Information and Software Technology 55 (2013) 1845–1859

Contents lists available at SciVerse ScienceDirect

Information and Software Technology

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

Combining service-orientation and software product line engineering: A systematic mapping study

Bardia Mohabbati [a,*], Mohsen Asadi [a], Dragan Gašević [a,b], Marek Hatala [a], Hausi A. Müller [c]

[a] Simon Fraser University, 13450 102 Ave. Surrey, BC V3T 5X3, Canada
[b] Athabasca University, 1 University Drive Athabasca, AB T9S 3A3, Canada
[c] University of Victoria, STN CSC Victoria, BC V8W 3P6, P.O. Box 3055, Canada

A R T I C L E I N F O

Article history:
Received 17 December 2012
Received in revised form 24 May 2013
Accepted 24 May 2013
Available online 13 June 2013

Keywords:
Systematic mapping
Service-oriented architecture
Software product lines

A B S T R A C T

Context: Service-Orientation (SO) is a rapidly emerging paradigm for the design and development of adaptive and dynamic software systems. Software Product Line Engineering (SPLE) has also gained attention as a promising and successful software reuse development paradigm over the last decade and proven to provide effective solutions to deal with managing the growing complexity of software systems.
Objective: This study aims at characterizing and identifying the existing research on employing and leveraging SO and SPLE.
Method: We conducted a systematic mapping study to identify and analyze related literature. We identified 81 primary studies, dated from 2000–2011 and classified them with respect to research focus, types of research and contribution.
Result: The mapping synthesizes the available evidence about combining the synergy points and integration of SO and SPLE. The analysis shows that the majority of studies focus on service variability modeling and adaptive systems by employing SPLE principles and approaches.
Result: In particular, SPLE approaches, especially feature-oriented approaches for variability modeling, have been applied to the design and development of service-oriented systems. While SO is employed in software product line contexts for the realization of product lines to reconcile the flexibility, scalability and dynamism in product derivations thereby creating dynamic software product lines.
Conclusion: Our study summarizes and characterizes the SO and SPLE topics researchers have investigated over the past decade and identifies promising research directions as due to the synergy generated by integrating methods and techniques from these two areas.

Contents

* Corresponding author. Tel.: +1 778 782 7431.
E-mail addresses: mohabbati@sfu.ca (B. Mohabbati), masadi@sfu.ca (M. Asadi), dgasevic@sfu.ca (D. Gašević), mhatala@sfu.ca (M. Hatala), hausi@cs.uvic.ca (H.A. Müller).

0950-5849/$ - see front matter
http://dx.doi.org/10.1016/j.infsof.2013.05.006

1. Introduction

Service-Orientation (SO), which is manifested in an architectural style that is commonly referred to as Service-Oriented Architecture (SOA), has substantially contributed to changing the perspective of todays software development. Service-oriented computing is gaining momentum as a means to develop scalable and flexible distributed applications. This momentum aims at addressing various challenges such as application integration, reusability, modularity and interoperability [38].

SO focuses on creating and developing application solutions by utilizing services as building blocks of software that encapsulate functionality and provide flexibility through dynamic binding. The visionary promise of SO is that applications are composed seamlessly by loosely-coupled services in order to create adaptive and dynamic systems. Services are characterized as a set of autonomous, platform-independent computational units which can be described, published, discovered, and dynamically composed and assembled [38]. SOA provides the architectural underpinnings to support software reuse and enables variability at both design- and run-time. Variability refers to the ability of a software system to be configured, customized and adopted to meet particular requirements [45]. However, SOA lacks support for modeling and managing variability that promotes configurability (i.e., customization) and systematically-managed reuse [3,48]. SOA artifacts, including service specifications and process models, are not generally designed with variability for planned and enforced reuse requirements in mind.

Variability is an important concept, and variability analysis, modeling and management have been core research subjects in software product line research. Software Product Line Engineering (SPLE) addresses the issues of engineering and developing software-intensive systems and supports large-scale reuse in the course of development. SPLE offer effective methods and techniques for variability modeling and systematic reuse in software development in order to (i) support configurable software architectures and (ii) enable for mass customization of software-intensive systems [41].

Combining and integrating SO and SPLE have been a subject of considerable research interest in recent years, observed through the literature (e.g., [2,9,11,15,20,22,33]), and dedicated workshop series [12,31,32]. Hence, we need to synthesize the evidence regarding the usefulness of combination. Therefore, we conducted a systematic mapping study, based on our preliminary results of our previous work [34], to analyze the existing research and relevant literature published on this topic. Systematic mapping studies (a.k.a. scoping studies) are designed and performed to provide a wide overview of a research area [28]. In software engineering, systematic mapping studies have been recommended when a research topic is new or not mature enough to comprise a set of comparable empirical studies [28,27]. The aim is to "map out" the research undertaken instead of answering detailed research questions in contrast to systematic literature reviews (SLRs), which derive very specific research questions [6,28,40]. Thus, a mapping study as a part of evidence-based software engineering is conducted if research evidence exists on a topic and provides an indication of the quantity of the evidence [28].

This paper presents a systematic mapping study that aims at collecting evidence about how SO and SPLE are combined or integrated with the aim to identify the research trends and categorize studies at a higher granularity level. Our goal is to provide a map of existing research and synthesize current evidence on the integration of two paradigms. Moreover, the outcomes of our mapping study can help to identify research challenges and to direct ongoing research.

The rest of paper is constructed as follows: Sections 2 and 3 describe the methodology followed in our conducted mapping study and classification scheme. Section 4 reports results. The threats to validity are given in Section 5. Section 6 presents a discussion. Section 7 conclude this work and highlights the direction of future work.

2. Systematic mapping and research method

Efforts in software engineering research are dedicated toward developing a standard methodology for conducting mapping studies [7,28,40]. Peterson et al. [40] describe methods for conducting mapping studies and discuss differences between systematic mapping studies and systematic literature reviews. Moreover, they provide guidelines for a broader set of situations where either or both systematic mapping studies and systematic literature reviews are appropriate and required to be conducted. A systematic literature review is conducted as a means of identifying, evaluating, interpreting, and comparing all available researches that are relevant to a particular research question and relative merits of competing technologies. In contrast, a systematic mapping study provides a systematic and objective procedure to determine the nature and extent of the empirical-study data to answer a particular research question [7].

Mapping studies often use the same basic methodology as systematic literature reviews; however, they aim at identifying and classifying all related research into a broad software engineering topic.

The software community has been working to towards well-defined methods to conduct mapping studies [40]. The procedure of systematic mapping study presented in this paper combines a well-organized set of proper practices both to undertake mapping study and to systematically review guidelines in the context of software engineering [5,26,40]. The combination of guidelines help us to leverage both systematic mapping and literature review techniques. The major steps of our systematic mapping process comprised: (1) definition of a protocol of the study and research questions; (2) exploratory search and data collection; and (3) analysis of the collected data and reporting. Fig. 1 summarizes a process followed in our systematic mapping study. The details of each step are described in the following subsections.

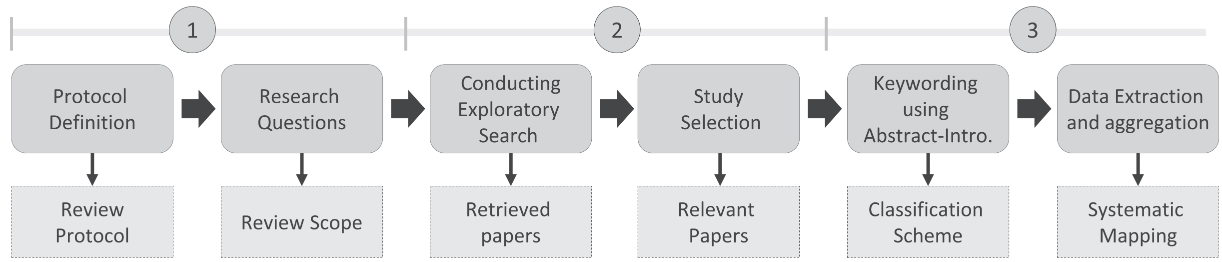

Fig. 1. Systematic mapping process.

2.1. Review protocol

A systematic mapping study starts by defining a review protocol specifying the plan of research and primarily including the research. The review protocol primarily includes the research questions, search process, study selection criteria, the classification schema, and the method to extract and analyze data. The review protocol is also important for other researchers who would like to either extend or replicate the study [28]. It is sufficient, in the case of the former, to include only new studies through the subsequent literature search. Our literature review protocol is adopted from the systematic literature review guidelines defined in [26]. We adopted the concepts such as protocol definition that derives the research in the study during the study definition, which improves reliability by describing different aspects of the review to conduct an unbiased mapping study [40].

The final protocol used in our study included the following steps: (1) defining the research questions, the scope and search strategy, the inclusion and exclusion criteria to select studies; (2) performing search, collecting and selecting studies using established protocol, and (3) classifying and analyzing papers and extracting and aggregating data to produce a comprehensive overview of the published literature on the studied topic. The protocol is reviewed incrementally and updated according to the newly collected data in the course of study. The protocol was also evaluated and incrementally revised by individual authors. The following subsections describe the content of the protocol in detail.

2.2. Research questions

The study aim is to provide an overview of approaches combining or integrating SO and SPLE, and of their goals. Our main objective is to characterize and summarize evidence and identify the existing research. The research questions for present mapping study is formulated with the help of Population, Intervention, Comparison, Outcome, Context (PICOC) criteria specified in [21]. Because the objective of this mapping study is not to find evidence on the comparison of approaches, methods or models; the "Comparison" attribute of PICOC was not utilized. The populations is included the studies that adopted both SO and SPLE. The intervention includes concepts, principles, approaches or techniques employed from one paradigm to another. The outcome of interest represents a mapping of studies, classification, type and quality of evidence relating to the adaptation of SO and SPLE. The context is within the domains of SOA and SPL with a focus on empirical studies and their application domain. The primary research question that guided our systematic mapping study is: *What is the available evidence regarding the integration or adoption of methods and principles of both SPLE and SO paradigms?*

To answer this question a set of sub-questions was derived from the primary research question to identify and analyzing the relevant literature discussions.

RQ1: *In which fora is research on integrating or combining SO and SPLE principles and practices published?* There are a few workshops specifically devoted to service-oriented and software product lines; however, our experience from earlier reviews shows that research may be published in different fora. What are the types of papers typically published (e.g., conference proceedings, journals, and workshops), and the actual publication venues? Publication chronology of the papers and distribution of studies in different venues substantially indicates the existence of research activity.

RQ2: *What is the focus and objective (motivation) of the existing research results on integrating or combining SO and SPLE principles and practices? Based on identified synergies, what are the identified characteristics of the possible exploitation?* To identify the focus and key issues which have motivated adoption of one paradigm into the other is the aim of this research question. We have also focused on giving an overview of approaches which are proposed or applied by the integration or combination of SO and SPLE principles.

RQ3: *What are the domains and contexts applied in these proposals?* The combination or integration of both paradigms can be applied in different contexts with different application domains such as management information. Therefore, this question intends to determine the application domains that the existing research focuses on.

RQ4: *What types of research and contribution are represented?* This question identifies the types of research (e.g., evaluation research, solution proposal, etc.) and contribution (contribution facet), such as method, model, language, etc.

