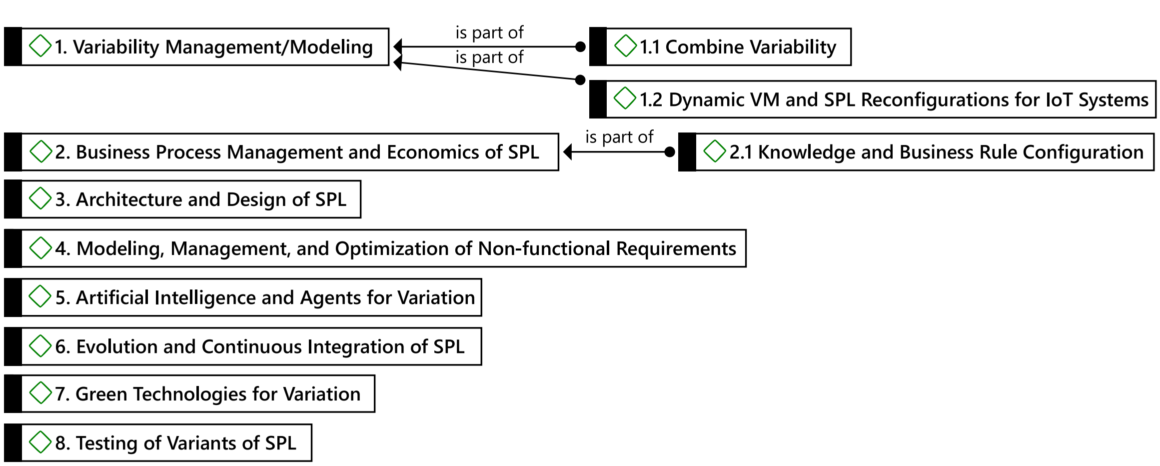

Fig. 6. Gaps categorized by research topics from selected primary studies.

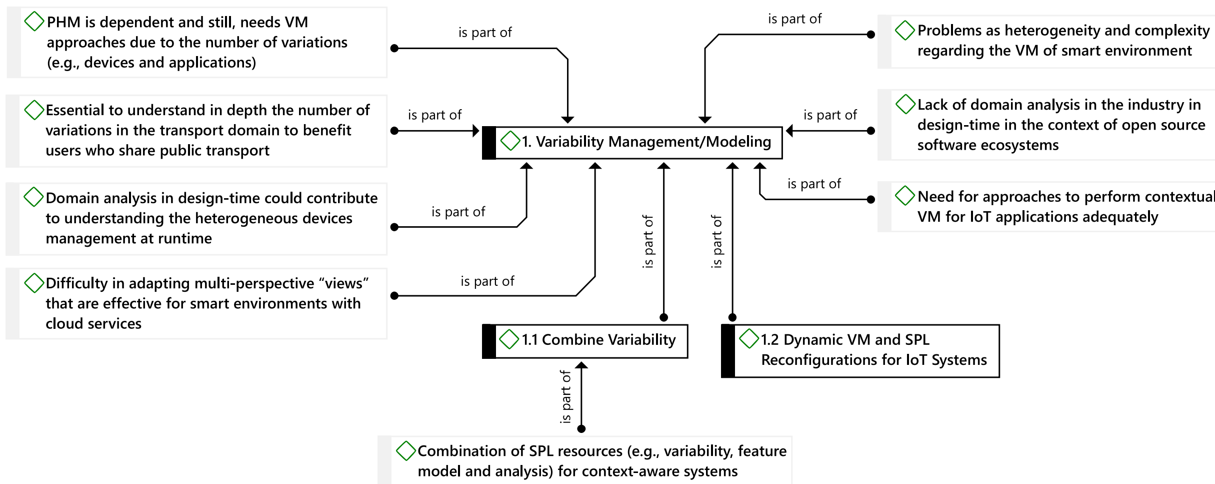

Fig. 7. Gaps categorized by "1. Variability Management/Modeling" research topics.

Sharif et al. [39] presented as a gap the difficulty of the students from the University of Malaysia on detection and tracking of University Bus location in Real-Time by means of a mobile application. Sharif et al. [39] applied VM activity to customize this mobile application. Based on Sharif et al. [39] we consider the correct application of VM activity may help to minimize the lack of management approaches for the transport domain. However, it is essential to understand in depth the number of variations in the transport domain to benefit users who share public transport.

Ayala et al. [40] presented as the main gap in the development of IoT systems owing to heterogeneous devices management. A CVL process was developed in Ayala et al. [40] to try to manage device reconfigurations at runtime. However, we believe that domain analysis in design-time could contribute to understanding the heterogeneous devices management at runtime.

Murguzur et al. [41] addressed gaps in the VM activity of smart environments for IoT applications based on cloud services. Murguzur et al. [41] developed a multi-perspective process containing three variability models. The variability model called the "resolution model" does not have the resolution in design-time and guidelines to assist in adapting the multi-perspective process and understanding the domain in different smart environments. Thus, we think that there a difficulty in adapting multi-perspective "views" that are effective for smart environments with cloud services.

Tekkalmaz et al. [42] addressed a gap as a VM involving problems in the smart environment creation in IoT through an engineering suite. These authors developed a lifecycle supported by a tool suite developed for smart environments management. The proposal by Tekkalmaz et al. [42] directs the responsibility of software engineers to define the functionalities of the smart environment without any kind of script with guidelines for VM activity. We consider problems as heterogeneity and complexity regarding the VM of smart environment features may not be solved by just one tool and by lifecycle.

Tomlein and Grønbæk [43] presented as a gap a VM runtime capabilities modeling of IoT systems in embedded open source software ecosystems. The capabilities were modeled in the industry in runtime using the VM mechanism. We identified that such capacity modeling does not encompass the lack of domain analysis in the industry in design-time in the context of open-source software ecosystems.

There is a lack in the application of VM in systems or contextual approaches in the IoT scenario [44,45] Abbas et al. [44] addressed as a gap, the difficulties in the contextual VM of different devices, environments, and specific users using the XML-based feature model. Mens et al. [45] addressed as the main gap the activation/deactivation of features in contextual VM of context-aware systems of mobile applications, which may occur in web applications, adaptive software (e.g., vehicles), and requirements reconfiguration. We highlighted the need for approaches to perform contextual VM for IoT applications adequately considering features variations, devices diversity, and their adaptations in smart environments.

4.2.1.1. Combine variability. This subcategory of Variability Management/Modeling presents the possibility of combine VM activity with other techniques or approaches existing in the literature in context-aware systems. Such a combination is explored by only two studies of Amja et al. [46,47]. Amja et al. [46] discussed gaps to combine variability, relational concept analysis (RCA)-based model, and a feature model for modeling and development of context-aware systems. Amja et al. [47] investigated gaps in combining relational concept analysis (RCA), descriptive logic (DL), and feature models in context-aware VM for the development of different IoT applications. According to Amja et al. [46,47] studies, we observed that few studies address the combination of SPL resources (e.g., variability, feature model and analysis) for context-aware systems.

4.2.1.2. Dynamic VM and SPL reconfigurations for IOT systems. This subcategory of Variability Management/Modeling is represented in Fig. 8.

The dynamic VM activity is adopted in the reconfiguration of IoT systems in adaptive systems [48] context-aware systems (e.g., automotive sector) [49,50] and robotized systems [51].

McGee and McGregor [48] explored gaps in adapting dynamic VM of adaptive systems in the context of the safety-critical system. Such systems may be designed and analyzed through the architecture analysis & design language (AADL). We consider that such adaptation may be performed at runtime, we identified a lack of empirical studies that present the adaptation of VM activity at design-time for safety-critical systems.

Mauro et al. [49] and Røst et al. [50] presented gaps in the feature model considering dynamic SPL context-aware reconfigurations focusing on the automotive sector. Such studies attempt to solve the gaps through a HyVar project with a solution to variability management in IoT combining domain-specific variability language (DSVL), as well as, toolchain. A hybrid approach to VM named HyVarRec was proposed by Mauro et al. [49] using the car dynamic SPL feature model example to demonstrate the reconfiguration engine for variabilities in feature models. Observing such studies, there is a lack of a guide with systematic guidelines for the adoption of context-aware reconfigurations in several domains beyond the automotive sector.

Heikkilä et al. [51] observed gaps in the reconfiguration of robotized systems at runtime. Thus, a conceptual project was proposed supported

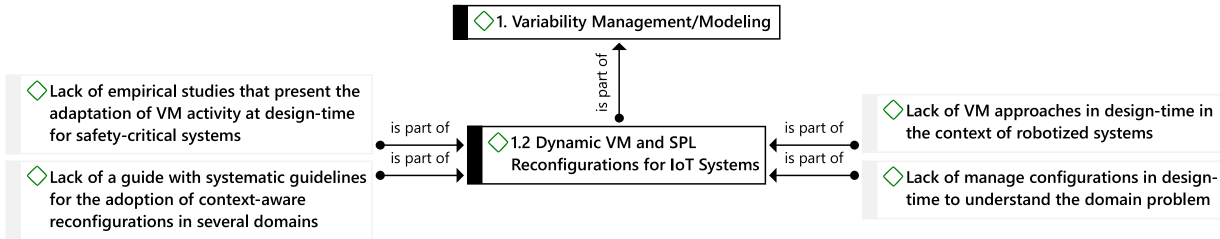

Fig. 8. Gaps categorized by "1.2 Dynamic VM and SPL Reconfigurations for IoT Systems" research topics.

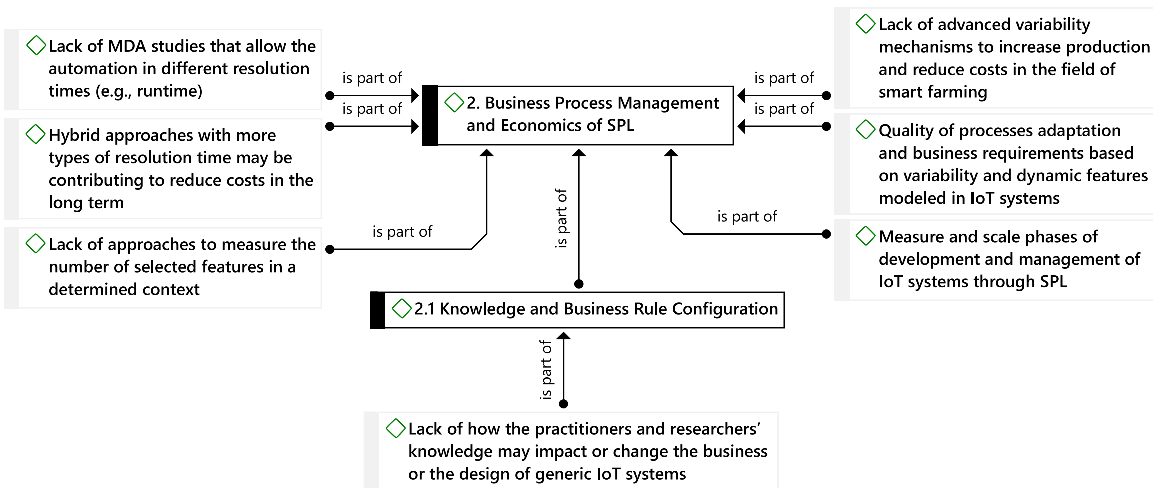

Fig. 9. Gaps categorized by "2. Business Process Management and Economics of SPL" research topics.

by an approach based on SPL that performs the VM in the reconfiguration of robotized systems through a reusable data model from derived configurations. We believe a lack of VM approaches in design-time in the context of robotized systems.

Dynamic VM reconfigurations are carried out generically in IoT systems [52] and in the context of wireless sensor-actuator networks (WSAN) [18]. Achtaich et al. [52] addressed the need to manage dynamic IoT system configurations since the existence of complex environments and different devices may be monitored, controlled, and reconfigured in runtime specific times through dynamic SPL to IoT systems. Ortiz et al. [18] presented gaps at the time of deployment-related to changes and dynamic reconfigurations in features model in product families of sensor network named wireless sensor-actuator network (WSAN). In both studies, we find a lack of manage configurations in design-time to understand the domain problem (e.g., WSAN).

4.2.2. *Business process management and economics of SPL*

This category presents gaps of studies in adapting business processes and requirements, as well as investigating possible costs associated with the impact of automation, and selection of features in SPL. Few studies apply metrics or trade-off analysis for managing IoT systems through SPL. Fig. 9 presents the gaps in this category and their subcategory *Knowledge and Business Rule Configuration*.

Three studies were identified on the SPL reuse automation and their associated costs [53,54] to features selection [55] considering the complexity and economic factors that may impact specific domains (e.g., agriculture) [54].

Venčkauskas et al. [53] investigated gaps in the automation and adaptation in systems design for body area network (BAN). This study proposes a model-driven approach (MDA) according to a feature-based variability model to perform the management of BAN systems. However, we identified the lack of MDA studies that allow the automation in different resolution times (e.g., runtime).

Cetina et al. [56] observed gaps in the automation operations and associated costs of autonomous systems for smart-homes architectures. Cetina et al. [56] discuss the autonomy of these systems at runtime. We believe that hybrid approaches with more types of resolution time may be contributing to reduce costs in the long term and encourage the development of new research.

Abbas et al. [55] identified as gaps the costs and complexity associated with the correct selection of features for the generation of IoT systems based on a binary pattern for nested cardinality constraints (BPNCC) approach. We think there is a gap concerning the lack of approaches to measure the number of selected features in a determined context.

Köksal and Tekinerdogan [54] explored gaps that impact the production and economy in the smart farming environment encompassing IoT-based farm management information systems (FMISs). In this scenario, an approach was designed using feature-driven domain analysis to understand and model smart farming requirements based on IoT reference architectures for FMISs. Based on Köksal and Tekinerdogan [54], we observed a lack of advanced variability mechanisms to increase production and reduce costs in the field of smart farming.

We find gaps in the quality of processes adaptation and business requirements based on variability and dynamic features modeled in IoT systems. Sinnhofer et al. [57,58] addressed the poor adaptation of processes against the production speed of companies to model constraints considering the improvement in developing low-cost IoT systems to reduce time-to-market. This study proposes the process variability framework project to minimize this gap of communication between variabilities and business processes. Suri et al. [59] encompassed the gap in managing variabilities and business requirements in the context of IoT due to the existence of different IoT devices in a cross-organization business process. Thus, the configurable process model (CPM) based on business process modeling language (BPMN) was proposed as an attempt to solve such a gap investigated in a few studies in the literature. Baresi

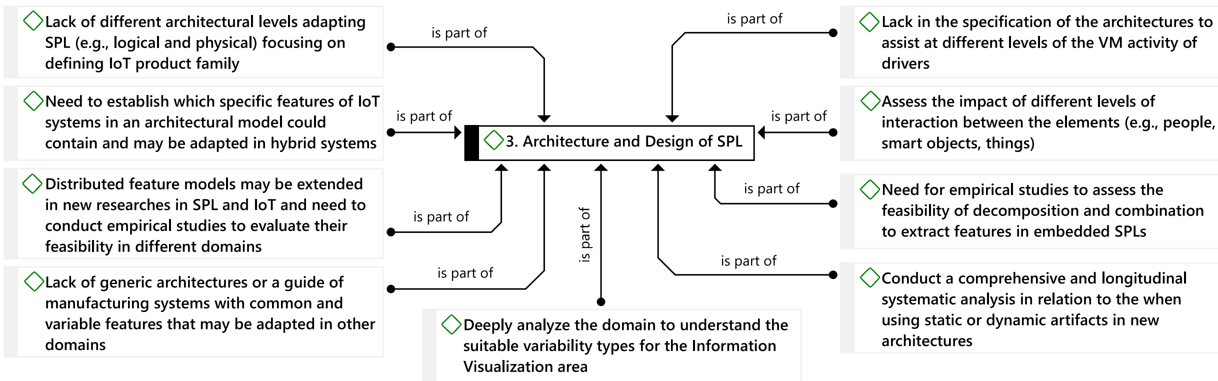

Fig. 10. Gaps categorized by “3. Architecture and Design of SPL” research topics.

et al. [36] presented as a gap the impact on features changes in service-oriented dynamic software product lines, considering a set of functionalities that must communicate correctly in runtime. In this scenario, this study proposes an approach based on dynamic SPL, common variability language (CVL), and business process execution language (BPEL) in which it is represented as an example in a home automation features model for smart homes.

