

Safety-driven Software Product Line architectures Design, A Survey Paper

Mozamil Ebnauf Elgodbe
College of Graduate Studies, Computer Science and
Information Technology
Sudan University of science & technology,
Khartoum, Sudan

Hany H. Ammar
Lane Department of Computer Science and
Electrical Engineering, College of Engineering and
Mineral Resources West Virginia University
Morgantown, USA

Abstract: Software architecture design is an important step of software development. Currently, there are various design methods available and each is focusing on certain perspective of architecture design. Especially, quality-based methods have received a lot of attentions and have been well developed for single system architecture design. However, the use of quality-based design methods is limited in software product line (SPL) because of the complexity and variabilities existing in SPL architecture. With the increasing attention to software safety, improving software safety has already become a more important issue, especially for safety-critical systems. This study aims at surveying existing research on Software Product Line Architecture (SPLA) design based on quality attributes, and to give an overview of the intersection of the areas of software product line architecture design and Safety Driven Design in order to classifying existing work, and discover open issues for further research. Also this study investigates safety analysis at the architectural level, and Safety-based Software Product Line Architecture Design (SSPLAD) approaches. Safety-driven software product line architecture design seems to be a “discussion” topic. The study shows that there are a large number of SPLA design methods. However, the use of safety-based design methods is limited in software product lines (SPL) due to the variability property that can potentially result in a large number of possible systems and because of the complexity existing in safety attribute itself.

Keywords: Software Architecture Quality attributes; Safety analysis; Architectural Design; Software Product Line Architectures; Safety-driven software product line architecture design.

1. INTRODUCTION

A software architecture is the structure of the software system. "It describes the software elements, their characteristics and they interact with each other" [1],[2]. A qualified software architecture provides a blueprint for system construction and composition. It is a main factor to a successful software development [3]. There are many challenges in software architecture design for example, modeling the non-functional requirements, especially those requirements on the quality of the software.

Non-functional requirements and quality attributes (e.g. maintainability, performance, reliability, safety and product evolution) are important parameters of software products. Quality requirements of a system serve as a bridge between business goals and software architectures [3]. There is a major role of Software architecture in the determination of software quality [4][5].

Importance of software architecture “Software architecture is not only concerned with structure and behavior, but also with usage, functionality, performance, resilience, reuse, comprehensibility, economic and technology constraints and tradeoffs” - The Rational Unified Process, 2002.

Software Product Line Architectures design The main task of the software product line architecture design is to develop the reference architecture which represents the base structure Of the member products [6]. There are several methods have been established to create PLAs[7].

Safety-driven SPLA Design The current work in systems engineering methods has focused on supporting a safety-

centric design process [8]. The General idea of these approaches is that safety should be a driver for design.

Problem Description and Motivation Software architecture design is an important or a critical step of software development. The software architecture community generally believes that quality attributes (such as performance, usability, security, reliability and modifiability) of a software system are primarily achieved through attention to software architecture. This means that the design decisions embodied by software architecture are strongly influenced by the need to achieve quality attribute goals.

Nowadays, there are various design methods available and each is focusing on certain perspective of architecture design. The quality-based methods have received a lot of attentions and have been well developed for single system architecture design. However, the use of quality-based design methods is limited in software product line (SPL) because of the complexity and variabilities existing in SPL architecture [9].

There are increasing in the attention of software safety, so how to improve software safety has already become a more important concerned issue, especially for the safety-critical systems [10]. Currently, the influence of architecture in assurance of software safety is being increasingly recognized. The design of the safety at the architecture level can effectively improve software or system safety [10].

Consequently, the research activities are scattered across many research communities, system domains (such as embedded systems or information systems), and quality

attributes. Similar approaches are proposed in multiple domains without being aware of each other.

This study aims at surveying existing research on Software Product Line Architecture (SPLA) design based on quality attributes, and to give an overview of the intersection of the areas of software product line architecture design and Safety Driven Design in order to classifying existing work, and discover open issues for further research.

Research Approach and Contribution

To connect the knowledge and provide a comprehensive overview of the current state of the art, this article provides a systematic literature review of the existing research on Software Product Line Architecture (SPLA) design based on quality attributes in order to identify useful approaches and needs for future research. Also this study investigates safety analysis at the architectural level, and safety-based software Product Line Architecture Design (SSPLAD) approaches.

Also, the purpose of this investigation was to study and compare the existing methods or approaches for the design of software product line architectures. The intention of this paper is not to provide an exhaustive survey on the area but provide a state-of-the-art of current PLA practices and help others to understand and contrast alternative approaches to product line design. This paper does neither guide in selecting the right approach for PLA design but opens up a basis for creation of such a decision tool.