The answer to RQ1 was retrieved by collecting and selecting related studies and providing distribution of publication fora which is given Section 4.1. A preliminary classification scheme was established and developed through keywording [28,40] with RQ2 and RQ3 answered through analyzing the 81 selected studies described in Sections 4.2 and 4.3, respectively and the answer to RQ4 retrieved by analyzing the types of research and contribution, summarized in Section 4.4.

2.3. Data sources

To collect primary studies, our exploratory search process included digital libraries as well as manual search of journals, conference and workshop proceedings of the most relevant organizations in the software engineering community. The pre-review search was performed using the IEEE Computer Society Digital Library, ACM Digital Library, Science Direct, Citeseer, SpringerLink, InderScience, and Google Scholar. These search engines cover the vast majority of published studies in software engineering. Moreover, the manually conducted search took into account topic-specific conference, workshop proceedings and technical reports regarding the research topics such as Software Product Line Conference (SPLC), Software Reuse, and Software Engineering Institutes (SEI) workshop series and technical reports on SOA and SPLE. These resources are considered the key publication resources in SPLE research area.

Table 1
List of related search terms.

Search Query (SQ)
SQ1: "Service-Oriented Product Line" OR "'Service-Oriented Software Product Line" OR [*SQ2* AND *SQ3*]
SQ2: ("software product line" OR SPL) OR ("product line" OR "product family") OR "process family" OR "product line engineering" OR "domain engineering" OR "application engineering" OR "variability" OR "variability modeling" OR "variability management" OR "variability analysis" OR "feature modeling" OR "feature analysis"
SQ3: ("service-based OR service-oriented" OR "service orientation") OR ("service-oriented architecture" OR 'SOA") OR "service computing" OR "service engineering" OR "service development" OR "web services" OR ("software reuse services" OR "service reuse") OR "service variability" OR "service customization" OR "service identification" OR ("service orchestration" OR "service composition")

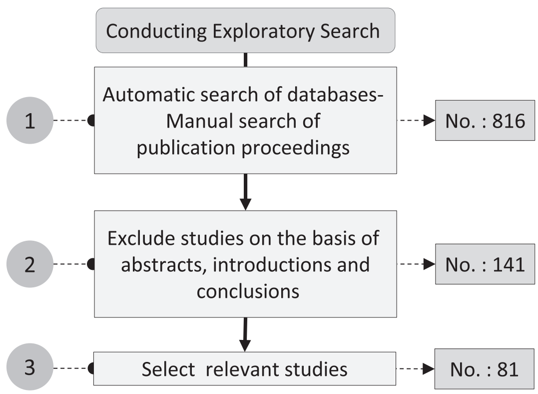

Fig. 2. Stages of search and study selection.

2.4. Search strategy

The search terms in our mapping study were constructed using the guidelines described in [13], i.e., through the following steps: (1) deriving major terms by breaking down the research questions; (2) identifying alternative spelling and synonyms for major terms; (3) checking the keywords in any relevant papers we already retrieved; (4) incorporating alternative spellings and synonyms using the logical operator 'OR'; and (5) using the logical operator 'AND' to link the major terms.

The main search terms are shown in Table 1. The devised search queries were launched for individual selected sources to identify and collect publications. The numerousness of features of digital libraries necessitated slight modification and calibration for respective query interfaces. Furthermore, we accumulated all the keywords given by the authors of the selected papers with the outcome being a tag cloud generated of terms including more than one frequency (see Fig. A.1). We considered four major keywords deriving our initial search: "service orientation/SO", "service-oriented architecture/SOA","software product line/SPL" and "software product line engineering/SPLE".

These terms– obviously the most frequent keywords according to the topic understudy, are excluded from generated tag cloud to reflect better most frequent alternative terms. It may guide and enhance search queries for further studies. It can be observed that the majority of captured keywords were also considered in our search queries.

In this paper, we report the results of a broad search covering the period from 2001 to 25th September 2011. Therefore, we are nearly certain that we have grasped every paper published in the period of the span of our study despite publication time lag. The details and full texts were recorded and managed by using the open source bibliography management tool Zotero,[1] primarily due to its integrated capability to manage, share, and synchronize resources among the researchers.

[1] http://www.zotero.org.

Fig. 2 shows the number of papers identified at each stage of exploratory search and study selection. In stage one, the titles, abstracts, and keywords of publications in the included digital libraries and databases were searched with the combination of given search terms. The outcome of manual search within selected publication fora is also included in this stage. This state resulted in 816 publications.

In stage two, the duplicated studies were removed, and it comprised initial inclusion and exclusion based on the title, abstract, introduction and conclusion sections. The initial inclusion criteria were studies with both paradigms. We collected 132 studies for further consideration. Furthermore, the study collection is followed by performing "snow-balling", such as pursuing related works and papers listed in the references of papers. We also communicated directly with some authors for secondary research to pinpoint any other studies that the database searches and manual search might have missed, all leading to nine additional publications. In total, we finally identified and collected 141 relevant publications and cleaned out the duplicated studies of the same author published in different venues. Typically the most recent ones were considered and included. In the final stage, 81 studies are included for this mapping study after applying selection criteria that are described in the next subsection.

2.5. Inclusion and exclusion criteria

The study selection processes is performed based on careful reading of the collected papers in order to select the most relevant studies which address our research questions. We define following inclusion and exclusion criteria for the selection of collected publications:

- *Inclusion*: Publications with a focus on some aspects of combination, integration or applications of both SO and SPLE.
- *Exclusion*: publications with no concentration on or proposal of an approach related to combination, integration or applications of both SO and SPLE.

Quality assessments of studies are important to restrict bias in conducting the review and to guide interpretation of findings [19]. However, evaluating the quality of a study is not straightforward due to the fact there is no unanimous or universal definition of quality. Because the goal of systematic mapping studies is to structure a large area within the research domain under review, systematic mapping study guidelines do not constitute a formal quality assessment criteria [40]. We used accepted criteria to assess the quality of the studies to test the suitability and selection of studies. Articles must have a minimal description of the context, clear objectives or research goals, and a consistent description of the proposed approach.

In this work, we further consider selected studies for more discussion about their proposed approaches by principles of good practice of conducting empirical research in software engineering proposed by Kitchenham et al. [29]. Owning to the fact that the majority of identified studies have been published in recent years, we did not subject them to stringent critical appraisal based on the number of citations.

2.6. Data extraction

The data extraction forms were designed to collect required information. We collected the following information from each study: title and authors; source and type (journal, conference, symposium, workshop, book chapter or technical report); date of publication; and collected information based on our classification scheme (cf. Section 3) and answers to individual research questions. The summary and description of research objectives and focus, research method and contributions were also included. The data are extracted and stored in spreadsheet. The workbook was shared among the authors for the collaboration and synchronization using the share workbook feature in MS Office. During the data extraction, each paper was read entirely and was also scrutinized using the inclusion and exclusion criteria.

3. Classification scheme

To address the research questions of this mapping study, a classification scheme is defined to analyze different facets of the analyzed studies in terms of *research focus*, *research type* and *type of contribution* as well as *proposed approaches* – for primary studies. We developed the classification schema by following keyword method [28,40]. Constituting the mapping publication and classification scheme were performed iteratively as a new study was added. The classification schema is created and developed by keywording the abstract, introduction and conclusion, and reading the full-text of studies in order to identify and cluster different facets and contributions of the papers. The set of keywords from different papers were incorporated to develop a high-level realization about the nature and contribution of the papers. This helped us to come up with a set of categories which are representative of the different facets of the research and underlying population.

To address the research question RQ2, we identified the following primary classes of research focus and their application domains (RQ3) (cf. Section 4.3): *Service Variability Modeling*, *Service Reuse*, *Service Identification*, *Service Configuration and Customization Management*, *Dynamic Software Product Line*, and *Adaptive Systems*.

The contribution type (RQ4) is classified into eight categiries: *Metrics*, *Tooling support*, *Method*, *Model*, *Language*, *Open items*, *Analytical discussion*, and *Comparison analysis*. Resarch type is classified into six categories based on adopted scheme proposed by Wieringa et al. [46]: *Evaluation research*, *Validation research*, *Solution proposal Conceptual proposal*, *Opinion-oriented papers*, and *Experience reports*.

During the data extraction, some studies could be classified into more than one category according to schema. Based on the classification schema we identified different classes of research focus and type of contribution discussed below.[2]

4. Results

This section describes the results obtained after conducting the systematic mapping study by following the method described in the previous section.

4.1. Demographic data

We provide an overview of the reviewed papers with respect to the types of publication, sources, and research trends. The answer to RQ1 was addressed by collecting information and distribution of publication fora. Fig. 3 depicts the distribution of the publication types and the grand total number of studies published according to the chronological order of publication years. Evidently, there is

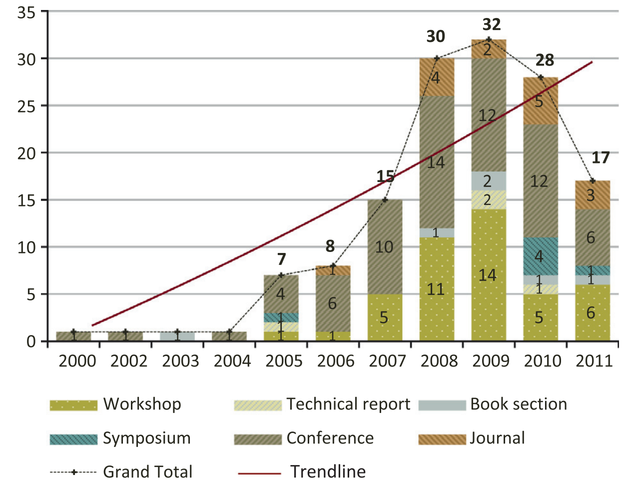

Fig. 3. Distribution of publication type by publication years.

an increased number of research studies published between 2005 and 2011, when an important international workshop was devoted to the intersect of SO and SPLE was organized (i.e., the SOAPL workshop series).

Research directions of the series of this workshop mainly focused on exploring how these two paradigms can benefit from each other [32,31,12]. The growing number of publications in recent years generally corroborates this trend, and many research works have recently become available (disregarding 25th September 2011 and 2012 when search processes ended since many studies may not have been indexed in digital libraries or made available by search engines). The trend testifies the interest of the research community in expanding this research area.

Fig. 4 shows the distribution of research studies published in workshops series, and Fig. 5 demonstrates the distribution of publication fora categorized by source including journal, conference, and symposium papers. The book sections and technical reports are succinctly named "others". All the sources are listed in Appendix B (cf. Tables B.1 and B.2).

The distribution of publications in terms of types and resources in this mapping study (RQ1) indicates the important data sources for this research area and further studies.

4.2. Research focus

Fig. 6 shows the distribution of research foci of identified and collected primary studies. If the focus or contribution was within more than one area, we assigned the paper to several research foci (cf. Table 2). The following subsections address research questions RQ2 and RQ3, where each of research foci is discussed, and an overview of studies is given with respect to their approaches reported.