We highlighted a few studies that show ways as gaps to measure and scale phases of development and management of IoT systems through SPL [10–60]. In Kneer and Kamsties [60], benchmark gaps in self-configuring systems (SCS) for IoT were observed to compare approaches of SCS for IoT. Anon et al. [10] identified gaps in the selection and application of technologies for the development and management of new IoT applications owing to the paradigm growth.

4.2.3. Knowledge and business rule configuration

This subcategory of Business Process Management and Economics of SPL, we present two studies with gaps regarding the lack of how the practitioners and researchers’ knowledge may impact or change the business or the design of generic IoT systems. Cecchinel et al. [61] gaps were identified in data collection as well as in the software engineers’ knowledge of development and deployment of heterogeneous shared sensing infrastructures based on SPL concepts. Jazayeri et al. [62] identified as a gap the knowledge that may impact software design as well as possible business rules objectives. Thus, a variability model was proposed as an approach using the orthogonal variability model (OVM) to represent the variabilities in store-oriented software ecosystems product line.

4.2.4. Architecture and design of SPL

This category presents studies with gaps in architectural design between SPL and IoT systems. It presents how the design of the VM activity has been carried out, the design of new methods, and the possibility of choosing the artifacts types that may be adapted in this context. Fig. 10 represents the gaps in this category.

The architectural design between SPL and IoT is presented in terms of its abstraction levels (physical and logical), dynamism, and integration with SPL in three main studies of Di Cola et al. [63], Santos and Machado [64], and Moritani and Lee [65].

Di Cola et al. [63] identified gaps in the design of a family of systems with logical architectures for IoT, considering the distinction between different levels and proposed logical architectures. Di Cola et al. [63] proposes variation points in such architectures to a family of IoT systems. We identified a lack of different architectural levels adapting SPL (e.g., logical and physical) focusing on defining IoT product family.

Santos and Machado [64] investigated as a gap the dynamic SPL architecture model for self-adaptive systems related to two main questions "*when or how to adapt*?" considering the specific software development based on context and actions to adapt (e.g., IoT or smart environments). Santos and Machado [64] study helps to motivate preliminary projects of new architectural models in DSPL. An experiment was developed to evaluate the architecture model at runtime. We consider the need to establish which specific features of IoT systems in an architectural model could contain and may be adapted in hybrid systems.

Moritani and Lee [65] investigated as gaps the need to manage product configurations supported by an architecture design for distributed features. In this context, this study proposed a DisFM approach. The guidelines aim to manage the distributed feature units included in feature models involving self-adaptive dynamic SPL. We think that the idea of distributed feature models may be extended in new researches in SPL and IoT and need to conduct empirical studies to evaluate their feasibility in different domains.

The VM activity design is carried out in three studies using a variation points mechanism [66], a product derivation method [67], and an interactive multi-perspective process in the context of smart green buildings (SGB) [41].

McGee et al. [66] explored as gaps limitations in the design variation points in architecture and implementation in IoT manufacturing systems. McGee et al. [66] present an assembly line using the monitoring system feature model and Bosch XDK. McGee et al. [66] discuss that the approach may be adapted in other systems, but we have not identified a guide for how to adapt the approach in different domains. We suppose in the lack of generic architectures or a guide of manufacturing systems with common and variable features that may be adapted in other domains.

Hosoai et al. [67] investigated as the main gap drivers development for small appliances and present a method that uses feature models for drivers VM by software engineers. Only a study was identified in relation to the VM of different drivers. Thus, we observed a lack in the specification of the architectures to assist at different levels of the VM activity of drivers.

Murguzur et al. [41] study highlights the lack of a multi-perspective process that involves interaction between people and things. New empirical studies should be carried out in the industry to assess the impact of different levels of interaction between the elements (e.g., people, smart objects, things).

New feature extraction methods [19] as well as the choice between static or dynamic artifacts [35], and thedesign of monitoring techniques [68] are discussed ways for the design of new IoT systems.

Yang and Zhang [19] identified as a gap and a lack of effective methods to extract features in embedded SPLs. This approach follows a process of combination to VM in the modeling of features with the use of the FODA method. Due to the limitation of the quantity and faults in methods to extract features discussed by Yang and Zhang [19], we identified that there need for empirical studies to assess the feasibility of decomposition and combination to extract features in embedded SPLs.

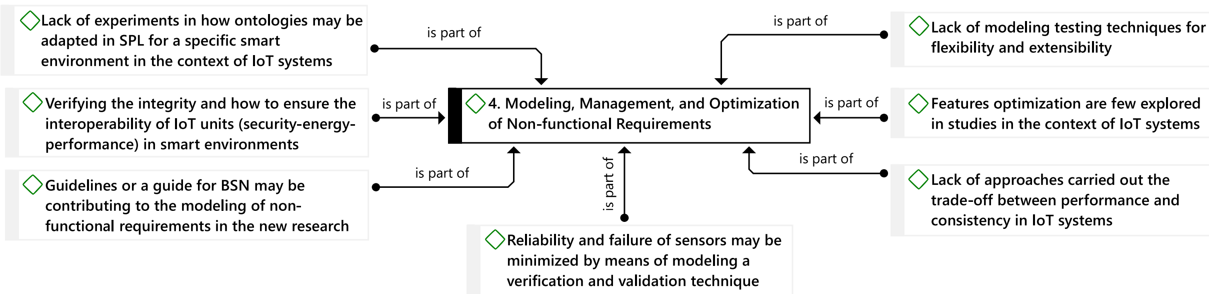

Fig. 11. Gaps categorized by "4. Modeling, Management, and Optimization of Non-functional Requirements" research topics.

According to Lee and Kang [35], most existing approaches in the literature use static software artifacts. The main discussion gap identified in Lee and Kang [35] is related to use static VS. dynamic artifacts. Thus, this study proposes a dynamic approach used in SPL at runtime, during monitoring, and based on five guidelines applied to the design of the architecture. We suppose a need to conduct a comprehensive and longitudinal systematic analysis in relation to the when using static or dynamic artifacts in new architectures.

Logre et al. [68] addressed as the main gap the monitoring and integrating information visualization in dashboards taking IoT devices into consideration, such as car sensors, ports, smartphones, among others. Thus, this study proposes an approach based on modeling composition techniques to allow the design of monitoring dashboards considering their variabilities. Logre et al. [68] mention that new empirical studies will be carried out in conjunction with the Human-Computer Interaction area to understand this context. We believe it is necessary to deeply analyze the domain to understand the suitable variability types for the Information Visualization area.

4.2.5. *Modeling, management, and optimization of non-functional requirements*

This category presents studies on the modeling and management of IoT systems, including non-functional requirements as reliability in the sensor management in networks, and also in the features reconfigurations optimization between SPL and IoT systems. Fig. 11 represents the gaps in this category.

Two studies present the modeling and management of feature models adapted in the context of smart objects [69] and IoT units [70].

Nešković and Matić [69] investigated as gaps in the modeling and mapping of context-aware information using ontologies through different contexts in feature models. Based on Nešković and Matić [69] we identified a lack of experiments in how ontologies may be adapted in SPL for a specific smart environment in the context of IoT systems.

Venčkauskas et al. [70] identified gaps in the management of bidirectional wireless communication between IoT units (security-energy-performance). Several approaches have attempted to solve these gaps in environments that are impacted by noise or network failure. This study proposes a framework through feature-based modeling for IoT units. We suppose that new researches may be verifying the integrity and how to ensure the interoperability of IoT units (security-energy-performance) in smart environments.

Non-functional requirements as dependability [37], reliability [37–71], maintenance [37], flexibility [72], and extensibility [72] are presented in dynamic SPLs, cloud-based services, and in a framework for modeling smart cities.

In Pessoa et al. [37] encompassed as gaps the dependability to reliability and maintenance in dynamic SPLs in the safety-critical domains. Pessoa et al. [37] compared variability approaches to develop plans and evaluate a dynamic SPL feature model in the BSN domain. We suggest guidelines or a guide for BSN may be contributing to the modeling of non-functional requirements in the new research.

Murwantara et al. [71] encompassed gaps in reliability and sensor failure in the attempt to propose an adaptive sensor-cloud approach for reliability management of IoT sensors in different cloud-based services. We observed that the reliability and failure of sensors may be minimized by means of modeling a verification and validation technique.

Smiari and Bibi [72] investigated gaps in smart city application engineering involving flexibility and extensibility in such an environment. Thus, the smart city application modeling framework (SCAMF) was designed to manage and optimize the development of new smart city applications. Despite the study of Smiari and Bibi [72] propose metrics for smart cities, we identified that there a lack of modeling testing techniques for flexibility and extensibility.

Abbas et al. [73] investigated the main gap in feature VM optimization and violations constraints due to the number of combinations of common and variable features for effectively deriving different products. In the last study, multi-objective optimum-BPNCC (MOO-BPNCC) approach was proposed to minimize this gap. However, the features optimization are few explored in studies in the context of IoT systems.

In Weckesser et al. [74], the main gaps identified are related to optimal configurations regarding the wireless sensor networks (WSN) in the context feature model (CFM), including using performance, consistency, and reconfiguration knowledge. Still, there a lack of approaches carried out the trade-off between performance and consistency in IoT systems to guarantee an appropriate optimization.

4.2.6. *Artificial intelligence and agents for variation*

This category presents two main gaps identified in three studies by Ayala et al. [40–75,76,77]. Fig. 12 represents the gap in this category.

The idea presented in the studies by Ayala et al. [40–77] is the adaptation of intelligent agents in a runtime environment according to the IoT systems and their devices. We consider conducting a domain analysis in design-time to assess the viability of agents in the context of managing IoT systems at runtime. Ayala et al. [76,77]. performed VM of the heterogeneous IoT devices through self-adaptive agents for ambient intelligence systems (ambient assisted living (AAL) or smart shopping) using SPL engineering. They investigated gaps of dynamic adaptation of agents in runtime considering their behavior according to changes in different environments using a variability model that applies MAPE-K cycle with self-management policies. Ayala et al. [75] investigated gaps in VM heterogeneous IoT devices through self-adaptive agents for ambient intelligence systems (e.g., ambient assisted living (AAL) or intelligent transport systems) using SPL engineering.

4.2.7. *Evolution and continuous integration of SPL*

We observe in this category the evolution management, changes, and the integration of different applications through SPL in IoT systems or devices. Fig. 13 represents the gaps in this category.

Gamez and Fuentes [78] observed as the main gap the evolution and deployment management of changes in ambient intelligence systems taking the diversity of existing devices and technologies into consideration. Observing the Gamez and Fuentes [78] study, we think in the possibility that few studies in the literature contain an approach for

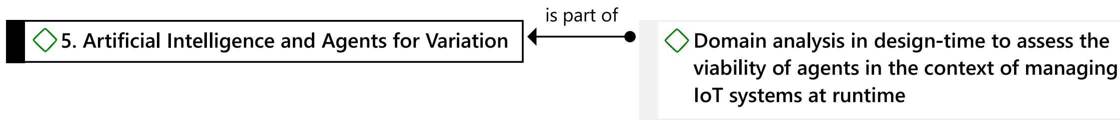

Fig. 12. Gaps categorized by “5. Artificial Intelligence and Agents for Variation” research topics.

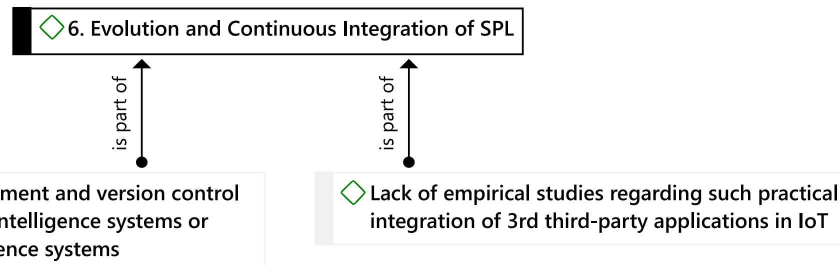

Fig. 13. Gaps categorized by “6. Evolution and Continuous Integration of SPL” research topics.

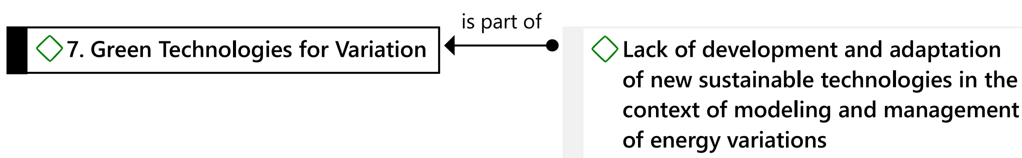

Fig. 14. Gaps categorized by “7. Green Technologies for Variation” research topics.

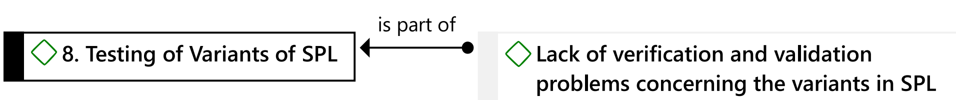

Fig. 15. Gaps categorized by “8. Testing of Variants of SPL” research topics.

the configuration management and version control to evolution ambient intelligence systems or environmental intelligence systems.

Tomlein and Grønbæk [79] presented as a gap in the installation and integration of third-party applications from different manufacturers in the context of embedded software ecosystems (ESE) for IoT. Only this study mentions the integration of third-party applications in the context of IoT. Thus, we suppose that there is a lack of empirical studies regarding such practical integration of third-party applications in IoT.

4.2.8. *Green technologies for variation*

This category presents studies with gaps in the lack of development and adaptation of new sustainable technologies in the context of modeling and management of energy variations based on the VM activity. Fig. 14 represents the gap in this category.

Marimuthu and Chandrasekaran [20] investigated as gaps in the development of energy-aware self-adaptive systems related to different operating conditions, policies, and behavior. In this scenario, a domain analysis framework was proposed for the FODA feature models for VM based on contextual information. Munoz [80]., Munoz et al. [81,82] presented as the main gap in the modeling and management problems of energy consumption variants. The proposed HADAS variability model manages different contexts of energy consumption through the SPL service as a collaborative repository.

4.2.9. *Testing of variants of SPL*

This category presents only one study regarding gaps in the lack of verification and validation problems concerning the variants in SPL. Fig. 15 represents the gap in this category.

In Damiani et al. [83], part of HyVar project of Mauro et al. [49] presented an extension of abstract behavioral specification (ABS) toolchain as a solution for the possible gap in checking problems owing to a large number of variants that an SPL may contain. In this context, the ABS modeling language and the toolchain allied to the support provided to delta-oriented programming (DOP) may facilitate checking these DOP SPLs. Fig. 15 represents the gaps in this category.

5. Discussion

The following sections provide a discussion on the answers to the RQs towards the specific objectives of this SLR.

5.1. *RQ1*

How are software product lines (SPLs) being applied in the context of the Internet of Things (IoT) systems?

We observe that SPL engineering is applied in the studies with SPLs for IoT systems through domain and application engineering. Domain engineering is adopted to describe a problem space for representing specific SPL customized products for IoT. Overall, among the 53 selected primary studies analyzed, SPLs were proposed in 29 studies to describe new proposals or VM for IoT systems. We observed that most SPLs presented in the primary studies were applied in studies using feature models. This type of model has been applied to propose new approaches in different smart environments and to conduct initial evaluations or case studies in IoT. Few SPLs are represented by the common variability language (CVL) [36] (e.g. [76,77]), UML component [51], and class diagrams [72].

The diversity among the new proposed SPLs is evident in most primary studies. However, the SPLs identified are described with few details and should be improved before being adopted in new researches. In this sense, these SPLs need to be expanded and evaluated in new researches to ensure that they are adopted correctly. Thus, we observed that most primary studies do not conduct systematic, rigorous experiments or case studies isolated in the academy or industry environments, aiming to generalize their results. A summary of each one of the studies is presented in the following paragraphs.

Table 4
Smart application areas/domains by the selected primary study.

Smart application areas/domains	Selected primary studies
Smart health (e.g., baby care system, health monitoring system, personalized health monitoring (PHM), body area network (BAN), body sensor network (BSN), ambient assisted living (AAL))	[37,38,46,47,53,55,65,77]
Smart home (e.g., home service robot)	[10,35,36,43,56,64,67]
Smart cities (e.g., smart shopping, public smart street light, transportation)	[39,41,60,72,75,76]
Smart infrastructure (e.g., ambient intelligence systems, shared sensing infrastructures, adaptive sensor-cloud)	[19,61,70,71,75,78]
Automotive domain	[45,49,50,63,83]
Smart energy (e.g., energy-aware self-adaptive systems)	[20,80–82]
Smart farming (e.g., smart irrigation)	[43,52,54]
Smart business process models	[57–59]
Wireless communication (e.g., wireless sensor actuator network (WSAN), wireless sensor network (WSN), human in the loop monitoring system (Bluetooth))	[18,66,74]
Software ecosystems (SECO) (e.g., embedded software ecosystems (ESE), store-oriented software ecosystems (e.g., mobile apps, web browser plug-ins, in-house software, open source software and web services))	[62,79]
Generic smart environments (e.g., smart environment development lifecycle)	[41,42]
Security settings, Speech recognition and Database (Berkeley DB)	[44,73]
Safety-critical domains (e.g., aviation, autonomous vehicles, nuclear energy)	[48]
Context-aware systems of mobile applications	[45]
Smart agents	[40]
Smart objects	[69]
Robotized systems	[51]
Dashboards design	[68]

Kneer and Kamsties's [60] study considers the smart city environment. These authors proposed a public smart street light feature model for self-configuring systems (SCSs). Such systems in IoT need a certain dynamism and variabilities that may be provided from SPL engineering in resolution time (e.g., requirements, design-time, and runtime). Some companies such as Siemens[6] have adopted SPL engineering after understanding its benefits for domain analysis and the prospect of new products to the worldwide market.

Initially, Table 4 categorizes the main smart application areas/domains related to SPLs concepts applied in the context of IoT. Most selected primary studies appear in smart health, smart homes, smart cities, smart infrastructure, and the automotive domain. Few studies investigated other domains as smarty energy, smart farming, smart business process models, and specific types of wireless communication. The least explored domains encompassing a minor number of studies are software ecosystems (SECO), other generic smart environments, safety-critical domains (e.g., aviation), and context-aware systems of mobile applications.

SPL engineering was also applied for process specifications from SPLs proposed in some selected studies. Ayala et al.'s [40–77] studies developed the smart shopping center SPL to illustrate a specific dynamic process named Self-StarMAS aiming to implement multi-agent systems (MAS) in IoT. Gamez and Fuentes [78] proposed an automated process using the FamiWare system family proposed based on the cardinality-based feature models in the context of smart buildings. Baresi et al. [36] illustrated a new process for home automation through the Smart Home SPL feature model.

In the context of smart homes, Cetina et al. [56] proposed a Smart Home dynamic SPL feature model containing the main features of security, automated illumination, and multimedia to provide autonomic systems based on this type of environment. Santos and Machado [64] developed a Smart Home System dynamic SPL through feature models using context sensors in a dynamic scenario to illustrate a new architectural model for SPL.

Beyond domain engineering applied to smart homes in IoT, SPL engineering was extensively applied in other domains, such as in IoT monitoring systems. Analyzing the majority of the studies, we discovered that the wireless sensor networks area, as well as monitoring systems, contain several SPLs, as in Amja et al.'s [46] study, which proposed a simplified health monitoring system feature model for context-aware applications in IoT. Pessoa et al. [37] proposed a dynamic body sensor network (BSN) SPL for monitoring health conditions in a smart environment. Further, another partial feature model named baby care system (BCS) was designed by Moritani and Lee [65] for monitoring baby sleep by means of distributed nodes (e.g., room temperature or noises).

Furthermore, in industrial monitoring applications of IoT, two studies of Venčkauskas et al. [53–70] proposed a smart sensor controller family represented by means of a partial feature model named securitySettings to model IoT units related to security-energy-performance. Abbas et al. [73] also designed another feature model named Security Settings to evaluate a multi-objective algorithm for managing different features. Abbas et al. [55] also proposed a pulse sensor IoT feature model to illustrate an approach based on the binary pattern (0 and 1 variabilities) using cardinality models. Weckesser et al. [74] used wireless sensor networks (WSN) in the context feature model (CFM) to evaluate optimal reconfigurations of SPL considering performance models.

In McGee et al. [66] a monitoring system feature model using Bosch XDK[7] devices is applied in the industrial scenario of IoT to monitor different applications considering specific protocols, line configuration, and data points. Cecchinel et al. [61] developed an unnamed SPL based on feature models and applied it to shared sensing infrastructures to automate data collection policies in sensor platforms. Hosoai et al. [67] managed sensors using a motion tracking system product family feature model to demonstrate and apply customization of small appliances in IoT systems in the industry.

Two SPLs were developed and applied in the robotics area to consider each robot as a thing in IoT within the smart industry. The main difference is in the context in which the robots are represented and managed. Lee and Kang [35] proposed a home service robot (HSR) SPL feature model to offer different services for humans, and Heikkilä et al. [51] used the Unified Modeling Language (UML) Component Diagram to represent robots through the object detection SPL. Considering the UML, the class diagram was adopted by Smiari and Bibi [72] based on rules to generate this class diagram from transformations using the smart retail platform feature model. In addition to this smart industry environment, Yang and Zhang's [19] study implemented a ventilator embedded SPL feature model to control the industrial ventilation system.

Apart from the previous environments in which SPL engineering was applied, Ortiz et al. [18], Logre et al. [68], and Achtaich et al. [52] applied the SPL in smart environments control, which involves sensors

[6] Siemens site: http://www.siemens.com.

[7] Bosch XDK site: https://xdk.bosch-connectivity.com/.

around people lives. Ortiz et al. [18] presented indoor smart environmental and guidance systems SPL based on the FODA feature model to generate museum guidance products, and Logre et al. [68] presented the SmartCampus feature model applied in dashboard customization for monitoring the academic life in a specific university campus. Achtaich et al. [52] developed a smart irrigation system feature model containing sensors for agriculture. Besides, Köksal and Tekinerdogan [54] developed two feature models as case studies titled smart tomato production and smart wheat production in the context of the IoT-based farm management information system (FMIS).

In the automotive sector, two studies applied SPL engineering and developed initial SPLs. Mauro et al. [49] proposed the Car feature model to evaluate context-aware reconfigurations in the Car Assistance System, and Di Cola et al. [63] developed the vehicle control systems product family with logical architecture for generating different car products.

5.2. RQ2

How is Variability Management (VM) of SPL carried out in IoT systems?

This research question will be answered based on the following points: (i) processes related to VM activity, (ii) VM resolution (e.g., design-time or runtime), and (iii) variation points and/or relationship types (e.g., mandatory, alternative, optional, AND, OR, excludes, and requires) to perform VM. Table 5 presents such points related to each one of the selected primary studies.

We observed that VM activity for IoT is carried out using the proposed IoT process that can be expanded to a multi-perspective process using mainly feature models with variation points, variants, relationships and constraints of the literature. Observing selected primary studies, feature models are the first step to variability management in IoT systems. Runtime resolution is most applied to adapt features of IoT systems due to their dynamic behavior. However, different models as UML appeared only once. Some tools identified in the studies help model IoT systems regarding runtime.

Domain analysis has been few explored in the literature based on the design phase of the SPL engineering. For instance, the orthogonal variability model (OVM) can be applied as a feature modeling strategy to manage ecosystems around IoT. The design of meta-models, frameworks, and the adaptation of specific languages (e.g., XML) can be useful to perform VM activity. Nevertheless, it is necessary a wide knowledge to combine these several approaches or languages to provide a correct VM activity.

Integration problems of the contextual variabilities generate several impacts to manage the diversity of things in IoT systems. These VM activity impacts can be minimized through approaches with rules mechanisms supported by a set of variation points. We believe that a hybrid standardized approach to VM activity should be a way to facilitate the correct management of things in the IoT systems.

Following is a summary of each of these studies.

The VM activity has been carried out to manage things through IoT process specifications from SPL engineering in the selected studies. Murguzur et al. [41] presented a multi-perspective process for IoT involving two perspectives that interact with people (e.g., group of stakeholders) and things (e.g., energy consumption) through the resolution model that contains variants mapped using specific relationships (mandatory, optional, alternative, AND, or OR). These variants are determined in "value resolution" for variability, in "fragment resolution" to represent variation points, in "perspective resolution" to show contextual data, and in "variability constraints" to validate such a modeling process. Sinnhofer et al. [57,58] proposed a *process variability framework* that combined variability management to customize the SPL business process for IoT systems to reduce costs and time-to-market, providing modeling constraints and improving the communication between process variabilities in IoT. Smiari and Bibi [72] provided the *smart city application modeling framework* (SCAMF) with a variability mechanism (which has basic relationships such as mandatory, optional, more-of (OR), and one-of (alternative)) to extend and capture the requirements of a smart city in the application design phase to simulate a feature model using a set of rules to generate a UML class diagram. Suri et al. [59] applied variants and variation points to manage replication and shareability in the *configurable process model* (CPM) level that can be used in the IoT domain to model variabilities. This proposed CPM process can be configured through operators, parameters, relationships types (AND, OR, and XOR) using constraints, and assignment policies related to the behavior of things. Furthermore, Köksal and Tekinerdogan [54] performed VM using feature domain analysis using the basic relationships such as mandatory, optional, and alternative, which were adapted according to the reference architecture, and each application feature model can be generated based on the smart production environment in an FMIS.

As presented in previous studies, there have been attempts to propose new processes to manage the variabilities of things in IoT. International research projects such as European Union H2020 *HyVar* Project,[8] described in Røst et al. [50] and Damiani et al. [83] studies, propose a toolchain that supports high reuse through a well-defined process for VM in runtime for IoT (originally applied in the automotive sector). Besides, in a previous paper, Mauro et al. [49] presented the HyVarRec—a hybrid variability reconfiguration engine (see Section 5.1) that can be applied to configurations for automotive feature models. In Tekkalmaz et al. [42], the variability management activity is performed in the GAMMA tool by means of several terms specified in the GAMMA domain dictionary (e.g., "Data Point Type" defines the features of an entity as well as dependencies in the automation domain). Furthermore, in Sharif et al. [39] the real-time bus tracking mobile application developed contained the VM adapted to the application requirements of the University of Malaysia. The features encompassing the requirements related to "select destination", "scan location by QR code", "detect user current location", and others. The main relationships defined were (i) "x" as functional requirements, (ii) "P" as an optional feature, and (iii) "c" as a common feature.

In the sense of organizing processes and managing things, problems of integration between things arise when different devices are considered. Thus, Tomlein and Grønbæk [79, 43] investigated the integration of *third-party* applications using contextual semantic variability models in runtime in the context of *embedded software ecosystems* (ESE) for IoT. In Jazayeri et al. [62], this integration problem is also applied in the variabilities representation approach based on *store-oriented software ecosystems* SPLs (e.g., the smart voice assistant named Amazon Alexa[9]) using the *orthogonal variability model* (OVM) in IoT systems. This approach aims to customize these ecosystems containing three views, encompassing business, application, and infrastructure views, which are represented using variations points and variants with common relationships (mandatory, optional, and alternative).

In software ecosystems, Mens et al. [45] investigated contextual feature modeling strategies, contextual VM approaches, and their runtime dynamic behavior. These are considered challenges in runtime context-aware systems owing to the diversity of things. In the attempt to solve such management problems, Nešković and Matić [69] developed a meta-model using ontologies to represent contextual variabilities in feature models in different views in design-time and runtime. Abbas et al. [44] aimed to generate customized IoT products using an XML-based feature model representation based on the meta-model to manage such variabilities (mandatory, alternative, with constraints) from the IoT-specific domain. Further, using common relationships (mandatory, optional, alternative, and OR), Amja et al. [47] examined how to use the relational concept analysis (RCA) and variability models to manage contextual variabilities.

[8] *HyVar* Project: https://www.hyvar-project.eu.

[9] Amazon Alexa site: https://developer.amazon.com/alexa.

Table 5
VM activity in main selected primary studies.

Proposals to VM activity	VM resolution	Relationship types/Mechanisms	Selected primary studies
Multi-perspective process for IoT	Runtime. Resolution model: Value, Fragment, and Perspective resolution.	Relationship types: mandatory, optional, alternative, AND, OR, and variability constraints.Mechanisms: variation point and variants.	Murguzur et al. [41]
Process variability framework	Runtime and design-time.	Relationship types: mandatory, optional, and constraints (requires).Mechanisms: variation point, variants, and variability.	Sinnhofer et al. [57,58]
Smart city application modeling framework (SCAMF)	Design-time	Relationship types (stereotypes): mandatory, optional, more-of (OR), and one-of (alternative).Mechanisms: a set of rules and variants.	Smiari and Bibi [72]
Configurable process model (CPM)	Design-time	Relationship types: AND, OR, and XOR.Mechanisms: variation points, variants, operators, parameters, and assignment policies.	Suri et al. [59]
FMIS development approach	Design-time	Relationship types: mandatory, optional, and alternative.Mechanisms: manages variants.	Köksal and Tekinerdogan [54]
HyVarRec - a hybrid variability reconfiguration engine	Runtime and Design-time	Relationship types: mandatory and optional.Mechanisms: variability realization mechanism.	Røst et al. [50], Damiani et al. [83] and Mauro et al. [49]
GAMMA	Runtime and Design-time	Mechanisms: basic data type, data point type, data store, data type, feature type, and service type.	Tekkalmaz et al. [42]
CVAnalysis	Runtime	Relationship types: "x" as functional requirements, "P" as an optional feature, and "c" as a common feature.Mechanisms: CVAnalysis model to variability management.	Sharif et al. [39]
Semantic application model	Runtime	Mechanisms: semantic variability models and semantic annotations	Tomlein and Grønbæk [43,79]
Variability model	Runtime	Relationship types: mandatory, optional, and alternative.Mechanisms: variations points and variants.	Jazayeri et al. [62]
Context-Aware Systems Variability	Runtime	Mechanisms: Context-feature-modeling strategies.	Mens et al. [45]
Model architecture	Runtime and Design-time	Relationship types: mapping roles, ordinary mapping, requires and excludes Mechanisms: metamodels	Nešković and Matić [69]
Variability management of IoT xml-based feature modeling	Runtime and Design-time	Relationship types: mandatory, optional, alternative, OR, constraints (include and exclude).Mechanisms: meta-model, XSLT files, and XML-schema	Abbas et al. [44]
Semantic model with ontology	Design-time	Relationship types: mandatory, optional, OR, alternative, and constraints (requires and excludes).Mechanisms: variation points, variants, context modeling, semantic relationship, and context rules.	Amja et al. [47]
Dynamic adaptive variabilities	Runtime and Design-time	Mechanisms: variation point (implicit, designed, bound, open, closed, static and dynamic), variants.	McGee and McGregor [84]
Framework for IoT in Cloud Computing	Runtime	Relationship types: Encompass three layers, “Device Layer”, “Central Hub Layer”, and Cloud Layer.Mechanisms: variation points, and variants.	Anon et al. [10]
HADAS variability model	Runtime and Design-time	Relationship types: yes/no decision (option), variable type (specific value), one-to-many, mandatory, and optional.Mechanisms: variability tree, variation points, and variants.	Munoz [80–82]
Domain analysis framework	Runtime and Design-time	Relationship types: mandatory, optional, alternative, OR, crosstree constraints (excludes and requires).Mechanisms: validate configurations named energy-aware application configurations (EAAC), and variants.	Marimuthu and Chandrasekaran’s [20]
Adaptive sensor-cloud model	Runtime	Relationship types: mandatory, OR, and alternative.Mechanisms: an adaptive mechanism to sensors.	Murwantara et al. [71]
Model-driven framework to smart health monitoring	Runtime and Design-time	Relationship types: mandatory, alternative (OR or XOR), optional, constraints (requires and excludes).Mechanisms: variation points, and variants.	Venčkauskas et al. [38]

McGee and McGregor [84] investigated dynamic adaptive variabilities in runtime applied to safety-critical systems for IoT, providing a set of variation points defined as “implicit” (not specified in previous SPL architecture), “designed” (describing design decisions), and “bound” (a variant can assume their location). Besides, variants were defined with specific states such as “open” (it is possible to place additional variants at a variation point) and “closed” (new variants are not added at a variation point) as well as with variant types that can be “static” or “dynamic.”