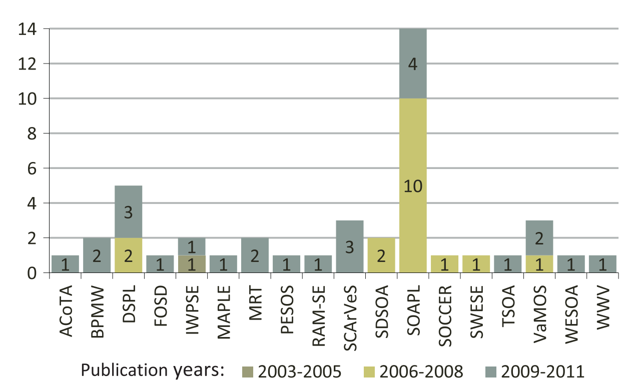

Fig. 4. Distribution of workshops series.

[2] The search process ended up on 25th September 2011.

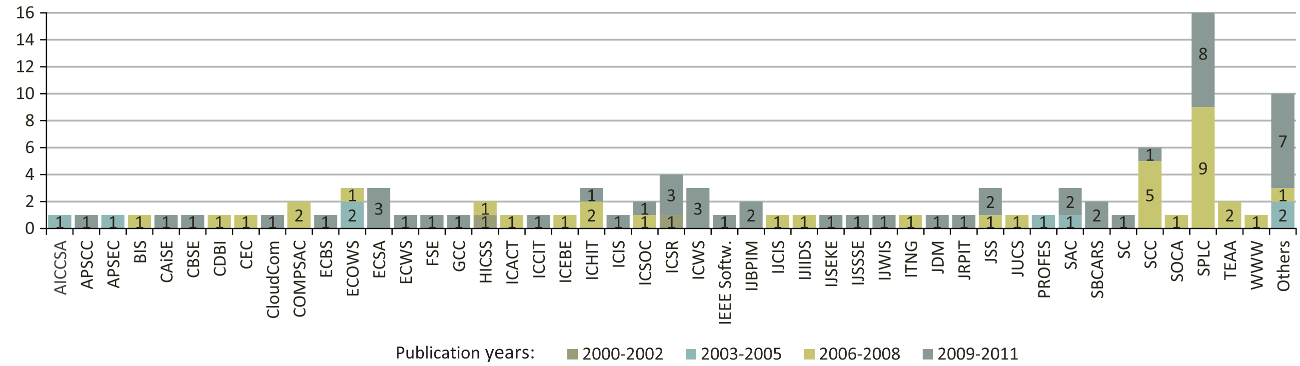

Fig. 5. Distribution of journal and conference papers.

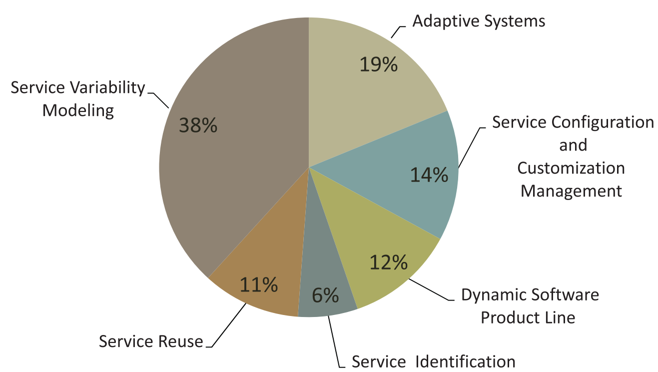

Fig. 6. Distribution of research foci of collected primary studies.

4.2.1. Service variability modeling

It is vital that variability modeling and management be supported to enable and promote reuse and customization of software-intensive systems [11]. Variability modeling, as a main principle of SPLE, is employed by many studies in the context of SO to develop highly variable services in a managed environment.

Variability modeling in SO differs from variability modeling in SPLE in several aspects. In SO, different levels of abstraction including business requirements, service compositions, service interface, and service implementation are all variables. Not only should we consider variability of intra-organization services and legacy systems, but also variability in third-party services should be taken into account, incorporated into methods and languages dedicated to SO (e.g., BPEL, WSDL), and considered in dynamic runtime [15,35].

In SPLE, feature modeling [24] is a widely used variability modeling technique to describe point of variability in an SPL. A feature model leverages the notion of *features* – visible characteristics of software systems to describe the variation points and variants. Features are defined as visible characteristics of software systems [24]. Feature models provide several types of variability relations (e.g., optional, alternative, and OR) as well as cross-tree dependency relations called integrity constraints. Several variability modeling techniques have been utilized to represent variability at various abstraction levels including requirements, composition of services (i.e., business process), service interfaces, and implementation of services. Notably, a large body of work has concentrated on modeling functional variability in the context of SOA while few approaches model variability in the quality of services.

A large number of the papers is incorporated in our study, employed feature-driven approaches for modeling functional and data variabilities in the context of SOA. Gruler et al. [S33,S38] and Zaupa et al. [S81] employed feature models [4] to represent functional variabilities in service operations. Similarly, Acher et al. [S2] modeled functional variabilities using feature models. For each service, they developed three feature models representing variability in service interfaces, service inputs, and service outputs. Schnieders and Puhlmann [S68] employed feature models to represent functional variability in the workflow systems. In addition, they extended Business Process Model and Notation (BPMN) by variability realization mechanisms such as inheritance, parameterization, extension, and design patterns to incorporate variability in the design artifacts. Diaz et al. [S25] extended the Web Services for Remote Portlet (WSRP) specification to support developing a family of Portlets. They also used a feature models to represent variability in consumer profiles (i.e., functional requirements of consumers) and mapped features to artifacts using a *feature x artifact* matrix. Nanjangud and Karthikeyan [S58] proposed a variation metamodel consisting of variation points and variant features for service modeling to support modeling functional variability that may exist in composite services and service interfaces.

Some of the collected studies employed feature models for modeling variability of both functional and non-functional (QoS) aspects of services. The results presented in Asadi et al. [S9], Boskovic et al. [S13,S14], and Mohabbati et al. [S57] represent variability

Table 2
Research foci and references.

Research foci	References
Service variability modeling	[S33],[S38],[S2],[S68],[S25],[S58],[S31],[S28],[S43],[S47],[S5], [S73],[S62],[S80],[S72],[S75],[S1],[S63],[S42],[S67],[S79],[S61],[S11],[S9],[S13],[S14],[S57],[S12],[S32],[S34],[S48],[S69],[S76]
Service identification	[S50],[S51],[S71],[S29],[S41],[S4],[S23]
Service reuse	[S16],[S49],[S15],[S81],[S25],[S10],[S59],[S40],[S8]
Service configuration and customization	[S55],[S54],[S74],[S2],[S62],[S35],[S42],[S57],[S77],[S66],[S60],[S72]
Dynamic software product line	[S49],[S64],[S65],[S36],[S20],[S18],[S19],[S39],[S70],[S80],[S46],[S48]
Adaptive systems	[S27],[S26],[S17],[S3],[S78],[S21],[8],[S45],[S52],[S6],[S56],[S44],[S53],[S8],[S37],[S24]

in the business requirements, service composition, and quality of services using feature models. They extended feature models with annotation properties to annotates features with required quality values. The DiVA approach [S31] proposed by Greenwood et al. captures variability in both functional and non-functional requirements of services and employs feature modeling to represent variability in the requirements and the composition of services. Moreover, an aspect-oriented model is utilized to encapsulate the implementation of service variants. Fantinato et al. [S28] used feature models to represent variability in Web services e-contract (WS-contract) definitions and quality of service. In their approach, mandatory, optional, and alternative elements of Web service contracts are identified and represented using feature models concepts. The feature model is mapped to the contract template, and configurations of the feature models are used to derive a concrete Web service contract. Kattepur et al. [S43] modeled variability and constraints of composite service orchestrations via feature models and proposed an effective technique for achieving feasible service configurations. La and Kim [S47] categorized different types of variability in Software-as-a-Service (SaaS) applications into attribute, logic, workflow, interface, persistency, context, and quality of service. They proposes a high level feature-oriented process to capture and model variability. As a result of variability modeling activity, decision tables containing a set of variable features along with their variability characteristics are generated. Medeiros et al. [S5] adopted feature-driven approach to illustrate both functional and non-functional variability in a set of services. Their work discusses variability possibly occurring in service interfaces, business processes, and service components.

Furthermore, other variability modeling languages, such as Orthogonal Variability Model (OVM) [41] and COVAMOF [43], have also been considered for modeling variability in the context of service-oriented software development. Sun et al. [S73] employed the COVAMOF variability modeling framework as a separate variability dimension used in combination with UML diagrams to model variability in the context of service-based systems. They concentrated on variability in service compositions and service interfaces and addressed both functional and non-functional aspects. In [S62], Nunes et al. focused on modeling functional variability at the requirement level (represented by using goal models) and process models (represented by using business process model notation) by OVM.

UML was used to model and represent variability in either a separate dimension or in internal models. Yu et al. [S80] extended the UML to represent variability in service compositions. The approach focuses on representing functional variability (i.e., difference in services) that provides various functionalities in a variation point. Stollberg and Muth [S72] extended SOAML (a UML profile for SOA modeling) and defined a service variability meta-model to describe necessary constructs to model variability in services. Moreover, they proposed a service variability specification models to define variable aspects of service interfaces including data and operations. An approach proposed by Wada et al. [S75], a UML profile is used to represent services and their non-functional requirements. They applied feature models to describe non-functional requirements and their variability and constraints relations in a separate model. A service application is generated according to configuration and UML models defined in the UML profile. Abu-Matar and Gomaa [S1] proposed a feature-driven multi-view variability modeling technique to model variability in service contracts, service composition, and service interfaces. Feature models are represented, using the UML, and SoaML is adopted to represent service contracts, composition and interfaces. Park et al. [S63] extended the meta-model of UML activity diagrams to integrate and show different kinds of functional variability in business processes. In their approach, variations were analyzed at different abstract levels and variability is refined by using several concepts – variation point, variation point type, variation point cardinality, and variants.

Variability notions, borrowed from SPLE, have been utilized to develop domain-specific languages to model variability in the SOA context. Karam et al. [S42] proposed a domain-specific visual language, called Visual Web Composition (Visual WebC), to demonstrate workflows and their variability. The proposed language consists of webpad elements, which are generic elements that can be replaced with Web services. Reinhartz-Berger et al. [S67] proposed an application-based, domain-modeling platform for expressing and managing variability in reference business process models for enterprise service-oriented applications. These models can be specialized and customized according to the requirements. Ye et al. [S79] proposed an XML-based language to model variability in business process models. At the business process model level, variations may exist in the activities, events, data, and gateways. Similarly, the approach proposed by Nguyen et al. [S61] aims at modeling variability in control flow, data flow, and message flow. Their approach extends BPMN to describe variability in a process-based service composition. In order to describe in a separate variability dimension, a new language called, Web Service Variability description Language (WSVL), based on the notion of features, was introduced. Bayer et al. [S11] proposed variability mechanisms to showcase families of business processes. Their approach extends process models with stereotype annotations which accommodate variability and are applied to both UML activity diagrams and BPMN. The business process model family is further configured and tailored by means of a configuration model to resolve the variation points and instantiate final service product. Boffoli et al. [S12] also modeled functional variability at the level of business process models. They used decision tables to capture variability and manage variations points and variants in process models.

In addition to modeling variability in the services, some approaches concentrate only on realization of variability in the services or validation of the variability models and service templates. Van Gurp and Savolainen [S34] present variability realization techniques that can be used in SOA context. They described nine variability realization techniques and proposed a process to select the right realization technique for implementing variability. Gröner et al. [S32] proposed an approach for validation of configurable process model describing the composition of services. Their approach employed description logic reasoning to validate the mapping between feature model describing the configuration space of services and composition of the services that are expressed by BPMN models in the domain engineering.