Other studies performed VM to benefit end-users by providing a framework to manage things and implement new applications for the IoT [10]. In Anon et al. [10] a framework was proposed through a feature model, which was divided through three layers—the “Device Layer” (containing end devices, things, e.g., microcontrollers such as Arduino and Raspberry Pi), “Central Hub Layer” (containing a remote application connected to end devices), and “Cloud Layer” (containing an interface for processing and storing data related to the Central Hub Layer). These layers consist of variation points (e.g., a microcontroller) and variants (e.g., Arduino or Raspberry Pi) that users can customize to generate new applications for IoT based on the cloud computing infrastructure.

Following the cloud computing context as well as sustainable software development thinking in IoT systems, developers may need an efficient energy model. In this sense, using the concept of VM, Munoz [80] and others [81, 82] proposed a set of variation points and variants from the HADAS variability model to manage the contexts of energy consumption. A variability tree with four levels was proposed, considering *Energy Consuming Concerns* (ECCs), design variants, technology alternatives, and the energy context. This approach provides a collaborative repository and generates sustainable analyses related to energy consumption that can be accessed by developers. In addition, Marimuthu and Chandrasekaran's [20] study also collaborated to manage energy consumption efficiently, proposing a domain analysis framework to energy-aware self-adaptive systems. This framework was represented by FODA models to apply VM to the contextual data of energy levels for IoT systems. From this, basic relationships were specified to manage energy-aware features regarding energy-efficient alternatives, such as mandatory (e.g., energy-friendly, energy-hungry), optional, alternative, and OR, as well as energy-aware constraints (excludes and requires) complemented by one mechanism to validate configurations named energy-aware application configurations (EAAC).

In Murwantara et al. [71] the adaptive sensor-cloud model was initially modeled from the feature model architecture. The main features involved sensor-type management, geolocation, and data types. A transition model, dynamic software product line (DSPL) DSPL, and model@runtime were used to implement such adaptive sensor-cloud mechanisms, aiming to adapt services for IoT and reconfigure sensors according to the smart environment.

Venčkauskas et al. [38] proposed a *Model-Driven Framework* that consists of variation points and variants to *personalized health monitoring* (PHM), including a prototype named *IoT Nodes (IoTN) Pulse Sensor* represented by the feature model with basic relationships (mandatory, optional, alternative (OR and XOR), and constraints (excludes and requires)) following the variation points of security level, energy consumption, and environment. The VM activity example performed can be accessed through remote users accessing the hardware prototype implemented with C# language, ZigBee, and two modules of gas sensors and nodes.

5.3. RQ3

Which Approaches, Frameworks or Platforms use SPL in IoT systems?

IoT approaches, frameworks or platforms are applied to SPL in different smart environments. Several distinct proposals support these environments as processes, business processes, meta-model, semantic approaches, contextual models, ontologies, and approaches with variability mechanism. The diversity of languages as CVL helps to these proposals for the IoT things management in the runtime phase, using *delta-oriented programming*, Notation 3, or UML diagrams representations (e.g., class diagram).

In design-time, some technologies as *domain-specific language*, FX-MAN component model, GAMMA tool suite, and representations as a *cardinality-based feature model* allow reducing modeling and complexity in heterogeneous IoT systems. However, these proposals do not provide generic ways to IoT modeling, management, integrate, and communicate things effectively. We believe that such proposals should be expanded to understand generic domain problems. Several runtime solutions were present in most studies before understanding the IoT scenario reality. A summary of each one of the studies is presented following.

In the context of smart cities for IoT systems, Kneer and Kamsties [60] developed a model-based framework for SCSs with smart street illumination composed by smart lamps evaluated through a case study using the *Public Smart Street Light* feature model. In the context of smart cities, the SCAMF framework proposed by Smiari and Bibi [72] contains a variability mechanism and has four layers of rules: (i) enterprise business, (ii) application business, (iii) interface adapters, and (iv) frameworks and drivers. The re-engineering of systems is performed based on these steps. First, the requirements are identified based on transform systems (e.g., feature models) in class diagrams.

Taking process specifications from SPL engineering into consideration, Ayala et al. [40–75] specified the *Self-StarMAS* process and *VSpec* tree to support the VM activity resolution process, including self-adapted agents for *Ambient Assisted Living* (AAL). In other studies of Ayala et al. [76, 77] the *Self-StarMAS* process was evaluated and improved through a combination of the *Self-StarMAS* process using common variability language (CVL) based on the *Model-Driven Design* (MDD). The *Self-StarMAS* process applies the dynamic adaptation of agents in runtime in different environments using a variability model that applies MAPE-K (*monitoring, analyzing, planning and execution - knowledge*) in conjunction with self-management policies. Gamez and Fuentes [78] also applied the *cardinality-based feature models* through middleware with an automated process. Thus, in this study, a *VML4FamiWare* language was specified supported through the developed tool Hydra to guarantee VM.

Considering approaches using the *cardinality-based feature model*, Abbas et al. [55] proposed a *binary pattern for nested cardinality constraints* (BPNCC) with five steps to calculate the number of combinations between features using a binary standard (0/not selected feature and 1/selected). Another study by Abbas et al. [73] proposed a *multi-objective optimum-BPNCC* (MOO-BPNCC) containing three paths (A, B, and C), using heuristics to minimize the complexity and execution time related to choosing adequate features according to the feature model provided. Thus, MOO-BPNCC can be considered as an algorithm evaluated using three different features models to verify what paths are appropriate for each scenario. Furthermore, Abbas et al. [44] attempted to apply XML-based feature models for reuse features to generate IoT applications from specific domains.

During our review, some processes and approaches were identified for smart homes for IoT systems. Among these studies, Baresi et al. [36] proposed a process based on SPL, using CVL and *business process execution language* (BPEL) for smart homes. The CVL performs the features management and process execution through BPEL. In this study, the Variability Designer tool was developed based on the CVL and allowed dynamic feature modeling. Among the processes identified in this SLR, Suri et al. [59] used CPM based on *business process modeling language (BPMN)* to model VM and business requirements in IoT. This process was extended using a Signavio[10] open-source environment to provide BPMN 2.0 graphically using specific variants and variation points to generate and process IoT products. Sinnhofer et al. [57, 58] also proposed a process named *process variability framework* project to facilitate the communication between variabilities and business processes through SPL engineering through VM represented using the pure::variants[11] tool. Murguzur et al. [41] used the meta-model (Eclipse Ecore[12]) approach of a multi-perspective process that separates the interaction of people and things in the VM for IoT applications based on cloud services.

Still, in the context of smart homes, Cetina et al. [56] proposed an approach based on the dynamic SPL applied in autonomic computing systems in the context of *Smart Homes*[13] through feature models with runtime dynamic variabilities. Santos and Machado [64] proposed a dynamic SPL architecture model to obtain specific contexts and reconfigurations in runtime from different data collected in order to investigate *When or how to adapt?* (e.g., IoT or another smart environment). This model was illustrated based on the *Smart Home System* dynamic SPL. Besides, Nešković and Matić [69] designed the contextual model of self-adaptive systems with an architectural model and meta-models

[10] Signavio site: https://www.signavio.com.
[11] Pure Systems site (pure::variants tool): https://www.pure-systems.com/.
[12] Eclipse Ecore tools site: https://www.eclipse.org/ecoretools/.
[13] Autonomic Homes site: http://www.autonomic-homes.com.

using rules from mapped ontologies and feature models for IoT systems through dynamic SPL.

In the scenario of IoT monitoring systems, Amja et al. [46] developed a semantic approach from the RCA model based on feature modeling for the contextual VM activity through SPL engineering. Besides, Pessoa et al. [37] proposed an approach to evaluate reliability and maintenance problems in dynamic SPLs in the BSN domain using *domain-specific language* (DSL). Moritani and Lee [65] developed the *DisFM* approach to dynamically manage distributed feature units using guidelines to self-adaptative systems in IoT, considering: (i) child of different feature units; (ii) optional/alternative features; and (iii) child features for multiples features.

Regarding IoT monitoring systems and WSNs, Venčkauskas et al. [53–70] proposed an approach based on the *model-driven approach* (MDA) to perform the management of *body area network* (BAN) systems, as well as modeling IoT units through the feature-based framework that validated context-aware configurations of these IoT units. In another study by Venčkauskas et al. [38] a framework based on reference architecture to model IoT systems for PHM was used for the smart health environment. Weckesser et al. [74] developed an approach with criteria to evaluate WSNs in CFMs for IoT systems. Besides, Cecchinel et al. [61] designed an approach for applying SPL concepts to manage and compose data collection policies in shared sensing infrastructures, managing five main elements: controller, memory, sensors, communication, and power supply, as well as develop a representation of distinct network topologies in an automated platform of sensors. Hosoai et al. [67] specified a method to derive drivers for small appliances. The driver's generation is based on the design represented by the UML/MARTE[14] model according to the different hardware provided.

In the robotics area, Lee and Kang [35] developed a feature-oriented approach used in SPL to reconfigure variabilities dynamically in runtime through *home service robot* (HSR) SPL in the context of customizing robots. Heikkilä et al. [51] defined a reusable data model approach to store semantic information of derived customized configurations for robotized systems. Moreover, Yang and Zhang [19] implemented an approach that follows a process to VM based on the FODA method to provide management of object detection by robots.

Beyond the approaches mentioned for specific smart environments in IoT, people and things can be managed as done in Ortiz et al.'s [18] study, which created a mechanism to manage dynamic SPL variabilities in the context information of *wireless sensor-actuator network* (WSAN) for museum guidance products. Logre et al. [68] proposed an approach and a tool developed in Java using AmCharts[15] to provide dashboard customization using VM in a smart campus within universities based on model composition techniques. Achtaich et al. [52] also proposed a framework for WSAN for smart monitoring, controlling, multi-instantiating, and reconfiguring devices in runtime. In the agriculture sector, Köksal and Tekinerdogan [54] proposed the IoT-based FMIS reference architecture design approach based on several reference architectures for FMIS. This architecture adopted the SPL domain and application engineering applied in the data acquisition, application, and business layers view of the basic IoT reference architecture, considering the FMIS scenario.

In the automotive sector, Mauro et al. [49], Damiani et al. [83] and Røst et al. [50] proposed approaches applied in IoT systems to manage things. Damiani et al. [83] proposed *abstract behavioral specification* (ABS) as the toolchain extension of the *HyVar* project and *HyVarRec* (hybrid variability reconfiguration engine), mentioned in [49, 50] as well as in Section 5.1. Damiani et al.'s [83] approach to checking problems in a large number of variants was supported by Delta-*oriented programming* (DOP) for SPL engineering. Furthermore, Di Cola et al. [63] proposed a FX-MAN component model for the IoT logical architectures (directly related to the automotive software standard AUTOSAR[16]) specifying variation points to customize IoT products based on a family of systems (e.g., automotive sector). In addition, Tekkalmaz et al. [42] developed a GAMMA tool suite and smart environment development lifecycle. First, the development lifecycle divided into five layers was presented. The GAMMA tool implements these layers using the SPL paradigm (design-time and runtime) and the terms defined in their process. GAMMA was implemented as a web application using RESTful API, NodeJS, MongoDB, and ReactJS. This tool includes data and service types, features, and other configurations to facilitate smart environment management for IoT systems. Sharif et al. [39] presented the bus tracking mobile application applied in real-time to a university campus. This project was developed using a GPS tracker inside a bus, PhoneGap to implement the mobile application to cross devices, Raspberry PI for transmitting data, and the Agile method supported by SPL engineering to provide reusable requirements.

Frameworks to integrate systems of third-party applications were developed for IoT, as done by Tomlein and Grønbæk [79,43] who designed application models and annotations through Notation 3[17] language for VM related to *embedded open source software ecosystems.* In this context, Anon et al. [10] proposed a framework for end-users that generates new applications for IoT using end devices (e.g., Arduino) in the context of the cloud computing paradigm infrastructure. In this sense of cloud computing, Munoz et al. [80,81,82] worked on managing energy effectively concerning software sustainability directed to developers. Thus, these studies designed the HADAS variability model, which contains tree variability structure layers adapting variation points and variants for managing different contexts of energy consumption. Marimuthu and Chandrasekaran [20] also investigated energy-aware self-adaptive systems but used the FODA models for VM based on the contextual information obtained for IoT systems to divide and measure distinct energy efficiency levels. Even in this scenario, Murwantara et al. [71] proposed an adaptive sensor-cloud model considering the IoT infrastructure and its possible sensors in the communication of the cloud computing paradigm.

The following section presents some threats to the validity of this SLR for gathering evidence to elucidate the positive or negative growth of SPL and IoT paradigms in executing new SLRs.

6. Threats to validity

The main threats to the validity of this SLR were identified as its limitations and are discussed in this section.

Construct validity. Encompass design and generalization capacity of the study [23]. In the literature, there are several definitions of IoT. We minimize such a threat by presenting a succinct background on such a paradigm to mitigate inaccurate assumptions or definitions on the IoT. Generic keywords for "LPS" and "IoT" were used in this study. We do not adopt specific terms as "web of things" or variations so as not to distort or interfere in the accuracy and quality of the study. However, of the generic keywords, a small number of studies were selected from digital databases, which allowed us to present gaps that we considered useful for the scientific community. Only three RQs were defined and improved for conducting this SLR. Such RQs provided ways to identify and discuss gaps between SPL and IoT. Sub-questions may be added in a new SLR to identify new researches involving VM and the approaches/frameworks/platforms in the SPL and IoT areas. Furthermore, new RQs can be defined in the connected areas of IoT as cyber-physical systems.

Internal validity. Ability to describe the stages to conduct the study [23]. Concerning the search for studies, six data sources were selected to be considered essential by the SPL and IoT community. No quality criteria were applied to this selection. Such data sources were selected

[14] UML/MARTE site: http://www.omg.org/omgmarte.
[15] AmCharts site: https://www.amcharts.com.
[16] AUTOSAR site: https://www.autosar.org/.
[17] Notation 3 site: https://www.w3.org/DesignIssues/Notation3.html.

for convenience. However, more data sources may be selected for a possible update to this SLR (e.g., Wiley Interscience), in addition to the inclusion of secondary studies (systematic mappings, systematic literature reviews, and surveys). A manual search of studies executed at conferences sites also will be performed in the future. No pilot study was conducted to test and validate the generic search query defined in this SLR. We believe a pilot study is unnecessary due few studies that the generic keywords have retrieved from digital databases. However, if the pilot study were performed, perhaps new words could minimize the quality of the retrieved studies. The keywords were adapted for search in each search database mechanism.

Conclusion validity. The ability to describe correct conclusions concerning the study performed [23]. Systematic inclusion and exclusion criteria help to select papers to discuss appropriate results based on RQs. Therefore, we believe our results and conclusions are valid according to systematic review criteria and process. Peer review may improve the quality of this study and mitigate this threat related to efforts in the data collection and classification phases.

External validity. Addresses factors about research causal relationship (evolution and replications) [23]. We may not guarantee all studies were retrieved due to the fact that each data source (digital database) was different at the time of the search. Unknown technique instabilities could have occurred internally in these data sources, which could have impacted the correct recovery of these studies. This fact may be considered a threat in this SLR, due to the generation of publication bias. Another threat is the replicate ability of this SLR taking data source instabilities into consideration. We may not guarantee the quality of searches on such data sources, but we believe in the research method stages employed in this SLR.