4.2.2. Service identification

A service can be seen as an abstract resource representing a capability of performing a task and providing a coherent functionality from the provider and requester entities[14,36]. Service identification is a critical challenge in early phases of the SOA life-cycle for developing service-oriented systems [30,37]. Service identification focuses on reusability and determination of what business functionalities should be provided by the target service in the application domain. Hence, it primarily involves identification of the right level of service abstraction and granularity, and previsioning of service capabilities from users' requirements.

Lee et al. [S50,S51] introduced a feature-oriented approach, adopted from SPLE, to analyze and identify reusable services. Feature analysis and modeling are employed to identify and group units of functionality (specified as features) to provide services at the right level of granularity in a service-oriented system. In [S71], Souer et al. discussed feature modeling for commonality analysis which leads to the identification of business process models and services in the context of content management systems and web applications. Galster et al. [S29] presented an approach for the identification of potential core services for service-oriented product lines based on a trade-off analysis between the added value of services and the structural stability degree of a system.

Kang et al. [S41] proposed a method for service identification for software product lines by using variability analysis and feature modeling. The proposed method provides guidelines about how to group features as service candidates and refine them to achieve a proper level of granularity, which specifies the scope of variability in functionalities exposed by services. A top-down approach for the systematic identification and documentation of service-oriented core assets is described by Medeiros et al. in [S4]. In their approach, variability and commonality analysis inspired by SPLE is adopted for service identification and analysis of similarities in functionalities of services in the early stages of service development. Furthermore, variability analysis is performed to generate architectural decision, which derives service customization.

In the work of Chen et al. [S23], feature-oriented analysis is employed to identify services to develop service-oriented systems. Their approach focuses on reengineering of service-oriented systems and restructuring legacy systems, where feature analysis of a particular system is performed and its results are used in service identification.

4.2.3. Service reuse

SOA, in which one of the key design principles is to provide pieces of reusable functionality to be exposed as service, utilizes services as fundamental elements to develop applications. To fully utilize the benefits of SOA, we need effective methodologies to support systematic, agile, and cost-effective reuse during the development of service-oriented applications. SPLE relies on a fundamental distinction of *development for reuse* and *development with reuse* to minimize reusability- an insight having been leveraged to improve design, development, and evolution of service-oriented applications [10,41].

In [S16], Butler postulated that SPLE can be employed to promote service engineering by systematic reuse. Lee and Kotonya [S49] offered an argument that SOA lacks support for systematic reusability, discussing that not adopting variability management approaches causes more difficulty to reuse services for mass customization and to achieve effective construction, maintenance and evolution of similar software service products. Hence, a motivation to improve reusability and increase flexibility leads to integrating SPLE with service-oriented development.

Bubak and Gomaa [S15] discussed how concepts of SPLE (e.g., static and dynamic feature modeling) can be leveraged to support and escalate service reuse. Their work also provided an argument on how an enterprise can possibly gain competitive advantage by coupling SOA capabilities with systematic and strategic reuse by adopting SPLE approaches. In [S81], Zaupa et al. also offered a discussion and reported lessons learned about creating Web applications based on SOA by following the principles of SPLE. Their work focused on an application generation and offered a method that consisted of a set of activities to define the application domain and service model and customize applications by applying feature modeling. In [S25], Diaz et al. [S25] described a feature-oriented method to develop and construct portlets as reusable services. Balzerani et al. [S10] proposed a software product line architecture for Web-based applications, built in a prescribed way on compositions of reusable service components with explicit variability management.

In [S7], Alferéz et al. argued about needs for systematic reuse and proposed a method to support reusability of Web Services and service compositions for mass-production environments. The proposed approach adopts SPL-based and model-driven development methods. Feature models are used to capture structural variability of business process models which are one of several ways to describe service composition. Alferéz et al. also described how the configuration of feature model, involving the selection and deselection of reusable features mapped to services, guided the construction of specific service composition. Street and Gomma [44] discussed architectural issues for reuse in SOA and, in particular, how to support sound architectural design practices necessary to build systems out of reusable services from ground up. In another work [S30], the same authors described an approach to exploit Web services to increase the reusability in SPLE.

Narendra et al. [S59] presented a formal approach to modeling variability at the business process and service levels to enhance and maximize reusability of SOA-based solutions. Their approach generates different variation of solutions to meet changing requirements. In [S40], Jiang et al. discussed reuse in service construction at the level of service interfaces and implementation in the process of developing families of Web Services. They aimed at improving the degree of service reusability in a product line of services by identifying and managing the points of variability, and as such, they proposed a patterned-based method based on UML to model and manage variability of services functionalities. Altintas et al. [S8] presented an approach that integrates product line architecture with reflective rule-based business-process modeling for service-oriented application development.

4.2.4. Service configuration and customization

Software-as-a-Service promises to enable software application vendors to deliver software functionalities in a flexible and scalable way to serve as many users with different business objectives as possible. Configuration and customization is another crucial issue in service development on account of different functional and quality requirements of users for particular domains or contexts. Most enterprises and service providers are inevitably obliged to tailor their service to meet requirements.

In [S55], Mietzner et al. used variability models, described by OVM [41], to systematically derive customization and deployment of services for individual users. In [S54], the same authors described how to generate customized processes and create deployable services by means of explicit variability models that define variability descriptors for the process at the logic layer. They put forth how the variability descriptors are transformed into WS-BPEL process models that guide the customization. Sun et al. [S74] introduced a method to develop customizable Web services by incorporating the SPLE principles into service composition. The proposed approach adopts domain and application engineering processes from SPLE to enable users to construct and customize composite Web services. Acher et al. [S2] proposed an approach to managing services as a product line architecture. Their approach uses multiple feature models to structure information concerning service variability. A proposed set of composition operations enables a composition of feature models for further configuration of composite services. In work of Nunes et al. [S62], SPLE and SOA were incorporated to develop service-oriented user agents. The proposed approach compromises the activities to construct and develop customized agents encapsulating functionalities of services that automate users' tasks.

Hadaytullah et al. [S35] proposed an approach to modeling and managing variability in business process models for service customization. UML activity diagrams are utilized to model and manage business process models. A customization is enacted and performed by using pre-defined specialization rules. In [S42], Karma et al. adopted a lightweight product line model to perform calibration and customization of Web service-based Web applications. The proposed approach aimed at facilitating the composition and customization of service for specific users. Their approach supports agile methods, which were particularly relied on a domain-specific visual language.

Mohabbati et al. [S57] proposed a feature-driven approach for the configuration and optimization of reference business process models and derivation of customized composite service according to users' functional and non-functional requirements. In [S77], Wang et al. used feature modeling to model variability in composite services in order to enable end-user service customization corresponding to variable requirements. Utilizing variability modeling for decision making, configuration and selection of alternative composition of services is also discussed by Petersen et al. [S66].

Nguyen and Colman [S60] proposed a feature-oriented approach to customize Web service interface. The feature model is used to describe variability and variation points existing in the service interface, which guides the customization of service. In [S72], Stollberg et al. proposed a method for service customization by variability modeling of service specifications, i.e., optional operations, message types as well as their dependencies.

4.2.5. Dynamic software product line

Dynamic Software Product Lines (DSPLs), as specific type of product lines [1], focus on the development model for adaptable and dynamically configurable software systems. DSPLs' goal is to support variability management at run-time (i.e., dynamically (re)binding variation points), variable quality and performance management, (re)configuration, and dynamic product derivation. Moreover, DSPL approaches aims at identifying reusable and dynamically reconfigurable core assets at development time, which are explicitly modeled as dynamic variability [18,1]. Hence, from service-oriented perspectives, service characteristics (e.g., dynamic discoverability and binding) can be leveraged to realize and implement product line architectures and support the development of DSPLs.

Lee and Kotonya [S49] posit that a service-oriented product line is a DSPL domain application. In their approach, features, which provide functionalities of applications, can be mapped to activities of workflow models or dynamic services by means of service analysis [S46,S48]. Hence, the variation of the features can be dynamically bound during run-time by means of selecting services according to their quality.

In [S64], Parra et al. argued that adopting SPLE to construct context-aware system based on SOA enabled a complete service development ranging from requirements to implementation. SPLE also enabled the context management throughout the software lifecycle. Parra et al. proposed a homogeneous context-aware DSPLs to build service-oriented applications and adapt them at run-time in accordance to their usage context. They, in another work [S65], proposed a feature-oriented approach to automating the derivation of product architectures from feature configuration by combining model-driven and aspect-oriented development. In their approach, service-components architectures are employed for a target platform. In [S36], Hallsteinsen et al. also discussed that the integration of SOA and SPLE principles enabled DSPLs and provided a suitable development model for configurable systems.

Cetina et al. [S20] utilized model-driven development and SPLE for modeling run-time variability, adaptation policies and reconfiguration of services. Some guidelines to design and implement a DSPL in the context of autonomic and smart homes is given in [S18,S19].

Istoan et al. [S39] investigated the feasibility of synergy between SOA and DSPLs in the context of home automation systems. They discussed significant advantages to address DSPL by combining converging synergy points of SOA and SPL. To this end, they proposed a service-based middleware designed to solve issues such as devices interoperability, run-time variability binding, and linkage facilities. In the research reported in [S70], a feature-oriented method to support run-time variability reconfiguration is proposed by Shen et al. They used feature model to capture run-time variations and their dependencies in order to provide high-level business views for adaptation.

In [S80], Yu et al. argued that SOA provided dynamic capabilities needed in many product lines while SPLE approaches enabled useful mechanisms to model dynamic applications implemented through service compositions. Their proposed approach was to build and adapt dynamic service applications based on these two paradigms, including three main phases: domain engineering encompassing the phase to define reference architectures as well as domain specific and reusable core assets based on services; application engineering that defined application architectures according to particular requirements; and run-time phase executed autonomously and configure requirement-based services.

4.2.6. Adaptive systems

One of the most promising characteristics of adaptive software systems is their ability to optimize service provisioning and automatically adapt their behavior at run-time based on *the environment* and guided by *objectives* and needs of the users. The typical goal of the adaptations is to provide the best possible services to users according to the users' requirements within a particular context and resource constraints and to support fault tolerance in a dynamic environment [25].

Elkhodary et al. [S27] discussed the notion of features and the role of feature modeling techniques from SPLE to alleviate the challenging difficulties of building self-adaptive systems. They showed that feature modeling could provide an effective mechanism for analytical modeling, which could capture interactions and identify conflicts among the goals in a system. Feature modeling enabled to manage the configuration space of an adaptive system and for reconfiguration of the adaptive system. In another work of [S26], they utilized feature models, specified at design-time and employed as analytical and decision modelsmodels by which to evaluate behavior and characteristics of a service-oriented application at run-time to make appropriate adaptation decisions. In [S17], Cetina et al. proposed a method based on the principles SPLE to develop adaptive pervasive service-oriented applications. In a like manner, feature models, as decision models for the run-time, are employed to guide the dynamical reconfiguration of an application. Acher et al. [S3] applied feature modeling to define the adaptation rules to determine which configuration of the adaptive system to be executed in each specific context. In [S78], White et al. employed a model-driven approach to managing complexity of developing adaptive services. Their approach used high-level adaptive models, defined by means of feature models, to derive a new service composition when a service failed.