7. Conclusions and future work

We pointed out the studies that did not solve the problem space and the main gaps related to “things” management even upon applying the SPL concepts. No related systematic reviews or mappings were identified up to this time. Therefore, we contributed by discussing how the SPL and VM were applied in IoT, presenting new research opportunities. In this study, we identified some gaps of SPL in IoT, as well as, relevant primary researches that apply SPL for variability management in IoT systems.

This SLR was conducted up to 12 March 2019 with six data sources (digital databases), from which 56 studies were selected. From the results obtained, three RQs were answered considering how SPL is applied, how VM is carried out, and which approaches/frameworks/platforms are contained in each study. The approaches and frameworks proposed in the selected studies present positive evidence for the adaptation of SPLs in the context of IoT. This finding revealed the existence of a diverse set of simple SPLs (29 studies) and approaches, frameworks or platforms that attempt to apply VM in different smart environments (final set of 53 studies). Considering an overview, the concept of VM in most approaches uses the representation of feature models for SPL and CVL, and some studies applied the UML component and class diagrams. The results also revealed some information available to facilitate the operation of the VM mechanism (e.g., variation points, variants, and constraints). However, few of them detail the process or the step-by-step operation to support their adaptation/extension in new researches or applications in industry. No guides were found to present steps or ways to apply VM activity in smart environments in IoT systems.

This SLR revealed that most studies, classified according to Wieringa et al. [30] as validation researches and solution proposals, use less rigorous evaluations in the studies. Furthermore, few studies were applied in the industry and evaluated empirically considering several smart environments using SPL in IoT systems. One case study was performed in the context of smart cities to allow control of street lamps and may be used by industrial smart parking systems. Besides, Bosch XDK devices have been investigated in the industrial sensing system to reuse monitoring IoT systems through variation points and may be useful in new smart environments.

The main gaps about how IoT may manage VM in heterogeneous devices included feature VM optimization, VM reconfiguration of robotized systems, VM of BAN systems, VM dashboard design,VM drivers using feature models, dynamic VM in SCS, knowledge and business rule variabilities, checking problems in large number of variants, lack of trade-off to VM in IoT, and VM third-party installation and integration in IoT.

Furthermore, we identified gaps related to context management in smart environments involving VM modeling constraints, contextual applications, energy-aware self-adaptive systems and consumption variants, VM in context-aware systems, lack of a multi-perspective process to VM, modeling of functionalities in embedded open-source software ecosystems, and VM in cross-organization business requirements and process.

These gaps indicated the growing interest, as well as, low maturity of the area supported by solutions that are shown in the literature as being in the development phase or finalized without applied results. As it is an emerging area, new researches are being conducted and will be consolidated over the next few years.

From the gaps presented in the primary studies, we believe that new SPLs may be proposed in some specific smart environments considering different types of IoT systems, including *autonomic software product line, software ecosystems, embedded software ecosystems, personalized health monitoring, wireless sensor-actuator network*, and *variability management of body area network*. It is worth noting that relevant smart environments such as smart energy, smart farming, and smart business have few relevant research.

This SLR may be used as a basis for guiding further revisions to investigate in detail how VM is being optimized, its impacts, as well as, possible software inspection techniques to detect defects in IoT systems management with SPL.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Appendix A. Selected primary studies

Table 6 groups the final set of primary studies ordered by year, including RQs, Authors, Title, Year, Research type, Data source, Publication type, and Publication venue.

Table 6
Final set of selected primary studies based on the inclusion and exclusion criteria (ordered by year).

RQ(s)	Author(s)	Title	Year	Research type	Data source	Publication type	Publication venue
RQ1, RQ2, and RQ3	Lee and Kang [35]	A Feature-Oriented Approach to Developing Dynamically Reconfigurable Products in Product Line Engineering	2006	Solution Proposal	IEEE Xplore	Conference	SPLC
RQ1, RQ2, and RQ3	Cetina et al. [56]	Autonomic Computing through Reuse of Variability Models at Runtime: The Case of Smart Homes	2009	Experience Paper	IEEE Xplore	Journal	IEEE Computer
RQ1, RQ2, and RQ3	Ortiz et al. [18]	Runtime Variability for Dynamic Reconfiguration in Wireless Sensor Network Product Lines	2012	Solution Proposal	Scopus	Conference	SPLC
RQ1, RQ2, and RQ3	Baresi et al. [36]	Service-Oriented Dynamic Software Product Lines	2012	Experience Paper	IEEE Xplore	Journal	IEEE Computer
RQ1, RQ2, and RQ3	Gamez and Fuentes [78]	Architectural Evolution of FamiWare using Cardinality-based Feature Models	2013	Validation Research	ScienceDirect	Journal	INFSOF
RQ2 and RQ3	Anon et al. [10]	Building a Framework for Internet of Things and Cloud Computing	2014	Validation Research	IEEE Xplore	Conference	iThings
RQ2 and RQ3	Murguzur et al. [41]	On the Support of Multi-perspective Process Models Variability for Smart Environments	2014	Solution Proposal	IEEE Xplore	Conference	MODELSWARD
RQ1, RQ2, and RQ3	Logre et al. [68]	Sensor Data visualization: A Composition-based Approach to Support Domain Variability	2014	Evaluation Research	Springer Link	Conference	ECMFA
RQ1, RQ2, and RQ3	Ayala et al. [40]	Towards a CVL Process to Develop Agents for the IoT	2014	Solution Proposal	Springer Link	Conference	UCAmI
RQ1, RQ2, and RQ3	Yang and Zhang [19]	A Feature-Oriented Modeling Approach for Embedded Product Line Engineering	2015	Solution Proposal	IEEE Xplore	Conference	FSKD
RQ1, RQ2, and RQ3	Ayala et al. [75]	A Software Product Line Process to Develop Agents for the IoT	2015	Validation Research	Compendex	Journal	Sensors
RQ2 and RQ3	Nešković and Matić [69]	Context Modeling based on Feature Models expressed as Views on Ontologies via Mappings	2015	Solution Proposal	Scopus	Journal	ComSIS
RQ1, RQ2, and RQ3	Kneer and Kamsties [60]	A Case Study on Self-configuring Systems in IoT Based on a Model-Driven Prototyping Approach	2016	Evaluation Research	Springer Link	Conference	ICIST
RQ2 and RQ3	Venčkauskas et al. [38]	A Model-Driven Framework to Develop Personalized Health Monitoring	2016	Validation Research	Scopus	Journal	Symmetry
RQ1, RQ2, and RQ3	Cecchinel et al. [61]	Automated Deployment of Data Collection Policies over Heterogeneous Shared Sensing Infrastructures	2016	Validation Research	IEEE Xplore	Conference	APSEC
RQ2 and RQ3	Tomlein and Grønbæk [79]	Building Models of Installations to Recommend Applications in IoT Software Ecosystems	2016	Evaluation Research	IEEE Xplore	Conference	FiCloud
RQ1, RQ2, and RQ3	Mauro et al. [49]	Context Aware Reconfiguration in Software Product Lines	2016	Evaluation Research	Scopus	Conference	VaMoS
RQ1, RQ2, and RQ3	Heikkilä et al. [51]	Dealing with Configurability in Robot Systems	2016	Solution Proposal	IEEE Xplore	Conference	MESA
RQ2 and RQ3	Marimuthu and Chandrasekaran [20]	Feature-Oriented Domain Analysis Framework for Energy-Aware Self-Adaptive Software	2016	Solution Proposal	IEEE Xplore	Conference	iThings
RQ2 and RQ3	Amja et al. [47]	Linking Relational Concept Analysis and Variability Model within Context Modeling of Context-aware Applications	2016	Solution Proposal	IEEE Xplore	Conference	ISSE
RQ1, RQ2, and RQ3	Venčkauskas et al. [53]	Model-Driven Approach for Body Area Network Application Development	2016	Validation Research	Scopus	Journal	Sensors
RQ1, RQ2, and RQ3	Venčkauskas et al. [70]	Modeling of Internet of Things Units for Estimating Security-Energy-Performance Relationships for Quality of Service and Environment Awareness	2016	Validation Research	Scopus	Journal	SEC
RQ2 and RQ3	Tomlein and Grønbæk [43]	Semantic Model of Variability and Capabilities of IoT Applications for Embedded Software Ecosystems	2016	Evaluation Research	IEEE Xplore	Conference	WICSA
RQ1, RQ2, and RQ3	Di Cola et al. [63]	Towards Defining Families of Systems in IoT: Logical Architectures with Variation Points	2016	Solution Proposal	Springer Link	Conference	IoT 360º
RQ2	McGee and McGregor [48]	Using Dynamic Adaptive Systems in Safety-critical Domains	2016	Solution Proposal	ACM Digital Library	Conference	SEAMS
RQ1, RQ2, and RQ3	Ayala et al. [76]	Using Models at Runtime to Adapt Self-managed Agents for the IoT	2016	Validation Research	Springer Link	Conference	MATES
RQ1, RQ2, and RQ3	Ayala et al. [77]	Using SPL to Develop AAL Systems Based on Self-adaptive Agents	2016	Validation Research	Springer Link	Conference	PAAMS
RQ2 and RQ3	Jazayeri et al. [62]	A Variability Model for Store-Oriented Software Ecosystems: An Enterprise Perspective	2017	Evaluation Research	Springer Link	Conference	ICSOC

(*continued on next page*)

Table 6 (*continued*)

RQ(s)	Author(s)	Title	Year	Research type	Data source	Publication type	Publication venue
RQ2 and RQ3	Munoz [80]	Achieving energy efficiency using a software product line approach	2017	Validation Research	Scopus	Conference	SPLC
RQ1, RQ2, and RQ3	Moritani and Lee [65]	An Approach for Managing a Distributed Feature Model to Evolve Self-Adaptive Dynamic Software Product Lines	2017	Validation Research	ACM Digital Library	Conference	SPLC
RQ2 and RQ3	Damiani et al. [83]	An Extension of the ABS Toolchain with a Mechanism for Type Checking SPLs	2017	Evaluation Research	Springer Link	Conference	IFM
RQ1, RQ2, and RQ3	Abbas et al. [55]	Binary Pattern for Nested Cardinality Constraints for Software Product Line of IoT-Based Feature Models	2017	Validation Research	IEEE Xplore	Journal	IEEE Access
RQ1, RQ2, and RQ3	Pessoa et al. [37]	Building reliable and maintainable Dynamic Software Product Lines: An investigation in the Body Sensor Network domain	2017	Validation Research	ScienceDirect	Journal	INFSOF
RQ2 and RQ3	Sinnhofer et al. [57]	Combined Variability Management of Business Processes and Software Architectures	2017	Evaluation Research	Compendex	Conference	BMSD
RQ1, RQ2, and RQ3	Amja et al. [46]	Combining Variability, RCA and Feature Model for Context-Awareness	2017	Solution Proposal	Scopus	Conference	INTECH
RQ2 and RQ3	Abbas et al. [44]	Contextual Variability Management of IoT Application with XML-based Feature Modelling	2017	Solution Proposal	Scopus	Journal	JATIT
RQ1 and RQ2	McGee et al. [66]	Designing for Reuse in an Industrial Internet of Things Monitoring Application	2017	Evaluation Research	Scopus	Conference	WASHES
RQ2 and RQ3	Munoz et al. [81]	Green software development and research with the HADAS toolkit	2017	Validation Research	Scopus	Conference	ECSA
RQ2 and RQ3	Munoz et al. [82]	HADAS and web services: Eco-efficiency assistant and repository use case evaluation	2017	Validation Research	IEEE Xplore	Conference	ES2DE
RQ2 and RQ3	Mens et al. [45]	Modeling and Managing Context-Aware Systems' Variability	2017	Experience Paper	Scopus	Journal	IEEE Software
RQ1, RQ2, and RQ3	Hosoai et al. [67]	System Product Line Engineering for Small Appliances with Driver Derivation	2017	Solution Proposal	IEEE Xplore	Conference	APSEC
RQ2 and RQ3	Sinnhofer et al. [58]	Combining Business Process Variability and Software Variability Using Traceable Links	2018	Evaluation Research	Springer Link	Conference	BMSD
RQ2 and RQ3	Suri et al. [59]	Configurable IoT-Aware Allocation in Business Processes	2018	Evaluation Research	Springer Link	Conference	SCC
RQ1, RQ2, and RQ3	Achtaich et al. [52]	Designing a Framework for Smart IoT Adaptations	2018	Solution Proposal	Springer Link	Conference	EAI/AFRICATEK
RQ2 and RQ3	Røst et al. [50]	HyVar: Scalable Hybrid Variability for Distributed Evolving Software Systems	2018	Solution Proposal	Springer Link	Conference	ESOCC
RQ1, RQ2, and RQ3	Abbas et al. [73]	Multi-Objective Optimum Solutions for IoT-Based Feature Models of Software Product Line	2018	Validation Research	IEEE Xplore	Journal	IEEE Access
RQ1, RQ2, and RQ3	Weckesser et al. [74]	Optimal Reconfiguration of Dynamic Software Product Lines Based on Performance-influence Models	2018	Evaluation Research	ACM Digital Library	Conference	SPLC
RQ1, RQ2, and RQ3	Santos and Machado [64]	Towards an Architecture Model for Dynamic Software Product Lines Engineering	2018	Validation Research	IEEE Xplore	Conference	IRI
RQ1, RQ2, and RQ3	Smiari and Bibi [72]	A Smart City Application Modeling Framework: A Case Study on *Re*-Engineering a Smart Retail Platform	2018	Evaluation Research	Scopus	Conference	SEAA
RQ2 and RQ3	Sharif et al. [39]	Real-Time Campus University Bus Tracking Mobile Application	2018	Evaluation Research	Scopus	Conference	ICT-ISPC
RQ2 and RQ3	Murwantara et al. [71]	Towards Adaptive Sensor-Cloud for Internet of Things	2018	Validation Research	Scopus	Journal	Telkomnika
RQ1, RQ2, and RQ3	Köksal and Tekinerdogan [54]	Architecture Design Approach for IoT-based Farm Management Information Systems	2018	Evaluation Research	Scopus	Journal	Precision Agriculture
RQ2 and RQ3	Tekkalmaz et al. [42]	An Engineering Tool Suite for Enhanced Smart Environments: Gamma	2018	Validation Research	Scopus	Conference	CEUR

Appendix B. Synthesized answers of each RQ

We present the synthesized answers of each RQ side by side in Table 7, based on the analysis and synthesis of the 53 selected primary studies.

Table 7
Research questions synthetized answered corresponding to each selected primary study.