A service-oriented analysis and design method to develop adaptable services is presented in [S21,8,S22] by Chang et al. The proposed method consider three types of variation points in service design: workflow, service composition, and business logic. At design-time, service variability is modeled and designed into service components and compositions to support service adaptation. By following SPLE concepts, Kim et al. [S45] used variability modeling to design reusable services requiring run-time adaptation and evolution.

In the work of [S52], Lee et al. employed SPLE concepts (e.g., variation points and variants) for service decision modeling of content adaptable services in the context of pervasive computing. Lee et al. argued that using SPLE enabled to define systematically decision strategies needed for adaptation. Alferez et al. [S6] proposed a method to design and implement context-aware autonomous services. In their method, variability models represented by feature models are used as adaptation policies to generate automatically execution plans and reconfigure service compositions. Mohabbati et al. [S56] and Kaviani et al. [S44] proposed a method for adaptable service composition in the context of ubiquitous computing by leveraging SPLE principals of variability modeling. In those two methods, feature models annotated by domain knowledge were used as decision models for run-time service composition and adaptation according to functional capability supported by devices in environments. Marinho et al. [S53] adopted feature-oriented approach based on SPLE principles to achieve dynamic adaptation and reconfiguration in the development of mobile and context-aware applications.

In [S8], an approach was proposed to integrate SPLE principles and rule-based business process management with architectural point of view being the main idea behind integration and proposed to enhance run-time adaptability. Their approach eas set up to support dynamic composition of feature aspects. In their proposed framework, rules and rule sets are expressed in terms of dynamic aspect and delegated facts, which are implemented with adaptive object model. Allsteinsen et al. [S37] proposed an approach to building adaptive service-oriented applications based on SPLE principlesthrough the employment of explicit variability models and product line architectures at run-time to deal with complexity of adaptation by an adaptation platform. His proposed platform provided a conceptual model and reference architecture for the adaptive applications. In the work of [S24], Clotet et al. presented an approach that integrated goal-oriented modeling and variability modeling techniques to support runtime monitoring and adaptation and argued that adaptive systems cannot be fully pre-specified because of variabilities and changes in users' requirements. Therefore, design and modeling techniques are required to describe evolvable system to systematically specify adaptation rules as far as possible in a declarative way.

4.3. Application domain

In this mapping study, we identified 13 general application domains to address the research question RQ3 by adopting and extending software application domain taxonomy [17,23]: finance and accounting; ubiquitous and pervasive computing; scientific; wholesale, retail, and distribution; mobile application service; education; government, military; insurance; health and medical service; telecommunication service; automation (including smart home, hotel, and vehicles services); office automation; management information system, web-based systems and applications. Fig. 7 shows the mapping of application domain and its distribution.

4.4. Contribution facets

Fig. 8 depicts a map of studies with respect to research foci, which is distributed over research contribution and type (RQ4) according to classification scheme. The vertical axis describes the research focus, and the left and right horizontal axis shows types and contributions respectively. The scatterplot represents the interconnected frequencies in each class intersections. The size of circles is proportional to the number of studies within the pair of classes. The number of studies on each side may differ, in view of the fact that some studies provide multiple contributions.

Different types of contribution are made by publications in this study. The contribution types are adopted and extended from [47]. We categorized following classes with respect to study descriptions: *Metrics*, *Tooling support*, *Method*, *Model*, *Language*, *Open items*, *Analytical discussion*, and *Comparison analysis*. *Metrics* concentrates on what to measure to characterize certain properties of the SO and SPLE integration. Tooling support refers to any kind of tools supporting integration of SO and SPLE, with tools mainly including research prototypes. *Method* includes descriptions of two processes: (1) a proposes approach and/or technique; and (2) the method of performing the integration. It may sometimes also involve a third process: the adaptation of one paradigm into another. The *Models* and *Languages* categories include any modeling methods or representation of information to be used by adopting existing languages or by introducing new languages to be used in the integration of SO and SPLE. *Open items* typically discuss open challenges and issues for integration of SO and SPLE. *Analytical discussion* includes papers which provided discussions or theoretical models; or taxonomies or categorization of concepts. *Comparison analysis* includes papers whose research is based on either a survey or categorization and comparison of existing approaches available in the literature.

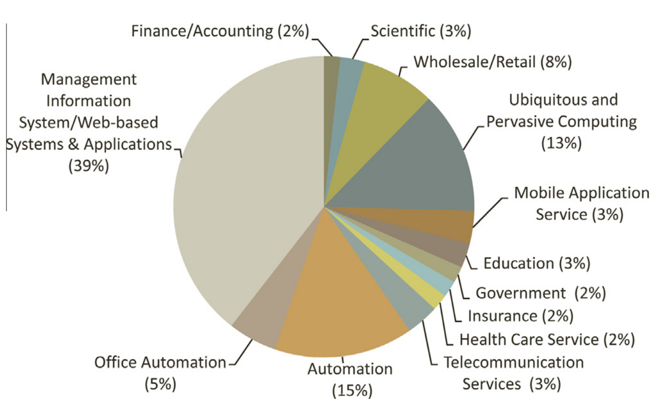

Fig. 7. Distribution of application domains.

The classification of research types is based on [46,16]. Research types are independent from specific focus areas and are categorized into the following classes [46,40]: (1) *Evaluation research*, which assesses a problem or implemented solution in practice and includes methods such as case study or field experiments; (2) *Validation research*, which focuses on investigating and proposing a novel technique or a solution not yet implemented in practices. Research is performed systematically using methods such as rigorous analysis, experiments, and simulations; (3) *Solution proposal*, which can be either a novel or significant extension of an existing approach or technique. In this category, pros and cons of research results are typically exemplified and critically discussed; (4) *Conceptual proposal* portrays new perspectives of existing phenomena and structuring the field in form of a taxonomy or a conceptual framework; (5) *Opinion-oriented papers* report authors positions on certain approaches; and (6) *Experience reports* offer insights in experience gained from one or more real-life projects.

5. Threats to validity

On account of the fact that there are some potential threats to the validity of the present mapping study and its results, the validity of the results from the mapping study has to be evaluated. Owing to the fact that subjective measurements are involved in the process of searching and collection, selecting studies, defining classification scheme, data extractions, and data analysis in the course of review, we need to address aspects of the validity and limitations of this study. To validate the credibility of the results, three types of threats should be discussed, as suggested by Perry et al. [39]: (1) *Construct validity* refers to what extend the inferences can be made and to what is investigated with respect to the research questions understudy; (2) *Internal validity* focuses on design and enactment of the study; and (3) *external validity* refers to the extent which the effects observed in the study are applicable to outside of the context of the study and can be generalized.

In terms of construct validity, the research questions and objectives of the present study is explicitly defined in the review protocol, which helps to achieve the same interpretations for other researchers to replicate the review if needed in the future. To formulate research questions and to derive the search strings, we took into account PICOC criteria, as recommended by [26] and adopted commonly in other mapping studies in different disciplines. We selected the digital libraries that include a very large set of publications in the software engineering fields. Furthermore, we collected relevant conference proceedings and journals as well as sources in which studies concerning service computing and software product lines are normally published. We defined and refined search queries based on the obtained results to maximize the selection of relevant papers for mapping study. In addition, we considered syn-

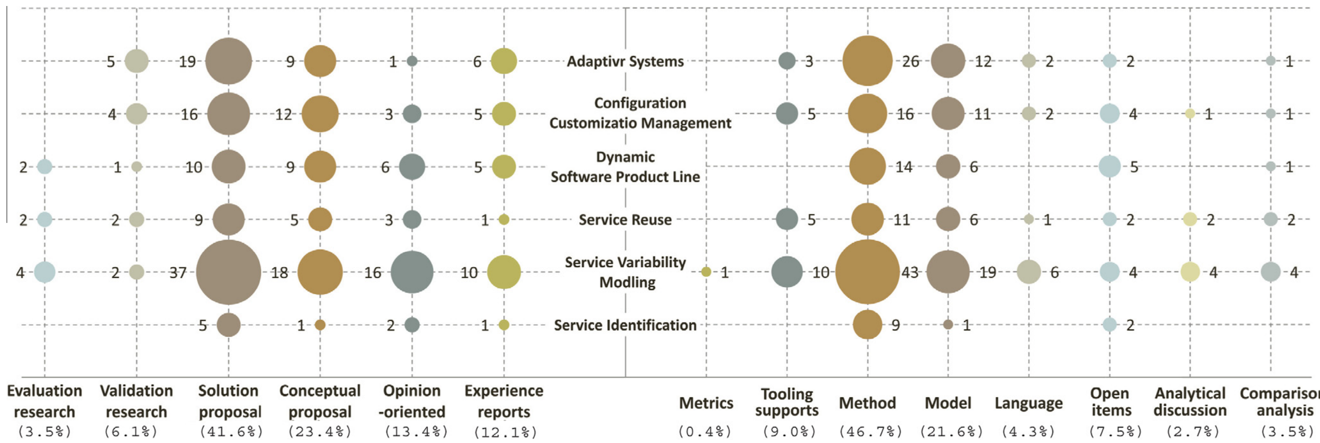

Fig. 8. Map of research foci with respect to contribution and research types.

onyms and the lexeme of the words. We applied the procedure which may be applied by the other researchers.

Moreover, another aspect of construct validity is to ascertain that all the relevant publications on the selected topic are included. Therefore, we incorporated as many studies as possible to invest exclusively on reliable information, excluding "gray publications" such as short papers, works in progress, unpublished, or unpreviewed literature [26] with the possibility of having left out some relevant studies; thus, we do not guarantee that all the primary studies have been covered although a list of all identified studies is available as a corpus reference[3] for the interested readers.

One way to address both internal and external validity threats is to replicate the mapping study [42]. In terms of the internal validity, the main limitation is increasing of bias which is due to the fact that synthesizing the extracted data is quite subjective. However, to mitigate threats, we defined and revised a review protocol given in Section 2.1.

Another source of threats is the classification scheme on account of the fact that schemes are subjective with no agreement about which one is the best. Furthermore, the classification is based on the research focus and contributions of studies according to terminology used by authors. That many researchers utilize different terminology to demonstrate their approaches is indicative of lack of standard terminology. Our classifying related studies with slightly different terminology in the same class was a measure we took against this threat. Another measure was to assign different classes to the studies possibly having multiple contributions. it is conceivable that others researchers may have encountered similar threats with different schemes in general; nonetheless, the authors verified the consistency of classifications of the first authors and both validated and cross-checked by other researchers.

Concerning the external validity threats, the result of the systematic mapping study were considered with respect to approaches given in both SO and SPLE topics. Thus, the classification presented and the conclusion drawn are only valid in the given context.

6. Discussion

SO engineering and SPLE have co-existed for a long time and a fair amount of research has been dedicated to each area by two different research communities. The research trend indicates that the combination and integration of both concepts, principles, and approaches are considered a new research direction [9,11,12,15,31–33]. To the best of our knowledge, there has been no mapping study in both SO and SPLE topics. Therefore, we conducted a systematic mapping study to provide a classification of research objectives and an overview of the existing studies that have investigated the integration of SO and SPLE principles, methods and approaches. The collected data presented indicates a growth in the number of works published in this area in the coming year, as illustrated by the trendline in Fig. 3.

The results of systematic mapping study enabled us to identify what research topics had pursued in the available literature. The primary research objectives and focuses of the identified studies in this field have been on service variability modeling, service reuse, service identification, service configuration and customization, dynamic software product line, and adaptive systems. The thematic analysis of mapping study shows that the majority of studies focused on service variability modeling (38%) and adaptive systems (19%) by applying SPLE principles and approaches. Moreover, tha majority of proposed approaches had been pursued the advantage of feature-oriented and methodologies introduced by SPLE. We identified that this body of research works aims at achieving resilient and effective modeling of service-oriented systems, supporting configurability in dynamic and adaptive systems, and reducing the system complexity. Whereas, SOAs are adopted in the SPLE context to reconcile the flexibility, scalability and dynamism in product line deviations and creating DSPL.

Furthermore, the classification scheme, which was applied in this study, paved the way for us to infer that the majority of studies are solution proposals (41.4%) and conceptual proposals (23.4%) with the primary focus on variability modeling and management. Some research foci present a relevant amount of entries in this mapping study; however, evaluation research and validation are weakly addressed in these studies. Our observation has pinpointed the shortcoming as caused by the recent research literature being in preliminary state or proposing some ways to integrate and combine SO and SPLE However, we believe there is need for the evaluation and validation of approaches, for instance, in industrial settings from a practitioners point of view.

The mapping study indicates that the combination of SO and SPLE are promising to help develop architectures for adaptive systems in responding effectively to dynamic functional and non-functional requirements, where the advantages stems from SO principles. Furthermore, it enables architectures to be reused in different instances, and supports variability managements, configuration and customization, where benefit stem from SPLE principles. In addition, dynamic software-product lines have been presented as a new direction in SPLE to deal with run-time adaptation to changing requirements and evolving context at different binding times. Hence, SOAs can be utilized to implements software core assets of product lines as services. From this perspective, dynamic behavior of service can reconcile dynamic characteristic into

[3] http://corr-sospl.sourceforge.net/.

SPL architecture to support DSPLs. It is noticeable that the research community with even specific venues, such as the Dynamic Software Product Line (DSPL) workshop at SPLC, has begun to explore variability management and adaptation at run-time. From the SPLE perspective, this is motivated by the need for more consolidated approaches that address run-time variability. Nevertheless, the results indicated that SPLE approaches for SO are still at an early stage and gaining maturity.

Our observation reveals that the identified studies mostly proposed a method (46.7%) or model (21.6%). There is lack of tooling supports with only 9% of included studies reporting some form of tooling supports. Similarly, we realize that research foci on service configuration-customization management and adaptive systems are topics with more entries than the other research foci with the distribution across the solution and conceptual proposals.

7. Conclusion and future work

This article provides an overview of existing research works that have applied or investigated the integration of SO and SPLE principles and approaches. We used a systematic mapping study method in order to achieve a classification of research objectives and overview of existing adopted approaches. Even though the results mark an escalation of works, they are a far cry from acceptably and reasonably high level of maturity. Our observation reveals that the integration or adoption of SO in SPLE (especially related to DSLPs) is
under-researched and a promising orientation for future research.

Our future work will embody our intention to conduct a full systematic literature review to investigate solutions to bridge the gaps between two paradigms and provide details and classification of approaches that have been applied. It is noteworthy that the mapping study thus far will serve the further study.

Appendix A

Fig. A.1.

Fig. A.1. A tag cloud generated based on frequent words retained following publications including keywords.

Appendix B. Publication fora

Tables B.1, B.2, B.3.

Table B.1
List of journals.

Acronym	Journal title
CDBI	CBDI Journal
IEEE Softw.	IEEE Software
IJBPIM	Journal of Business Process Integration and Management
IJCIS	Journal of Cooperative Information Systems
IJIIDS	Journal of Intelligent Information and Database Systems
IJSEKE	Journal of Software Engineering and Knowledge Engineering
IJSSSE	Journal of Systems Science and Systems Engineering
IJWIS	Journal of Web Information Systems
JDM	Journal of Database Management
JRPIT	Journal of Research and Practice in Information Technology
JSS	Journal of Systems and Software
JUCS	Journal of Universal Computer Science

Table B.2
List of workshops.

Acronym	Workshops title
ACoTA	International Workshop on Automated Configuration and Tailoring of Applications
BPMW	Business Process Management Workshops
DSPL	International Workshop on Dynamic Software Product Lines
FOSD	International Workshop on Feature-Oriented Software Development
IWPSE	International Workshop on Principles of Software Evolution
MAPLE	International Workshop on Model-Driven Product Line Engineering
MRT	International Workshop Models@run.time at Models
PESOS	International Workshop on Principles of Engineering Service Oriented Systems
RAM-SE	International Workshop on Reflection, AOP and Meta-Data for Software Evolution
SCArVeS	International Workshop on Services, Clouds, and Alternative Design Strategies for Variant-Rich Software Systems
SDSOA	International Workshop on Systems development in SOA environments
SOAPL	International Workshop on Service-Oriented Architectures and Software Product Lines
SOCCER	International Workshop on Service-Oriented Computing: Consequences for Engineering Requirements
SWESE	International Workshop on Semantic Web Enabled Software Engineering
TSOA	International Workshop on Telecom Service Oriented Architectures
VaMOS	International Workshop on Variability Modelling of Software-Intensive Systems
WESOA	International Workshop on Engineering Service-Oriented Applications
WWV	International Workshop on Automated Specification and Verification of Web Systems

Table B.3
List of conferences and symposiums.

Acronym	Conferences title
AICCSA	ACS/IEEE International Conference on Computer Systems and Applications
APSCC	Asia–Pacific Services Computing Conference
APSEC	Asia–Pacific Software Engineering Conference
BIS	International Conference on Business Information Systems
CAiSE	International Conference Advanced Information Systems Engineering
CBSE	International Conference on Component-Based Software Engineering
CEC	International Conference on E-Commerce Technology
CloudCom	International Conference on Cloud Computing
COMPSAC	International Computer Software and Applications Conference
ECBS	International Conference on Engineering of Computer Based Systems
ECWS	European Conference on Web Services
ECSA	European Conference on Software Architecture
FSE	International Symposium on Foundations of software engineering
GCC	International Conference on Grid and Cooperative Computing
HICSS	Annual Hawaii International Conference on System Sciences
ICACT	International Conference on Advanced Communication Technology
ICCIT	International Conference on Computer and Information Technology
ICEBE	International Conference on E-Business Engineering
ICHIT	International Conference on Convergence and Hybrid Information Technology
ICIS	International Conference on Computer and Information Science
ICSOC	International Conference on Service-Oriented Computing
ICSR	International Conference on Software Reuse
ICWS	International Conference on Web Services
ISCID	International Symposium on Computational Intelligence and Design
ITNG	International Conference on Information Technology
PROFES	International Conference on Product Focused Software Process Improvement
SAC	ACM symposium on Applied Computing
SBCARS	Brazilian Symposium on Software Components, Architectures and Reuse
SC	International Conference on Software Composition
SCC	International Conference on Services Computing
SEKE	International Conference on Software Engineering & Knowledge Engineering
SERVICES	World Congress on Services
SOCA	International Conference on Service-Oriented Computing and Applications
SPLC	International Software Product Line Conference
TEAA	International Conference on Trends in enterprise application architecture
WWW	International Conference on World Wide Web

References

[1] J. Andersson, J. Bosch, Development and use of dynamic product-line architectures, in: IEE Proceedings-Software 152.1, 2005, pp. 15–28.
[2] S. Apel, C. Kaestner, C. Lengauer, Research challenges in the tension between features and services, in: Proceedings of the 2nd International Workshop on Systems Development in SOA Environments (SDSOA '08), ACM, Leipzig, Germany, 2008, pp. 53–58.
[3] A. Arsanjani et al., SOMA: a method for developing service-oriented solutions, IBM Systems Journal 47 (3) (2008) 377–396.
[4] D. Batory, Feature models, grammars, and propositional formulas, Software Product Lines (2005) 7–20.
[5] P. Brereton et al., Lessons from applying the systematic literature review process within the software engineering domain, Journal of Systems and Software 80 (4) (2007) 571–583.
[6] D. Budgen et al., Using mapping studies in software engineering, in: PPIG'08, Lancaster University, 2008, pp. 195–204.
[7] D. Budgen et al., Using mapping studies in software engineering, in: Proceedings of PPIG 2008, Lancaster University, 2008, pp. 195–204.
[8] S.H. Chang, S.D. Kim, A variability modeling method for adaptable services in service-oriented computing, in: Proceedings of the 11th International Software Product Line Conference, IEEE, 2007, pp. 261–268.
[9] A. Classen et al., Modelling variability in self-adaptive systems: towards a research agenda, in: Proc. of International Workshop on Modularization, Composition and Generative Techniques for Product-line Engineering, 2008, pp. 19–26.
[10] P. Clements, L. Northrop, Software product lines: practices and patterns, 2001.
[11] S.G. Cohen, R. Krut, Managing Variation in Services in a Software Product Line Context, Tech. rep. CMU/SEI-2010-TN-007, Software Engineering Institute, Carnegie Mellon University, May 2010.
[12] S.G.S. Cohen, R.W. Krut, Proc. of the 1st Workshop on Service-Oriented Architectures and Product Lines: What is the Connection? (SOAPL'07), Tech. rep. CMU/SEI-2008-SR- 006, Carnegie Mellon University, 2008.
[13] O. Dieste, A.G. Padua, Developing search strategies for detecting relevant experiments for systematic reviews, in: Proceedings of the First International Symposium on Empirical Software Engineering and Measurement (ESEM '07), IEEE Computer Society, Washington, DC, USA, 2007, pp. 215–224.
[14] T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall PTR, 2005. ISBN: 0131858580.
[15] M. Galster, Describing variability in service-oriented software product lines, in: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ACM, Copenhagen, Denmark, 2010, pp. 344–350.
[16] R. Glass, I. Vessey, V. Ramesh, Research in software engineering: an analysis of the literature, Information and Software Technology 44 (8) (2002) 491–506. ISSN: 0950-5849.
[17] R.L. Glass, I. Vessey, Contemporary application-domain taxonomies, IEEE Software 12 (4) (1995) 63–76. ISSN: 0740-7459.
[18] S. Hallsteinsen et al., Dynamic software product lines, in: Computer 41.4, 2008, pp. 93–95.
[19] J.E. Hannay, D.I.K. Sjoberg, T. Dyba, A systematic review of theory use in software engineering experiments, in: IEEE Trans. Softw. Eng. 33.2, February 2007, pp. 87–107. ISSN: 0098-5589.
[20] A. Helferich, G. Herzwurm, S. Jesse, Software product lines and service-oriented architecture: a systematic comparison of two concepts, in: Proceedings of the 1st Int. Workshop on Service-Oriented Architectures and Software Product Lines (SOAPL'07), Kyoto, Japan, 2007, p. 21.
[21] J. Higgins, S. Green, C. Collaboration, Cochrane Handbook for Systematic Reviews of Interventions, vol. 5, Wiley Online Library, 2008.
[22] P.A. Istoan Software Product Lines and Service Oriented Architectures: can they be connected? Tech. rep. IRISA, 2009.
[23] C. Jones, Software Assessments, Benchmarks, and Best Practices, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2000. ISBN: 0-201-48542-7.
[24] K. Kang et al., FORM: a feature-; oriented reuse method with domain-; specific reference architectures, in: Annals of Software Engineering 5.1, 1998, pp. 143–168.
[25] R. Kazhamiakin et al., Adaptation of service-based systems, Service Research Challenges and Solutions for the Future Internet, vol. 6500, Springer, Berlin, Heidelberg, 2010, pp. 117–156.
[26] B. Kitchenham, S. Charters, Guidelines for performing Systematic Literature Reviews in Software Engineering, Tech. rep. EBSE 2007-001, Keele University and Durham University Joint Report, 2007.
[27] B. Kitchenham et al., Systematic literature reviews in software engineering: a systematic literature review, Information and Software Technology 51 (1) (2009) 7–15.
[28] B.A. Kitchenham, D. Budgen, O. Pearl Brereton, Using mapping studies as the basis for further research: a participant-observer case study, Information and Software Technology 53 (6) (2011) 638–651. ISSN: 0950-5849.
[29] B.A. Kitchenham et al., Preliminary guidelines for empirical research in software engineering, IEEE Transactions on Software Engineering 28 (8) (2002) 721–734. ISSN: 0098-5589.
[30] T. Kohlborn et al., Identification and analysis of business and software services & #8212; a consolidated approach, IEEE Transactions on Services Computing 2 (1) (2009) 50–64. ISSN: 1939-1374.
[31] R.W. Krut, S.G. Cohen, 2nd Workshop on service-oriented architectures and software product lines: putting both together (SOAPL'08), in: Proceedings of the 12th International Software Product Line Conference, Carnegie Mellon University, San Francisco, California, 2009, pp. 115–147.
[32] R. W Krut, S.G. Cohen, 3rd Workshop on service-oriented architectures and software product lines: enhancing variation (SOAPL'09), in: Proceedings of the 13th International Software Product Line Conference, Carnegie Mellon University, San Francisco, California, 2009, pp. 301–302.
[33] J. Lee, G. Kotonya, Combining service-orientation with product line engineering: challenges and practices, IEEE Software 99 (1) (2010) 35–41.
[34] E. Murugesupillai, B. Mohabbati, D. Gašević, A preliminary mapping study of approaches bridging software product lines and service-oriented architectures, in: Proceedings of the 15th International Software Product Line Conference (SPLC '11), ACM, New York, NY, USA, 2011, pp. 11:1–11:8.
[35] T. Nguyen et al., Managing service variability: state of the art and open issues, in: Proceedings of the 5th Workshop on Variability Modeling of Software-Intensive Systems (VaMoS '11), ACM, New York, NY, USA, 2011, pp. 165–173.
[36] M.P. Papazoglou, Service-oriented computing: concepts, characteristics and directions, in: International Conference on Web Information Systems Engineering, 0, 2003, p. 3.
[37] M. Papazoglou, W. Van Den Heuvel, Service-oriented design and development methodology, International Journal of Web Engineering and Technology 2 (4) (2006) 412–442. ISSN: 1476-1289.