Paper (author/title)	RQ1: How are software product lines (SPL) being applied in the context of Internet of Things (IoT) systems? Presents the SPL related to each study.	RQ2: How is the variability management (VM) of SPL carried out in IoT systems? Presents specific locations that VM occurs.	RQ3: Which approaches, frameworks or platforms use SPL in IoT systems? Presents the proposals identified in each study.
Kneer and Kamsties [60]/A Case Study on Self-configuring Systems in IoT Based on a Model-driven Prototyping Approach	*Public smart street light* feature model	Apply VM in *Self-configuring systems* (SCS) for IoT	Unnamed framework for SCS for IoT was proposed in a case study with the application of VM based on a small evaluation of the p*ublic smart street light* feature model.
Lee and Kang [35]/A Feature-oriented Approach to Developing Dynamically Reconfigurable Products in Product Line Engineering	*Home service robot* (HSR) SPL feature model	The HSR SPL illustrated the proposed approach and allowed the dynamic reconfiguration of product features from variation points management	Unnamed dynamic approach used in SPL at runtime during monitoring and based on five guidelines applied to the design of the architecture.
Yang and Zhang [19]/A Feature-oriented Modeling Approach for Embedded Product Line Engineering	*Ventilator embedded* SPL feature model to control basic operations of a ventilation system	Follows a process with eight basic activities to specify in detail the VM in the modeling of features using the FODA method	Unnamed approach.
Ayala et al. [75]/A Software Product Line Process to Develop Agents for the IoT	*Smart shopping center* SPL specifies and illustrates the functionality of the process based on SPL	*Self-StarMAS* process was implemented using CVL and has a mechanism that allows the reuse of architectural components for the development of *Multi-Agent Systems* (MAS) in IoT	*Self-StarMAS* process using agents in ambient intelligence for IoT systems.
Moritani and Lee [65]/An Approach for Managing a Distributed Feature Model to Evolve Self-adaptive Dynamic Software Product Lines	*Baby care system* (BCS) partial feature model was proposed to monitor the baby while sleeping to exemplify *DisFM* approach	For VM, the guidelines of the *DisFM* were specified: (i) child of different feature units, (ii) optional/alternative features, and (iii) child features for multiples features	*DisFM* approach with guidelines aimed to manage distributed feature units in feature models.
Gamez and Fuentes [78]/Architectural Evolution of FamiWare using Cardinality-based Feature Models	*FamiWare* system family based on *Cardinality-based Feature Models*	Allows different features customization of configurations and constraints included in the *FamiWare*	*VML4FamiWare* language performs the VM activity.
Cecchinel et al. [61]/Automated Deployment of Data Collection Policies over Heterogeneous Shared Sensing Infrastructures	*Unnamed SPL*, but several feature models were adapted	Performed automatically through feature models adaptation. Thus, such management was represented as a *shared sensing infrastructures* example by means of five main infrastructures components: controller, memory, sensors, communication, and power supply	The unnamed approach applies SPL concepts to manage and compose data collection policies as well as the representation of network topologies based on an automated platform.
Cetina et al. [56]/Autonomic Computing through Reuse of Variability Models at Runtime: The Case of Smart Homes	*Smart-home* dynamic SPL feature model	Applies feature models with runtime variability in the development of autonomous systems for *Smart Home* architectures	The unnamed approach based on dynamic SPL applied in autonomic computing.
Abbas et al. [55]/Binary Pattern for Nested Cardinality Constraints for Software Product Line of IoT-based Feature Models	*Pulse sensor* IoT feature model was used to illustrated and evaluate BPNCC approach	BPNCC approach was proposed through five steps to calculate the number of combinations between features using a binary standard (0 - not selected feature; 1 - selected) with the following main commons relationships in the VM of SPL: mandatory, optional, alternative and OR	*binary pattern for nested cardinality constraints* (BPNCC) approach.
Abbas et al. [73]/Multi-Objective Optimum Solutions for IoT-based Feature Models of Software Product Line	Three feature models called *security settings*, *speech recognition*, and *BerkeleyDB* feature models were used to evaluate proposed approach MOO-BPNCC	Presents an extended approach BPNCC applying *multi-objective optimum-BPNCC* (MOO-BPNCC) containing three paths (A, B, and C) to optimize features models	*Multi-objective optimum-BPNCC* (MOO-BPNCC).
Pessoa et al. [37]/Building reliable and maintainable Dynamic Software Product Lines: An investigation in the Body Sensor Network domain	*Body sensor network* (BSN) dynamic SPL feature model	Compared variability approaches and used DSL specification to develop plans (instances) and evaluate this dynamic SPL in the BSN domain, do not describe the VM in detail	Unnamed approach for the implementation and evaluation of reliability and maintenance problems in dynamic SPLs in the BSN domain.

(continued on next page)

Table 7 (*continued*)

Paper (author/title)	RQ1: How are software product lines (SPL) being applied in the context of Internet of Things (IoT) systems? Presents the SPL related to each study.	RQ2: How is the variability management (VM) of SPL carried out in IoT systems? Presents specific locations that VM occurs.	RQ3: Which approaches, frameworks or platforms use SPL in IoT systems? Presents the proposals identified in each study.
Amja et al. [46]/Combining Variability, RCA and Feature Model for Context-awareness	*Simplified health monitoring system* feature model was used to presents the proposed approach	Apply relational concept analysis (RCA) and feature modeling with contextual VM by means of SPL engineering	Semantic approach in which model RCA and feature modeling were combined.
Mauro et al. [49]/Context Aware Reconfiguration in Software Product Lines	*Car* dynamic *SPL* feature model	*HyVarRec* hybrid approach consists of VFs that apply constraints on the selection of specific features in contextual information with the main objective of dynamically reconfiguring dynamic SPL for the automotive sector	*HyVarRec*, a hybrid variability reconfiguration engine applied to configurations for feature models.
Røst et al. [50]/HyVar: Scalable Hybrid Variability for Distributed Evolving Software Systems	–	This paper highlights the *HyVar Project* as a solution to variability management in IoT	*HyVar Project* approach, combining initially a framework and d*omain-specific variability language* (DSVL) as well as a toolchain for distributed applications focusing on the automotive sector.
Heikkilä et al. [51]/Dealing with Configurability in Robot Systems	*Object detection* represented by UML component diagram demonstrates the application of the proposed approach	The proposed approach, based on *systems and software product line* (SSPL), performs the VM in the reconfiguration of robotized systems	An unnamed approach that uses the proposed reusable data model to store semantic information of derived configurations in robotized systems.
Achtaich et al. [52]/Designing a Framework for Smart IoT Adaptations	*Smart irrigation system* dynamic SPL is presented as a toy example to explain the proposed framework	For the management of such devices, the following steps were defined in a framework: (i) the d*evice management* (DM) platform; (ii) defined dimensions such as system, context, and environment; and (iii) selected dynamic SPL	Unnamed framework for smart monitoring, control, multi-instantiation, and reconfiguration devices in runtime by means of dynamic SPL to IoT systems.
Venčkauskas et al. [53]/Model-driven Approach for Body Area Network Application Development	*Smart sensor controller* generates a family of systems consisting of BAN sensor controllers to measure the *quality-of-service* (QoS)	For VM, consider their relationships mandatory, optional, and alternative features and a meta-specification by means of specialized, abstract, and concrete features related to variation points or variants	Unnamed approach based on *model-driven approach* (MDA) to perform the management of BAN systems.
Venčkauskas et al. [70]/Modeling of Internet of Things Units for Estimating Security-energy-performance Relationships for Quality of Service and Environment Awareness	*securitySettings* partial feature model was used in experimental evaluations of modeled IoT units	A set of relationships was specified in a framework through a model with valid context-aware configurations of IoT units and supported by QoS evaluations	Unnamed framework was proposed for feature-based modeling of IoT units.
Ortiz et al. [18]/Runtime Variability for Dynamic Reconfiguration in Wireless Sensor Network Product Lines	*Indoor smart environmental and guidance systems* of WSAN product family were modeled based on FODA feature model to explain the approach proposed	To manage changes, the structure of variabilities was created at runtime dynamically based on a mechanism to manage dynamic SPL variabilities in the context information of WSAN	Unnamed approach.
Logre et al. [68]/Sensor Data Visualization: A Composition-based Approach to Support Domain Variability	*SmartCampus* feature model to manage the variability of widgets in the generation of new dashboards to *SmartCampus* project	Management was based on model composition techniques to allow the design of monitoring dashboards considering their variabilities	Unnamed approach was proposed, and a tool developed in Java using AmCharts was employed to implement this approach for dashboard customization.
Baresi et al. [36]/Service-oriented Dynamic Software Product Lines	*Smart home* feature model for home automation	The CVL performs the features management and process execution of a dynamic SPL with BPEL	Unnamed approach based on dynamic SPL, CVL, and *business process execution language* (BPEL).
Hosoai et al. [67]/System Product Line Engineering for Small Appliances with Driver Derivation	*Motion tracking system product family* was modeled as an example to present the proposed method	This method uses feature models to drivers VM. It is limited in the driver's generation based on the design represented by the model UML/MARTE and information provided by different hardware	Unnamed method was proposed to systematically derive drivers for small appliances.
Di Cola et al. [63]/Towards Defining Families of Systems in IoT: Logical Architectures with Variation Points	*Vehicle control systems product family*	Proposes a set of specified variation points for the VM based on features model	FX-MAN component model for specification of logical architectures.

(*continued on next page*)

Table 7 (*continued*)

Paper (author/title)	RQ1: How are software product lines (SPL) being applied in the context of Internet of Things (IoT) systems? Presents the SPL related to each study.	RQ2: How is the variability management (VM) of SPL carried out in IoT systems? Presents specific locations that VM occurs.	RQ3: Which approaches, frameworks or platforms use SPL in IoT systems? Presents the proposals identified in each study.
Ayala et al. [40]/Towards a CVL Process to Develop Agents for the IoT	*Smart shopping center* SPL	A *VSpec* tree was defined and used to support management in the variabilities resolution process in IoT systems, including VM of self-adapted agents for a*mbient assisted living* (AAL)	*Self-StarMAS* process.
(i) Ayala et al. [76]/Using Models at Runtime to Adapt Self-managed Agents for the IoT and (ii) Ayala et al. [77]/Using SPL to Develop AAL Systems Based on Self-adaptive Agents	*Smart shopping center* SPL	The *Self-StarMAS* process applies dynamic adaptation of agents in runtime considering its behavior according to changes in different environments using a variability model that applies MAPE-K	*Self-StarMAS* process was evaluated and improved with self-management properties considering *VSpec* tree and combining the *Self-StarMAS* process CVL with MDD.
Weckesser et al. [74]/Optimal Reconfiguration of Dynamic Software Product Lines Based on Performance-influence Models	Use adaptive WSNs *context feature model* (CFM) example as a case study in IoT to propose an approach regarding consistent properties, optimal reconfigurations decisions, and performance influences of configurations using as core dynamic SPL-based runtime	This approach was defined considering flexibility and context performance goals, as well as effectiveness, efficiency, and applicability criteria in evaluation using WSNs CFM in IoT case study. This approach also learns performance-influence models in design-time after implementation and execution of new runtime representation	Unnamed approach.
Santos and Machado [64]/Towards an Architecture Model for Dynamic Software Product Lines Engineering	The only example called *smart home system* domain was designed (Java platform, MQTT broker, and OSGi framework) to demonstrate the dynamic SPL architecture model proposed according to MAPE-K model activities to enable reconfigurations in runtime	This dynamic SPL architecture model investigated *When or How to adapt?* specific software owing to gaps based on context and actions to adapt (e.g., IoT or smart scenario), which is divided into three main steps. A set of technologies were used, such as Arduino, Raspberry Pi, and JSON, for the development of *smart home systems*. Thus, it was possible to illustrate this dynamic SPL architecture model to obtain the specific context and reconfigurations in runtime from the different information collected	Unnamed dynamic SPL architecture model.
McGee et al. [66]/Designing for Reuse in an Industrial Internet of Things Monitoring Application	A monitoring system feature model that uses specific protocols, Bosch XDK devices, line configuration, and data points	McGee et al. [66] designed variation points and observed the missed requirements, and attempted to mitigate them with ALISA tools (architecture analysis & design language, AADL)	–
Smiari and Bibi [72]/A Smart City Application Modeling Framework: A Case Study on *Re*-engineering a Smart Retail Platform	Only presents a simple *smart retail platform* feature model in a case study to initially evaluate SCAMF framework proposed	*Smart city application modeling framework* (SCAMF) provided a variability mechanism with several relationships such as mandatory, optional, more-of (OR), and one-of (alternative), used with rules to transform a feature model in a UML class diagram	SCAMF using feature models and class diagrams.
Köksal and Tekinerdogan [54]/Architecture Design Approach for IoT-Based Farm Management Information Systems	Presents two feature models as case studies named smart tomato production and smart wheat production	The VM activity is performed using feature domain analysis employing basic relationships such as mandatory, optional, alternative, which were adapted according to the reference architecture, and each application feature model can be generated based on smart production environment in an IoT-based FMIS	The IoT-based FMIS reference architecture design approach. This architecture adopted SPL domain and application engineering. Thus, this architecture was also developed considering the decomposition view (e.g., modules and sub-modules containing different feature types) and deployment view (e.g., software to the hardware deployed in a cloud server and client (farmer)).
McGee and McGregor [84]/Using Dynamic Adaptive Systems in Safety-critical Domains	–	A set of variation points were specified in the attempt of solving the safety-critical systems' variability in runtime. However, no approach was presented to solve these problems as well as limitations	–
Venčkauskas et al. [38]/A Model-driven Framework to Develop Personalized Health Monitoring	–	*IoT nodes* (IoTN) feature model was presented to illustrate the proposed framework	Model-oriented framework used VM and feature models in reference architecture modeling and IoT systems prototyping for *personalized health monitoring* (PHM).

(*continued on next page*)

Table 7 (*continued*)

Paper (author/title)	RQ1: How are software product lines (SPL) being applied in the context of Internet of Things (IoT) systems? Presents the SPL related to each study.	RQ2: How is the variability management (VM) of SPL carried out in IoT systems? Presents specific locations that VM occurs.	RQ3: Which approaches, frameworks or platforms use SPL in IoT systems? Presents the proposals identified in each study.
Jazayeri et al. [62]/A Variability Model for Store-oriented Software Ecosystems: An Enterprise Perspective	–	A variability model was proposed as an approach using the *orthogonal variability model* (OVM) to represent the variabilities in *store-oriented software ecosystems* SPL. For example, this variability model analyzed IoT systems such as smart voice assistant named Amazon Alexa	Unnamed variability model.
Damiani et al. [83]/An Extension of the ABS Toolchain with a Mechanism for Type Checking SPLs	–	*Abstract behavioral specification* (ABS) modeling language and the toolchain allied to the support provides *delta-oriented programming* (DOP) to check DOP SPLs	ABS toolchain extension proposed as a solution applied to check problems owing to the large number of variants that an SPL can contain.
Anon et al. [10]/Building a Framework for the Internet of Things and Cloud Computing	–	This study introduces the idea of a framework that applies the VM in SPL to IoT applications	Unnamed framework.
Tomlein and Grønbæk [79]/Building Models of Installations to Recommend Applications in IoT Software Ecosystems	–	This study presents VM problems in the installation and integration of third-party applications in *embedded software ecosystems* (ESE) for IoT	An approach was proposed to allow the implementation of specific models and installation types as well as specifying requirements from the users' point of view.
Sinnhofer et al. [57]/Combined Variability Management of Business Processes and Software Architectures	–	A *process variability framework* project in an attempt to facilitate the communication between variabilities and business processes by means of a combined variability approach to customization SPLs. Thus, this study used SPLE tool pure::variants to investigate business variability	*Process variability framework.*
Sinnhofer et al. [58]/Combining Business Process Variability and Software Variability Using Traceable Links	–	This study is a continuation of the previous work of Shinnhofer et al. [57] and the main difference is that the authors discussed a possible improvement of the approach proposed. Furthermore, the modeling constraints were studied and analyzed to combine few improvements obtained in this study. For instance, domain experts can use this combined approach to minimize development costs and time-to-market	*Process variability framework.*
Nešković and Matić [69]/Context Modeling based on Feature Models expressed as Views on Ontologies via Mappings	–	Proposes an approach with an architectural model and meta-models with rules to map ontologies and feature models in these systems in dynamic SPL	Unnamed approach proposed for the contextual modeling of self-adaptive systems.
Abbas et al. [44]/Contextual Variability Management of IoT Application with XML-based feature modelling	–	An approach was proposed using XML-based feature models for reuse to resources and functionalities. It is possible to select features for generating IoT applications from domain modeling performed through an XML-*schema meta-model* and XSLT files	Unnamed approach.
Marimuthu and Chandrasekaran [20]/Feature-oriented Domain Analysis Framework for Energy-aware Self-adaptive Software	–	The domain analysis framework was proposed for the FODA model VM based on contextual information for *energy-aware self-adaptive systems*. The context was divided into energy efficiency levels to avoid excessive energy consumption and SPL basic relationships features were considered in the model	Unnamed domain analysis framework.
(i) Munoz [80]/Achieving Energy Efficiency using a Software Product Line Approach; (ii) Munoz et al. [81]/Green Software Development and Research with the HADAS Toolkit; and (iii) Munoz et al. [82]/HADAS and Web Services: Eco-efficiency Assistant and Repository Use Case Evaluation	–	HADAS variability model approach has tree variability structure layers according to concepts of variation points and variants used for VM in different contexts of energy consumption	HADAS variability model.