[38] M. Papazoglou et al., Service-oriented computing: a research roadmap, International Journal of Cooperative Information Systems 17 (2) (2008) 223–255.
[39] D. Perry, A. Porter, L. Votta, Empirical studies of software engineering: a roadmap, in: Proceedings of the Conference on the Future of Software Engineering, ACM, 2000, pp. 345–355.
[40] K. Petersen et al., Systematic mapping studies in software engineering, in: EASE '08: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, 2008.
[41] K. Pohl, G. Böckle, F.J. van der Linden, Software Product Line Engineering: Foundations, Principles and Techniques, Springer-Verlag Inc., New York, 2005. ISBN: 3540243720.
[42] F. Shull et al., The role of replications in empirical software engineering, Empirical Software Engineering 13 (2) (2008) 211–218.
[43] M. Sinnema et al., Covamof: a framework for modeling variability in software product families, Software Product Lines (2004) 25–27.
[44] J. Street, H. Gomaa, Software architectural reuse issues in service-oriented architectures, in: Proceedings of the 41st Annual Hawaii International Conference on System Sciences. IEEE, January 2008, pp. 316–316.
[45] M. Svahnberg, J. Van Gurp, J. Bosch, A taxonomy of variability realization techniques, Software: Practice and Experience 35 (8) (2005) 705–754.
[46] R. Wieringa et al., Requirements engineering paper classification and evaluation criteria: a proposal and a discussion, in: Requirements Engineering 11.1, November 2005, pp. 102–107. ISSN: 0947-3602.
[47] M. Zelkowitz, D. Wallace, Experimental models for validating technology, Computer 31 (5) (1998) 23–31.
[48] L.-J. Zhang et al., Variation-oriented analysis for SOA solution design, in: IEEE International Conference on Services Computing, 2007, pp. 560–568.

Reference of studies included in mapping study

[S1] M. Abu-Matar, H. Gomaa, Variability modeling for service oriented product line architectures, in: Proceedings of the 15th International Software Product Line Conference (SPLC '11), IEEE Computer Society, Washington, DC, USA, 2011, pp. 110–119.
[S2] M. Acher et al., Managing variability in worklow with feature model composition operators, in: Proceedings of the 9th International Conference on Software Composition (SC'10), LNCS, Springer, 2010, p. 16.
[S3] M. Acher et al., Modeling context and dynamic adaptations with feature models, in: Proceedings of the 4th International Workshop Models @run.time at Models (MRT'09), Denver, Colorado, USA, 2009, p. 10.
[S4] F.M. Medeiros et al., Towards an approach for service-oriented product line architectures, in: Proceedings of the 3rd International Workshop on Service-Oriented Architectures and Software Product Lines (SOAPL'09), San Francisco, CA, USA, 2009, pp. 151–164.
[S5] F.M. Medeiros et al., Designing a set of service-oriented systems as a software product line, in: Proceedings of the 2010 Fourth Brazilian Symposium on Software Components, Architectures and Reuse (SBCARS '10), IEEE Computer Society, Washington, DC, USA, 2010, pp. 70–79.
[S6] G. Alferéz, V. Pelechano, Context-aware autonomous web services in software product lines, in: 2011 15th International Software Product Line Conference (SPLC), August 2011, pp. 100–109.
[S7] G. Alferez, V. Pelechano, Systematic reuse of web services through software product line engineering, in: 2011 Ninth IEEE European Conference on Web Services (ECOWS), September 2011, pp. 192–199.
[S8] N.I. Altintas, S. Cetin, Integrating a software product line with rule-based business process modeling, in: Proceedings of the 31st VLDB Conference on Trends in Enterprise Application Architecture (TEAA'05), Springer-Verlag, Berlin, Heidelberg, 2006, pp. 15–28.
[S9] M. Asadi et al., Model-driven development of families of service-oriented architectures, in: Proceedings of the First International Workshop on Feature-Oriented Software Development, FOSD '09, ACM, Denver, Colorado, 2009, pp. 95–102.
[S10] L. Balzerani et al., A product line architecture for web applications, in: Proceedings of the 2005 ACM Symposium on Applied Computing (SAC '05), NY, USA, 2005, pp. 1689–1693.
[S11] J. Bayer et al., Process family engineering: Modeling variant-rich processes. Tech. rep. No. 18/2005, PESOA Report, 2005.
[S12] N. Boffoli et al., Managing SOA system variation through business process lines and process oriented development, in: Proceedings of the 3rd International Workshop on Service-Oriented Architectures and Software Product Lines (SOAPL'09), SEI, Carnegie Mellon University, 2009.
[S13] M. Bošković et al., Automated staged configuration with semantic web technologies, International Journal of Software Engineering and Knowledge Engineering 20 (4) (2010) 459–484.
[S14] M. Bošković et al., Developing families of software services: a semantic web approach, Journal of Research and Practice in Information Technology (2011).
[S15] O. Bubak, H. Gomaa, Applying software product line concepts in service orientation, International Journal of Intelligent Information and Database Systems 2 (4) (2008) 383–396.
[S16] J. Butler, Applying product line techniques to SOA, Part I and II, CDBI Journal (2006) 22–31.
[S17] C. Cetina, J. Fons, V. Pelechano, Applying software product lines to build autonomic pervasive systems, in: Proceedings of the 12th Software Product Line Conference, IEEE, 2008, pp. 117–126.
[S18] C. Cetina et al., Mass customisation along lifecycle of autonomic homes, in: Proceedings of the 3rd International Workshop on Dynamic Software Product Lines (DSPL'09), San Francisco, USA, 2009, pp. 43–50.
[S19] C. Cetina et al., Designing and prototyping dynamic software product lines: techniques and guidelines, in: Proceedings of the 14th international conference on Software product lines: going beyond (SPLC'10), Springer-Verlag, Jeju Island, South Korea, 2010, pp. 331–345.
[S20] C. Cetina et al., Strategies for variability transformation at run-time, in: Proceedings of the 13th International Software Product Line Conference, SPLC '09, Carnegie Mellon University, San Francisco, California, 2009, pp. 61–70.
[S21] S.H. Chang, S.D. Kim, A service-oriented analysis and design approach to developing adaptable services, in: IEEE International Conference on Services Computing (SCC'07), IEEE, SLC, USA, 2007, pp. 204–211.
[S22] S.H. Chang, H.J. La, S.D. Kim, A comprehensive approach to service adaptation, in: Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications, SOCA '07, IEEE, Washington, DC, USA, 2007, pp. 191–198.
[S23] F. Chen, S. Li, W.C.-C. Chu, Feature analysis for service-oriented reengineering, in: Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC '05), IEEE Computer Society, DC, USA, 2005, pp. 201–208.
[S24] R. Clotet et al., Dealing with changes in service-oriented computing through integrated goal and variability modelling, in: Proceedings of the 2nd International Workshop on Variability Modelling of Software-Intensive Systems (Va-MoS'08), Essen, Germany, 2008, pp. 43–52.
[S25] O. Díaz, S. Trujillo, S. Pérez, Turning portlets into services: the consumer profile, in: Proceedings of the 16th International Conference on World Wide Web, ACM, Banff, Alberta, Canada, 2007, pp. 913–922.
[S26] A. Elkhodary, N. Esfahani, S. Malek, FUSION: a framework for engineering self-tuning self-adaptive software systems, in: Proceedings of the 18th International Symposium on Foundations of Software Engineering, FSE '10, ACM, Santa Fe, New Mexico, USA, 2010, pp. 7–16.
[S27] A. Elkhodary, A. Malek, A. Esfahani, On the role of features in analyzing the architecture of self-adaptive software systems, in: Proceedings of the 4th International Workshop Models@run.time at Models (MRT'09), Denver, Colorado, USA, 2009.
[S28] M. Fantinato, M. De Toledo, I. De Gimenes, WS-contract establishment with QoS: an approach based on feature modeling, International Journal of Cooperative Information Systems 17 (3) (2008) 373–407.
[S29] M. Galster, A. Eberlein, Identifying potential core assets in service-based systems to support the transition to service-oriented product lines, in: Proceedings of the 18th International Conference and Workshops on Engineering of Computer-Based Systems, ECBS '11, IEEE Computer Society, Washington, DC, USA, 2011, pp. 179–186.
[S30] H. Gomaa, M. Saleh, Software product line engineering for Web services and UML, in: Proceedings of the ACS/IEEE 2005 International Conference on Computer Systems and Applications, AICCSA '05, IEEE Computer Society, DC, USA, 2005.
[S31] P. Greenwood et al., Modelling service requirements variability: the DiVA way, in: Service Engineering, Springer, Vienna, 2011, pp. 55–84.
[S32] G. Gröner et al., Validation of families of business processes, in: Proceedings of the 23rd International Conference on Advanced Information Systems Engineering, CAiSE'11, Springer-Verlag, London, UK, 2011, pp. 551–565.
[S33] A. Gruler, A. Harhurin, J. Hartmann, Development and configuration of service-based product lines, in: Proceedings of the 11th International Software Product Line Conference, IEEE, 2007, pp. 107–116.
[S34] J. van Gurp, J. Savolainen, Service grid variability realization, in: Proceedings of the 10th International on Software Product Line Conference, IEEE Computer Society, 2006, pp. 85–94.
[S35] H. Hadaytullah, K. Koskimies, T. Systa, Using model customization for variability management in service compositions, in: IEEE International Conference on Web Services, ICWS '09, IEEE, 2009, pp. 687–694.
[S36] S. Hallsteinsen, S. Jiang, R. Sanders, Dynamic software product lines in service oriented computing, in: Proceedings of the 3rd International Workshop on Dynamic Software Product Lines (DSPL '09), San Francisco, USA, 2009, pp. 28–34.
[S37] S. Hallsteinsen et al., Using product line techniques to build adaptive systems, in: Proceedings of the 10th International on Software Product Line Conference, SPLC '06, IEEE Computer Society, Washington, DC, USA, 2006, pp. 141–150.
[S38] A. Harhurin, J. Hartmann, Service-oriented commonality analysis across existing systems, in: Proceedings of the 12th International Software Product Line Conference, IEEE, Limerick, Ireland, 2008, pp. 255–264.
[S39] P. Istoan et al., Dynamic software product lines for service-based systems, Ninth IEEE International Conference on Computer and Information Technology (CIT '09), vol. 2, IEEE, 2009, pp. 193–198.
[S40] J. Jiang, A. Ruokonen, T. Systa, Pattern-based variability management in web service development, in: Proceedings of the 3rd European Conference on Web Services, ECOWS '05, IEEE Computer Society, Washington, DC, USA, 2005, pp. 83–94.
[S41] D. Kang, C.-y. Song, D.-K. Baik, A method of service identification for product line, in: Proceedings of the 3rd International Conference on Convergence and Hybrid Information Technology, ICCIT '08, IEEE Computer Society, Washington, DC, USA, 2008, pp. 1040–1045.