(*continued on next page*)

Table 7 (*continued*)

Paper (author/title)	RQ1: How are software product lines (SPL) being applied in the context of Internet of Things (IoT) systems? Presents the SPL related to each study.	RQ2: How is the variability management (VM) of SPL carried out in IoT systems? Presents specific locations that VM occurs.	RQ3: Which approaches, frameworks or platforms use SPL in IoT systems? Presents the proposals identified in each study.
Amja et al. [47]/Linking Relational Concept Analysis and Variability Model within Context Modeling of Context-aware Applications	–	Applied RCA and feature modeling with contextual VM through SPL engineering	Unnamed semantic approach proposed in which the model RCA and feature modeling were combined.
Mens et al. [45]/Modeling and Managing Context-aware Systems Variability	–	Clarify challenges in dynamic behavior in modeling and VM in context-aware systems	Selected studies with variability approaches including contextual modeling, management, and variabilities implementation.
Murguzur et al. [41]/On the Support of Multi-perspective Process Models Variability for Smart Environments	–	Encompass VM activity of smart environments for IoT applications based on cloud services that involve interaction between people and things	Unnamed approach proposed by means of a meta-model (Ecore) related to the specification of a multi-perspective process that separates people and things in VM.
Tomlein and Grønbæk [43]/Semantic Model of Variability and Capabilities of IoT Applications for Embedded Software Ecosystems	–	A semantic application model was proposed to model constraints and VM considering evaluation and deployment topology through industry collaboration	Semantic application model proposed and annotations by means of Notation 3 language for VM of IoT applications in *embedded open source software ecosystems*.
Suri et al. [59]/Configurable IoT-aware Allocation in Business Processes	–	*Configurable process model* (CPM) based on *business process modeling language* (BPMN) to manage variabilities and business requirements in IoT	CPM process based on BPMN. The process proposed was extended using a Signavio open-source environment to model BPMN 2.0 process graphically.
Sharif et al. [39]/Real-Time Campus University Bus Tracking Mobile Application	–	The VM activity was adopted in the modeling of the requirements model of mobile applications to real-time bus tracking at the University of Malaysia. The main feature relationships defined were: (i) "x" as functional requirements, (ii) "P" as an optional feature, and (iii) "c" as a common feature	CVAnalysis: bus-tracking mobile application applied in real-time to University campus.
Murwantara et al. [71]/Towards Adaptive Sensor-cloud for the Internet of Things	–	The adaptive sensor-cloud model was initially modeled from feature model architecture using different sensor-type management and provided reconfiguration of sensors according to the smart environment	Proposes an adaptive sensor-cloud model considering IoT infrastructure.
Tekkalmaz et al. [42]/An Engineering Tool Suite for Enhanced Smart Environments: GAMMA	–	The variability management activity is performed using GAMMA tool	This paper presents two main contributions: a GAMMA tool suite and a smart environment development lifecycle, which emerged based on the implementation of the tool.

Appendix C. Study quality assessment

We present the data of each selected primary study in Table 8 related to study quality assessment criteria defined.

Table 8
Final set of selected primary studies based on the application of the study quality assessment criteria.

Author(s)	Title	1."Is the paper based on research (or is it a discussion paper based on expert opinion)? Yes/No."	What research method was used: Experiment, Quasi-Experiment, Lessons learned, Case study, Opinion Survey, Tertiary Study, Other (specify)? Note: This is to be based on our reading of the paper not the method claimed by the author of the paper.	Is there a clear statement of the aims of the study? Yes/ Partly/No. Score as 1, 0.5, 0. Interpolation is permitted.
Lee and Kang [35]	A feature-oriented approach to developing dynamically reconfigurable products in product line engineering	Yes	Other (simple study called Case study)	0.5 (generic objective)

(*continued on next page*)

Table 8 (*continued*)

Author(s)	Title	1."Is the paper based on research (or is it a discussion paper based on expert opinion)? Yes/No."	What research method was used: Experiment, Quasi-Experiment, Lessons learned, Case study, Opinion Survey, Tertiary Study, Other (specify)? Note: This is to be based on our reading of the paper not the method claimed by the author of the paper.	Is there a clear statement of the aims of the study? Yes/ Partly/No. Score as 1, 0.5, 0. Interpolation is permitted.
Cetina et al. [56]	Autonomic computing through reuse of variability models at runtime: the case of smart homes	No	Expert opinion	1
Ortiz et al. [18]	Runtime variability for dynamic reconfiguration in wireless sensor network product lines	Yes	Other (motivational example of the approach)	0.5 (generic objective)
Baresi et al. [36]	Service-oriented dynamic software product lines	No	Expert opinion	1
Gamez and Fuentes [78]	Architectural evolution of famiware using cardinality-based feature models	Yes	Other (simple study called case study)	1
Anon et al. [10]	Building a framework for the internet of things and cloud computing	Yes	Other (simple study called Case study)	1
Murguzur et al. [41]	On the Support of Multi-perspective Process Models Variability for Smart Environments	Yes	Other (implement a prototype of the multi-perspective process)	1
Logre et al. [68]	Sensor Data visualization: A composition-based approach to support domain variability	Yes	Experiment	1
Ayala et al. [40]	Towards a CVL process to develop agents for the IoT	Yes	Other (motivational example with a prototype)	1
Yang and Zhang [19]	A feature-oriented modeling approach for embedded product line engineering	Yes	Other (motivational example with a prototype)	1
Ayala et al. [75]	A Software Product Line Process to Develop Agents for the IoT	Yes	Other (simple study called Case study)	1
Nešković and Matić [69]	Context modeling based on feature models expressed as views on ontologies via mappings	Yes	Other (a simple example)	1
Kneer and Kamsties [60]	A case study on self-configuring systems in iot based on a model-driven prototyping approach	Yes	Other (simple study called Case study)	1
Venčkauskas et al. [38]	A model-driven framework to develop personalized health monitoring	Yes	Case study (real scenario)	1
Cecchinel et al. [61]	Automated deployment of data collection policies over heterogeneous shared sensing infrastructures	Yes	Experiment	1
Tomlein and Grønbæk [79]	Building models of installations to recommend applications in IoT software ecosystems	Yes	Case study (real scenario)	1
Mauro et al. [49]	Context-aware reconfiguration in software product lines	Yes	Case study (real scenario)	1
Heikkilä et al. [51]	Dealing with configurability in robot systems	Yes	Other (motivational running example)	0.5 (generic objective)
Marimuthu and Chandrasekaran [20]	Feature-oriented domain analysis framework for energy-aware self-adaptive software	Yes	Other (motivational running example)	1
Amja et al. [47]	Linking relational concept analysis and variability model within context modeling of context-aware applications	Yes	Other (motivational running example)	1
Venčkauskas et al. [53]	Model-driven approach for body area network application development	Yes	Other (motivational running example)	1
Venčkauskas et al. [70]	modeling of internet of things units for estimating security-energy-performance relationships for quality of service and environment awareness	Yes	Experiment	1
Tomlein and Grønbæk [43]	Semantic model of variability and capabilities of IoT applications for embedded software ecosystems	Yes	Other (motivational running example)	1
Di Cola et al. [63]	Towards defining families of systems in IoT: logical architectures with variation points	Yes	Other (motivational example with a prototype)	1
McGee and McGregor [48]	Using dynamic adaptive systems in safety-critical domains	Yes	Other (motivational example with a prototype)	1

(*continued on next page*)

Table 8 (*continued*)

Author(s)	Title	1."Is the paper based on research (or is it a discussion paper based on expert opinion)? Yes/No."	What research method was used: Experiment, Quasi-Experiment, Lessons learned, Case study, Opinion Survey, Tertiary Study, Other (specify)? Note: This is to be based on our reading of the paper not the method claimed by the author of the paper.	Is there a clear statement of the aims of the study? Yes/ Partly/No. Score as 1, 0.5, 0. Interpolation is permitted.
Ayala et al. [76]	Using models at runtime to adapt self-managed agents for the IoT	Yes	Other (motivational running example)	1
Ayala et al. [77]	Using SPL to develop AAL systems based on self-adaptive agents	Yes	Other (motivational running example)	1
Jazayeri et al. [62]	A variability model for store-oriented software ecosystems: an enterprise perspective	Yes	Experiment	1
Munoz [80]	Achieving energy efficiency using a software product line approach	Yes	Experiment	1
Moritani and Lee [65]	An approach for managing a distributed feature model to evolve self-adaptive dynamic software product lines	Yes	Experiment	1
Damiani et al. [83]	An extension of the abs toolchain with a mechanism for type checking SPLs	Yes	Case study (real scenario)	1
Abbas et al. [55]	Binary pattern for nested cardinality constraints for software product line of IoT-based feature models	Yes	Experiment	1
Pessoa et al. [37]	Building reliable and maintainable dynamic software product lines: an investigation in the body sensor network domain	Yes	Case study (real scenario)	1
Sinnhofer et al. [57]	Combined variability management of business processes and software architectures	Yes	Case study (real scenario)	1
Amja et al. [46]	Combining variability, RCA and feature model for context-awareness	Yes	Other (motivational example with a prototype)	1
Abbas et al. [44]	Contextual variability management of iot application with xml-based feature modelling	Yes	Other (motivational running example)	1
McGee et al. [66]	Designing for reuse in an industrial internet of things monitoring application	Yes	Case study (real scenario)	1
Munoz et al. [81]	Green software development and research with the HADAS toolkit	Yes	Experiment	1
Munoz et al. [82]	HADAS and web services: Eco-efficiency assistant and repository use case evaluation	Yes	Case study (real scenario)	1
Mens et al. [45]	Modeling and managing context-aware systems' variability	No	Expert opinion	1
Hosoai et al. [67]	System product line engineering for small appliances with driver derivation	Yes	Other (only presents the approach)	1
Sinnhofer et al. [58]	Combining business process variability and software variability using traceable links	Yes	Case study (real scenario)	1
Suri et al. [59]	Configurable IoT-aware allocation in business processes	Yes	Case study (real scenario)	1
Achtaich et al. [52]	designing a framework for smart IoT adaptations	Yes	Other (motivational example with a prototype)	1
Røst et al. [50]	HyVar: scalable hybrid variability for distributed evolving software systems	Yes	Case study (real scenario)	1
Abbas et al. [73]	Multi-objective optimum solutions for IoT-based feature models of software product line	Yes	Experiment	1
Weckesser et al. [74]	Optimal reconfiguration of dynamic software product lines based on performance-influence models	Yes	Case study (real scenario)	1
Santos and Machado [64]	Towards an architecture model for dynamic software product lines engineering	Yes	Experiment	1
Smiari and Bibi [72]	A smart city application modeling framework: a case study on *re*-engineering a smart retail platform	Yes	Experiment	1
Sharif et al. [39]	Real-time campus university bus tracking mobile application	Yes	Case study (real scenario)	1

(*continued on next page*)

Table 8 (*continued*)

Author(s)	Title	1."Is the paper based on research (or is it a discussion paper based on expert opinion)? Yes/No."	What research method was used: Experiment, Quasi-Experiment, Lessons learned, Case study, Opinion Survey, Tertiary Study, Other (specify)? Note: This is to be based on our reading of the paper not the method claimed by the author of the paper.	Is there a clear statement of the aims of the study? Yes/Partly/No. Score as 1, 0.5, 0. Interpolation is permitted.
Murwantara et al. [71]	Towards adaptive sensor-cloud for the internet of things	Yes	Experiment	1
Köksal and Tekinerdogan [54]	Architecture design approach for iot-based farm management information systems	Yes	Case study (real scenario)	1
Tekkalmaz et al. [42]	An engineering tool suite for enhanced smart environments: gamma	Yes	Experiment	1

References

[1] R. Capilla, J. Bosch, K.-C. Kang, Systems and Software Variability Management: Concepts, Tools and Experiences, Springer, Heidelberg, 2013, doi:10.1007/978-3-642-36583-6.
[2] F. Linden, K. Schmid, E. Rommes, Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering, Springer, 2007.
[3] J. Bosch, J. Lee, Software product lines: going beyond, in: Proc. Int. Conf. Softw. Prod. Line Conf., Jeju Island, South Korea, 2010, p. 548.
[4] P. Clements, L. Northrop, Software Product Lines: Practices and Patterns, Addison-Wesley Longman Publishing Co., Inc, Boston, MA, 2002.
[5] K. Pohl, G. Böckle, F. van der Linden, Software Product Line Engineering: Foundations, Principles, and Techniques, Springer, 2005.
[6] L. Chen, M. Ali Babar, A systematic review of evaluation of variability management approaches in software product lines, Inf. Softw. Technol. 53 (2011) 344–362, doi:10.1016/j.infsof.2010.12.006.
[7] M. Galster, D. Weyns, D. Tofan, B. Michalik, P. Avgeriou, Variability in software systems - A Systematic literature review, IEEE Trans. Softw. Eng. 40 (2013) 282–306, doi:10.1109/TSE.2013.56.
[8] L. Chen, M. Ali Babar, N. Ali, Variability management in software product lines: a systematic review, in: Proc. Int. Softw. Prod. Line Conf., Carnegie Mellon University, 2009, pp. 81–90.
[9] A.K. Thurimella, B. Bruegge, Issue-based variability management, Inf. Softw. Technol 54 (2012) 933–950, doi:10.1016/j.infsof.2012.02.005.
[10] F. Anon, V. Navarathinarasah, M. Hoang, C.-H. Lung, Building a framework for internet of things and cloud computing, in: Proc. IEEE Int. Conf. Internet Things (IThings), IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput., 2014, pp. 132–139, doi:10.1109/iThings.2014.28.
[11] A. Whitmore, A. Agarwal, L. Da Xu, The internet of things - A Survey of topics and trends, Inf. Syst. Front. 17 (2015) 261–274, doi:10.1007/s10796-014-9489-2.
[12] I. Lee, K. Lee, The internet of things (IoT): applications, investments, and challenges for enterprises, Bus. Horiz. 58 (2015) 431–440, doi:10.1016/j.bushor.2015.03.008.
[13] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of things (IoT): a vision, architectural elements, and future directions, Futur. Gener. Comput. Syst. 29 (2013) 1645–1660, doi:10.1016/j.future.2013.01.010.
[14] R. Buyya, A.V. Dastjerdi, Internet of things: principles and paradigms, 2016. doi:10.1016/C2015-0-04135-1.
[15] J. Bosch, R. Capilla, R. Hilliard, Trends in systems and software variability, IEEE Softw. 32 (2015) 44–61, doi:10.1109/MS.2015.74.
[16] R. Capilla, J. Bosch, K.-.C. Kang, Systems and Software Variability Management: Concepts, Tools and Experiences, Springer, Heidelberg, 2013.
[17] K.C. Kang, S.G. Cohen, J.A. Hess, W.E. Novak, A.S. Peterson, Feature-Oriented domain analysis (FODA) Feasibility Study: CMU/SEI-90TR-21, 1990.
[18] Ó. Ortiz, A.B. García, R. Capilla, J. Bosch, M. Hinchey, Runtime variability for dynamic reconfiguration in wireless sensor network product lines, in: Proc. Int. Softw. Prod. Line Conf, 2012, pp. 143–150, doi:10.1145/2364412.2364436.
[19] G. Yang, Y. Zhang, A feature-oriented modeling approach for embedded product line engineering, in: Proc. Int. Conf. Fuzzy Syst. Knowl. Discov, IEEE, 2015, pp. 1607–1612, doi:10.1109/FSKD.2015.7382185.
[20] C. Marimuthu, K. Chandrasekaran, Feature-Oriented domain analysis framework for energy-aware self-adaptive software, in: Proc. IEEE Int. Conf. Internet Things IEEE Green Comput. Commun, IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data, 2016, pp. 773–776, doi:10.1109/iThings-GreenCom-CPSCom-SmartData.2016.163.
[21] D. Miorandi, S. Sicari, F. De Pellegrini, I. Chlamtac, Internet of things: vision, applications and research challenges, Ad. Hoc. Netw. 10 (2012) 1497–1516, doi:10.1016/j.adhoc.2012.02.016.
[22] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, in: Proc. Int. Conf. Softw. Eng, ACM Press, 2007, p. 1051, doi:10.1145/1134285.1134500.
[23] B. Kitchenham, D. Budgen, P. Brereton, Evidence-Based Software Engineering and Systematic Reviews, CRC Press, 2015.
[24] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: an update, Inf. Softw. Technol 64 (2015) 1–18, doi:10.1016/j.infsof.2015.03.007.
[25] R.T. Geraldi, E. OliveiraJr, Defect types and software inspection techniques: a systematic mapping study, J. Comput. Sci. 13 (2017) 470–495, doi:10.3844/jcssp.2017.470.495.
[26] P.A. Mota Silveira Neto, I. Carmo Machado, J.D. McGregor, E.S. Almeida, S.R. Lemos Meira, A systematic mapping study of software product lines testing, Inf. Softw. Technol 53 (2011) 407–423, doi:10.1016/j.infsof.2010.12.003.
[27] S. Barney, K. Petersen, M. Svahnberg, A. Aurum, H. Barney, Software quality trade-offs: a systematic map, Inf. Softw. Technol. 54 (2012) 651–662, doi:10.1016/j.infsof.2012.01.008.
[28] B. Mohabbati, M. Asadi, D. Gašević, M. Hatala, H. a. Müller, Combining service-orientation and software product line engineering: a systematic mapping study, Inf. Softw. Technol. 55 (2013) 1845–1859, doi:10.1016/j.infsof.2013.05.006.
[29] B. Kitchenham, P. Brereton, A systematic review of systematic review process research in software engineering, Inf. Softw. Technol. 55 (2013) 2049–2075, doi:10.1016/j.infsof.2013.07.010.
[30] R. Wieringa, N. Maiden, N. Mead, C. Rolland, Requirements engineering paper classification and evaluation criteria: a proposal and a discussion, Requir. Eng. 11 (2005) 102–107, doi:10.1007/s00766-005-0021-6.
[31] C. Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: Proc. Int. Conf. Eval. Assess. Softw. Eng., ACM Press, 2014, pp. 1–10, doi:10.1145/2601248.2601268.
[32] C. Krupitzer, F.M. Roth, S. VanSyckel, G. Schiele, C. Becker, A survey on engineering approaches for self-adaptive systems, Pervasive Mob. Comput. 17 (2015) 184–206, doi:10.1016/j.pmcj.2014.09.009.
[33] A. Sánchez Guinea, G. Nain, Y. Le Traon, A systematic review on the engineering of software for ubiquitous systems, J. Syst. Softw. 118 (2016) 251–276, doi:10.1016/j.jss.2016.05.024.
[34] J. DeFranco, M. Kassab, P. Laplante, N. Laplante, The nonfunctional requirement focus in medical device software: a systematic mapping study and taxonomy, Innov. Syst. Softw. Eng. 13 (2017) 81–100, doi:10.1007/s11334-017-0301-6.
[35] J. Lee, K.C. Kang, A feature-oriented approach to developing dynamically reconfigurable products in product line engineering, in: Proc. Int. Softw. Prod. Line Conf., IEEE, 2006, pp. 131–140, doi:10.1109/SPLINE. 2006.1691585.
[36] L. Baresi, S. Guinea, L. Pasquale, Service-Oriented dynamic software product lines, IEEE Comput. 45 (2012) 42–48, doi:10.1109/MC.2012.289.
[37] L. Pessoa, P. Fernandes, T. Castro, V. Alves, G.N. Rodrigues, H. Carvalho, Building Reliable, Maintainable Dynamic Software Product Lines, An investigation in the body sensor network domain, Inf. Softw. Technol. 86 (2017) 54–70, doi:10.1016/j.infsof.2017.02.002.
[38] A. Venčkauskas, V. Štuikys, J. Toldinas, N. Jusas, A model-driven framework to develop personalized health monitoring, Symmetry (Basel) 8 (2016) 2–18, doi:10.3390/sym8070065.
[39] S.A. Sharif, M.S. Suhaimi, N.N. Jamal, I.K. Riadz, I.F. Amran, D.N.A. Jawawi, Real-Time campus university bus tracking mobile application, in: Proc. Seventh ICT Int. Student Proj. Conf., IEEE, 2018, pp. 1–6, doi:10.1109/ICT-ISPC.2018.8523915.
[40] I. Ayala, M. Amor, L. Fuentes, Towards a cvl process to develop agents for the iot, in: Proc. Int. Conf. Ubiquitous Comput. Ambient Intell., 2014, pp. 304–311, doi:10.1007/978-3-319-13102-3_51.
[41] A. Murguzur, X. De Carlos, S. Trujillo, On the support of multi-perspective process models variability for smart environments, in: Proc. Int. Conf. Model. Eng. Softw. Dev., 2014, pp. 1–6.
[42] M. Tekkalmaz, E. Tapci, Ö. Aydemir, An engineering tool suite for enhanced smart environments: GAMMA, in: Ceur Workshop Proc., 2018, pp. 1–13.
[43] M. Tomlein, K. Grønbaek, Semantic model of variability and capabilities of iot applications for embedded software ecosystems, in: Proc. Work. IEEE/IFIP Conf. Softw. Archit., 2016, pp. 247–252, doi:10.1109/WICSA.2016.17.
[44] A. Abbas, I.F. Siddiqui, S.U.J. Lee, Contextual variability management of iot application with XML-Based feature modelling, J. Theor. Appl. Inf. Technol. 95 (2017) 1300–1308.
[45] K. Mens, R. Capilla, H. Hartmann, T. Kropf, Modeling and managing context-aware systems' variability, IEEE Softw. 34 (2017) 58–63, doi:10.1109/MS.2017.4121225.

[46] A.M. Amja, A. Obaid, H. Mili, Combining variability, RCA and feature model for context-awareness, in: Proc. Int. Conf. Innov. Comput. Technol., 2017, pp. 15–23, doi:10.1109/INTECH.2016.7845015.

[47] A.M. Amja, A. Obaid, H. Mili, P. Valtchev, Linking relational concept analysis and variability model within context modeling of context-aware applications, in: Proc. IEEE Int. Symp. Syst. Eng., 2016, pp. 1–8, doi:10.1109/SysEng.2016.7753184.

[48] E.T. McGee, J.D. McGregor, in: Proc. Int. Work. Softw. Eng. Adapt. Self-Managing Syst., ACM Press, 2016, pp. 115–121, doi:10.1145/2897053.2897062.

[49] J. Mauro, M. Nieke, C. Seidl, I.C. Yu, Context aware reconfiguration in software product lines, in: Proc. Int. Work. Var. Model. Software-Intensive Syst., 2016, pp. 41–48, doi:10.1145/2866614.2866620.

[50] T.B. Røst, C. Seidl, I.C. Yu, F. Damiani, E.B. Johnsen, C. Chesta, HyVar: scalable hybrid variability for distributed evolving software systems, in: Proc. Eur. Conf. Serv. Cloud Comput., 2018, pp. 159–163, doi:10.1007/978-3-319-79090-9_12.

[51] T. Heikkilä, T. Dobrowiecki, L. Dalgaard, Dealing with configurability in robot systems, in: Proc. IEEE/ASME Int. Conf. Mechatron. Embed. Syst. Appl., 2016, pp. 1–7, doi:10.1109/MESA.2016.7587120.

[52] A. Achtaich, N. Souissi, R. Mazo, C. Salinesi, O. Roudies, Designing a framework for smart IOT adaptations, in: Proc. Int. Conf. Emerg. Technol. Dev. Ctries., 2018, pp. 57–66, doi:10.1007/978-3-319-67837-5_6.

[53] A. Venčkauskas, V. Štuikys, N. Jusas, R. Burbaitė, Model-Driven approach for body area network application development, Sensors 16 (2016) 670, doi:10.3390/s16050670.

[54] Ö. Köksal, B. Tekinerdogan, Architecture design approach for iot-based farm management information systems, Precis. Agric. (2018) 1–33, doi:10.1007/s11119-018-09624-8.

[55] A. Abbas, I.F. Siddiqui, S.U.-J. Lee, A.K. Bashir, Binary pattern for nested cardinality constraints for software product line of iot-based feature models, IEEE Access 5 (2017) 3971–3980, doi:10.1109/ACCESS.2017.2680470.

[56] C. Cetina, P. Giner, J. Fons, V. Pelechano, Autonomic computing through reuse of variability models at runtime: the case of smart homes, IEEE Comput. 42 (2009) 37–43, doi:10.1109/MC.2009.309.

[57] A.D. Sinnhofer, A. Höller, P. Pühringer, K. Potzmader, C. Orthacker, C. Steger, C. Kreiner, Combined variability management of business processes and software architectures, in: Proc. Int. Symp. Bus. Model. Softw. Des., 2017, pp. 36–45, doi:10.5220/0006527300000000.

[58] A.D. Sinnhofer, P. Pühringer, F.J. Oppermann, K. Potzmader, C. Orthacker, C. Steger, C. Kreiner, Combining business process variability and software variability using traceable links, in: Proc. Int. Symp. Bus. Model. Softw. Des., 2018, pp. 67–86, doi:10.1007/978-3-319-78428-1_4.

[59] K. Suri, W. Gaaloul, A. Cuccuru, Proc. Int. Conf. Serv. Comput., 119–136 (2018), doi:10.1007/978-3-319-94376-3_8.

[60] F. Kneer, E. Kamsties, A case study on self-configuring systems in iot based on a model-driven prototyping approach, in: Proc. Int. Conf. Commun. Comput. Inf. Sci., 2016, pp. 732–741, doi:10.1007/978-3-319-46254-7_59.

[61] C. Cecchinel, S. Mosser, P. Collet, Automated deployment of data collection policies over heterogeneous shared sensing infrastructures, in: Proc. Asia-Pacific Softw. Eng. Conf., 2016, pp. 329–336, doi:10.1109/APSEC.2016.053.

[62] B. Jazayeri, O. Zimmermann, G. Engels, D. Kundisch, A variability model for store-oriented software ecosystems: an enterprise perspective, in: Proc. Int. Conf. Serv. Comput., 2017, pp. 573–588, doi:10.1007/978-3-319-69035-3_42.

[63] S. Di Cola, K. Lau, C. Tran, C. Qian, Towards defining families of systems in IoT logical architectures with variation points, in: IoT 360o, 2016, pp. 419–426, doi:10.1007/978-3-319-47063-4_43.

[64] E. Santos, I. Machado, Towards an architecture model for dynamic software product lines engineering, in: Proc. IEEE Int. Conf. Inf. Reuse Integr., 2018, pp. 31–38, doi:10.1109/IRI.2018.00013.

[65] B.I. Moritani, J. Lee, An approach for managing a distributed feature model to evolve self-adaptive dynamic software product lines, in: Proc. Int. Syst. Softw. Prod. Line Conf., B, 2017, pp. 107–110, doi:10.1145/3109729.3109743.

[66] E.T. McGee, M. Krugh, J.D. McGregor, L. Mears, Designing for reuse in an industrial internet of things monitoring application, in: Proc. Work. Soc. Human, Econ. Asp. Softw., 2017, pp. 19–25, doi:10.1145/3098322.3098323.

[67] S. Hosoai, N. Noda, T. Kishi, System product line engineering for small appliances with driver derivation, in: Proc. Asia-Pacific Softw. Eng. Conf., 2017, pp. 389–392, doi:10.1109/APSEC.2016.067.

[68] I. Logre, S. Mosser, P. Collet, M. Riveill, Sensor data visualisation: a composition-based approach to support domain variability, in: Proc. Eur. Conf. Model. Found. Appl., 2014, pp. 101–116, doi:10.1007/978-3-319-09195-2_7.

[69] S. Nešković, R. Matić, Context modeling based on feature models expressed as views on ontologies via mappings, Comput. Sci. Inf. Syst. 12 (2015) 961–977, doi:10.2298/CSIS141031035N.

[70] A. Venčkauskas, V. Štuikys, R. Damasevicius, N. Jusas, Modelling of internet of things units for estimating security-energy-performance relationships for quality of service and environment awareness, Secur. Commun. Netw. 9 (2016) 3324–3339, doi:10.1002/sec.1537.

[71] I.M. Murwantara, H. Tjahyadi, P. Yugopuspito, A. Aribowo, I.A. Lazarusli, Towards adaptive sensor-cloud for internet of things, TELKOMNIKA (Telecommunication Comput. Electron. Control.) 16 (2018) 2771–2781, doi:10.12928/TELKOMNIKA.v16i6.11557.

[72] P. Smiari, S. Bibi, A smart city application modeling framework: a case study on *re*-engineering a smart retail platform, in: Proc. Euromicro Conf. Softw. Eng. Adv. Appl., IEEE, 2018, pp. 111–118, doi:10.1109/SEAA.2018.00027.

[73] A. Abbas, I. Farah Siddiqui, S.U.-J. Lee, A. Kashif Bashir, W. Ejaz, N.M.F. Qureshi, Multi-Objective optimum solutions for iot-based feature models of software product line, IEEE Access 6 (2018) 12228–12239, doi:10.1109/ACCESS.2018.2806944.

[74] M. Weckesser, R. Kluge, M. Pfannemüller, M. Matthé, A. Schürr, C. Becker, Optimal reconfiguration of dynamic software product lines based on performance-influence models, in: Proc. Int. Conf. Syst. Softw. Prod. Line, ACM Press, 2018, pp. 98–109, doi:10.1145/3233027.3233030.

[75] I. Ayala, M. Amor, L. Fuentes, J.M. Troya, A software product line process to develop agents for the IoT, Sensors 15 (2015) 15640–15660, doi:10.3390/s150715640.

[76] I. Ayala, J.M. Horcas, M. Amor, L. Fuentes, Using models at runtime to adapt self-managed agents for the IoT, in: Proc. Ger. Conf. Multiagent Syst. Technol, 2016, pp. 155–173, doi:10.1007/978-3-319-45889-2_12.

[77] I. Ayala, M. Amor, L. Fuentes, Using SPL to develop aal systems based on self-adaptive agents, Trends Pract. Appl. Scalable Multi-Agent Syst. (2016) 263–275, doi:10.1007/978-3-319-40159-1_22.

[78] N. Gamez, L. Fuentes, Architectural evolution of famiware using cardinality-based feature models, Inf. Softw. Technol. 55 (2013) 563–580, doi:10.1016/j.infsof.2012.06.012.

[79] M. Tomlein, K. Grønbaek, Building models of installations to recommend applications in iot software ecosystems, in: Proc. IEEE Int. Conf. Futur. Internet Things Cloud, 2016, pp. 9–16, doi:10.1109/FiCloud.2016.10.

[80] D.-.J. Munoz, Achieving energy efficiency using a software product line approach, in: Proc. Int. Syst. Softw. Prod. Line Conf., B, ACM Press, 2017, pp. 131–138, doi:10.1145/3109729.3109744.

[81] D.-.J. Munoz, M. Pinto, L. Fuentes, Green software development and research with the hadas toolkit, in: Proc. Eur. Conf. Softw. Archit. Companion, 2017, pp. 205–211, doi:10.1145/3129790.3129818.

[82] D.-J. Munoz, M. Pinto, L. Fuentes, HADAS and web services: eco-efficiency assistant and repository use case evaluation, in: Proc. Int. Conf. Energy Sustain. Small Dev. Econ., 2017, pp. 1–6, doi:10.1109/ES2DE.2017.8015334.

[83] F. Damiani, M. Lienhardt, R. Muschevici, I. Schaefer, An extension of the abs toolchain with a mechanism for type checking SPLs, in: Proc. Int. Conf. Integr. Form. Methods, Springer, 2017, pp. 111–126, doi:10.1007/978-3-319-66845-1_8.

[84] E.T. McGee, J.D. McGregor, Using dynamic adaptive systems in safety-critical domains, in: Proc. Int. Work. Softw. Eng. Adapt. Self-Managing Syst., 2016, pp. 115–121, doi:10.1145/2897053.2897062.