[S42] M. Karam et al., A product-line architecture for web service-based visual composition of web applications, Journal of Systems and Software 81 (6) (2008) 855–867.
[S43] A. Kattepur et al., Variability modeling and QoS analysis of web services orchestrations, in: Proceedings of the 2010 IEEE International Conference on Web Services, ICWS '10, IEEE Computer Society, DC, USA, 2010, pp. 99–106.
[S44] N. Kaviani et al., Semantic annotations of feature models for dynamic product configuration in ubiquitous environments, in: Proceedings of the 4th International Workshop on Semantic Web Enabled Software Engineering, 2008.
[S45] Y. Kim, K.-G. Doh, Adaptable web services modeling using variability analysis, in: Proceedings of the 2008 Third International Conference on Convergence and Hybrid Information Technology, ICCIT '08, IEEE Computer Society, Washington, DC, USA, 2008, pp. 700–705.
[S46] G. Kotonya, J. Lee, D. Robinson, A consumer-centred approach for service-oriented product line development, in: Joint Working IEEE/IFIP Conference on Software Architecture, 2009 European Conference on Software Architecture (WICSA/ECSA '09), IEEE, 2009, pp. 211–220.
[S47] H.J. La, S.D. Kim, A systematic process for developing high quality SaaS cloud services, in: Proceedings of the 1st International Conference on Cloud Computing, CloudCom '09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 278–289.
[S48] J. Lee, K. Kang, A feature-oriented approach to developing dynamically reconfigurable products in product line engineering, in: 2006 10th International Software Product Line Conference, 2006, pp. 10–140.
[S49] J. Lee, G. Kotonya, Combining service-orientation with product line engineering: challenges and practices, IEEE Software 99 (1) (2010) 35–41.
[S50] J. Lee, D. Muthig, M. Naab, A feature-oriented approach for developing reusable product line assets of service-based systems, Journal of Systems and Software 83 (7) (2010) 1123–1136.
[S51] J. Lee, D. Muthig, M. Naab, An approach for developing service oriented product lines, in: Proceedings of the 2008 12th International Software Product Line Conference, SPLC '08, IEEE Computer Society, Washington, DC, USA, 2008, pp. 275–284.
[S52] S.J. Lee, S.D. Kim, A rendezvous of content adaptable service and product line modeling, in: Proceedings of the 6th International Conference on Product Focused Software Process Improvement, PROFES'05, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 69–83.
[S53] F.G. Marinho et al., A software product line for the mobile and context-aware applications domain, in: Proceedings of the 14th International Conference on Software Product Lines, SPLC 10, Springer-Verlag, 2010, pp. 346–360.
[S54] R. Mietzner, F. Leymann, Generation of BPEL customization processes for SaaS applications from variability descriptors, Proceedings of the 2008 IEEE International Conference on Services Computing, vol. 2, IEEE Computer Society, Washington, DC, USA, 2008, pp. 359–366.
[S55] R. Mietzner et al., Variability modeling to support customization and deployment of multi-tenant-aware software as a service applications, in: Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems, PESOS '09, IEEE, 2009, pp. 18–25.
[S56] B. Mohabbati, N. Kaviani, D. Gašević, Semantic variability modeling for multi-staged service composition, in: Proc. of the 3rd Int. Workshop on Service-Oriented Architectures and Software Product Lines (SOAPL '09), vol. 2, San Fransico, CA, USA, 2009.
[S57] B. Mohabbati et al., Development and configuration of service-oriented systems families, in: Proceedings of the 2011 ACM Symposium on Applied Computing, SAC '11, ACM, TaiChung, Taiwan, 2011, pp. 1606–1613.
[S58] N.C. Narendra, K. Ponnalagu, Towards a variability model for SOA-based solutions, in: Proceedings of the 2010 IEEE International Conference on Services Computing, IEEE Computer Society, 2010, pp. 562–569.
[S59] N.C. Narendra et al., Variation-oriented engineering (VOE): enhancing reusability of SOA-based solutions, in: Proceedings of the 2008 IEEE International Conference on Services Computing, SCC '08, vol. 1, IEEE Computer Society, DC, USA, 2008, pp. 257–264.
[S60] T. Nguyen, A. Colman, A feature-oriented approach for web service customization, in: Proceedings of the 2010 IEEE International Conference on Web Services, ICWS '10, IEEE, DC, USA, 2010, pp. 393–400.
[S61] T. Nguyen, A. Colman, J. Han, Modeling and managing variability in process-based service compositions, in: Proceedings of the 9th International Conference on Service-Oriented Computing, ICSOC'11, Springer-Verlag, Paphos, Cyprus, 2011, pp. 404–420.
[S62] I. Nunes et al., Building service-oriented user agents using a software product line approach, in: Proceedings of the 11th International Conference on Software Reuse: Formal Foundations of Reuse and Domain Engineering (ICSR '09), Springer-Verlag, Falls Church, Virginia, 2009, pp. 236–245.
[S63] J. Park, M. Moon, K. Yeom, Variability modeling to develop flexible service-oriented applications, Journal of Systems Science and Systems Engineering 20 (2) (2011) 193–216.
[S64] C. Parra, X. Blanc, L. Duchien, Context awareness for dynamic service-oriented product lines, in: Proceedings of the 13th Int. Software Product Line Conference, San Fran-cisco,CA, USA, 2009, pp. 131–140.
[S65] C. Parra et al., Feature-based composition of software architectures, in: Proceedings of the 4th European conference on Software Architecture, ECSA'10, Springer-Verlag, 2010, pp. 230–245.
[S66] K. Petersen, N. Bramsiepe, K. Pohl, Applying variability modeling concepts to support decision making for service composition, in: Proceedings of the Service-Oriented Computing: Consequences for Engineering Requirements, IEEE Computer Society, DC, USA, 2006, p. 1.
[S67] I. Reinhartz-Berger, P. Soffer, A. Sturm, Organisational reference models: supporting an adequate design of local business processes, International Journal of Business Process Integration and Management 4 (2) (2009) 134–149.
[S68] A. Schnieders, F. Puhlmann, Variability mechanisms in e-business process families, in: Proceedings of International Conference on Business Information Systems, 2006, pp. 583–601.
[S69] S. Segura et al., A taxonomy of variability in web service flows, in: Proceedings of the First International Workshop on Service-Oriented Architecture and Software Product Lines (SOAPL'07), Kyoto, Japan, September 2007.
[S70] L. Shen et al., Towards feature-oriented variability reconfiguration in dynamic software product lines, in: Proceedings of the 12th International Conference on Top Productivity through Software Reuse, ICSR'11, Springer-Verlag, Pohang, South Korea, 2011, pp. 52–68.
[S71] J. Souer et al., Identifying commonalities in web content management system engineering, in: IJWIS 7.3, 2011, pp. 292–308.
[S72] M. Stollberg, M. Muth, Service customization by variability modeling, Service-Oriented Computing. IC-SOC/ServiceWave 2009 Workshops, vol. 6275, Springer, Berlin, Heidelberg, 2010, pp. 425–434.
[S73] C.-a. Sun et al., Modeling and managing the variability of Web service-based systems, Journal of Systems and Software 83 (3) (2010) 502–516.
[S74] H. Sun, R.R. Lutz, S. Basu, Product-line-based requirements customization for web service compositions, in: Proceedings of the 13th International Software Product Line Conference, SPLC '09, Carnegie Mellon University, San Francisco, California, 2009, pp. 141–150.
[S75] H. Wada, J. Suzuki, K. Oba, A feature modeling support for non-functional constraints in service oriented architecture, in: IEEE International Conference on Services Computing (SCC '07), IEEE, 2007, pp. 187–195.
[S76] H. Wada, J. Suzuki, K. Oba, Leveraging early aspects in end-to-end model driven development for non-functional properties in service oriented architecture, Journal of Database Management 22 (2) (2011) 93–123.
[S77] J. Wang, J. Yu, A business-level service model supporting end-user customization, Service-Oriented Computing – ICSOC 2007 Workshops, vol. 4907, Springer, Berlin, Heidelberg, 2009, pp. 295–303.
[S78] J. White, H.D. Strowd, D.C. Schmidt, Creating self-healing service compositions with feature models and mi-crorebooting, International Journal of Business Process Integration and Management 4 (1) (2009) 35–46.
[S79] E. Ye et al., An approach to designing service-oriented product-line architecture for business process families, The 9th International Conference on Advanced Communication Technology, vol. 2, IEEE, 2007, pp. 999–1002.
[S80] J. Yu, P. Lalanda, P. Bourret, An approach for dynamically building and managing service-based applications, in: Proceedings of the 2010 IEEE Asia–Pacific Services Computing Conference, APSCC '10, IEEE Computer Society, 2010, pp. 51–58.
[S81] F. Zaupa et al., A service-oriented process to develop web applications, Journal of Universal Computer Science 14 (8) (2008) 1368–1387.