In general, with the survey we aim to achieve the following objectives:

- to give an overview of the intersection of the areas of software product line architecture design and Safety attribute.
- provide a basic classification framework in form of a taxonomy to classify existing architecture design approaches.
- provide an overview of the current state of the art in the product line architecture design domain.
- point out current trends, gaps, and directions for future research.

Organization. The rest of the paper is organized as follows. First, Section 2 presents the Overview of PLA design, The Quality-based Design, and Safety-driven design. The section 3 outlines the research method and the underlying protocol for the systematic literature review. The first contribution of this article, a taxonomy or architecture design approaches that has been derived from an iterative analysis of the existing research literature is presented in Section 4. The second contribution, a classification of existing architecture design approaches according to this taxonomy, is presented in Section 5. Finally, Section 6 identifies future research

directions based on the survey results and Section 7 presents the conclusions.

2. OVERVIEW

In this section we present an overview of the main concepts that are frequently relevant in the context of software architecture, architecture design, software PLA, quality attributes, and safety attribute.

2.1 PLA design

What is Software Product Line? Software product line is defined as “A set of software-intensive systems sharing a common managed set of features that satisfy the specific needs of a particular market segment or mission [11]”. These Systems are developed from a common core of assets (e.g. a common architecture) in a prescribed way.

Software Product Line (SPL) engineering is about developing a collection of systems which share great commonalities[3],[14],[15]. The idea of SPL was initiated by Parnas [16] and has been further developed by Kang et al [17]. The concept of SPL is to discover both commonalities and variabilities among member products of the product family.

(Liliana Dobrica, Eila Niemela,2003) [9].Product-line (PL) and reusable software components are suitable approaches for embedded systems, which are often re-engineered from existing systems. Important issues in the development and maintenance of these software systems are functionality and quality. Although there are some similarities between embedded systems regarding quality attributes, there are also differences. If a quality attribute is important to one product-line domain, it does not necessarily mean it is important to another one.

Developing a reference architecture which represent the base structure of the member products is the main task of the software product line architecture design [3].

The Software Product Line Architecture (SPLA) [3] provides a coarse grain picture of structure in the software product family. It initiates the architecture design for the member product. In the architecture design of a product line, it must accommodate the variability and dependency of functionality in the components that is derived from the feature model [3].

In the last two decades, software product line Architectures have been used successfully in industry for building families of systems of related products, maximizing reuse, and exploiting their variable and configurable options [10],[12],[13].

The creation and validation of product line software architectures are inherently more complex than those of software architectures for single systems.

Figure 1 illustrates examples of product line architecture for embedded system [18], which is the Product Line Architecture for a Microwave Oven.

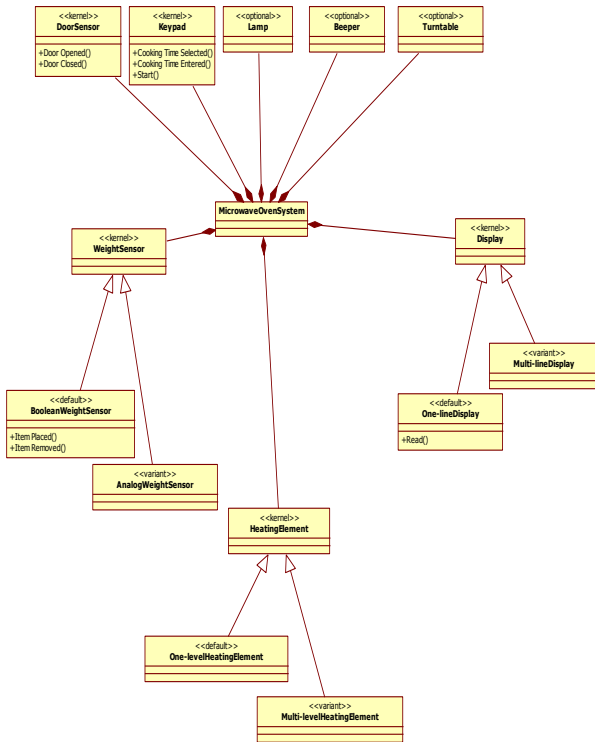


Figure1: Product Line Architecture for a Microwave

Definition of Terms Used in the Example Class Diagram:

Kernel: Kernel in product lines represents the mandatory features for the product line members. i.e.: they cannot be omitted in products.

- The stereotype <<kernel>> is used to specify Kernel in UML class diagrams.

Optional: Optionality in product lines means that some features are elective for the product line members, which means they can be omitted in some products and included in others.

- The stereotype <<optional>> is used to specify optionality in UML class diagrams.
- The optionality can concern classes, packages, attributes or operations. So the <<optional>> stereotype can be applied to Classifier, Package and Feature meta-classes.

Variant: Variant classes are modeled using UML inheritance and stereotypes. Each variation point will be defined by an abstract class and a set of subclasses.

- The abstract class will be defined with the stereotype <<variant>> and
- each subclass will be stereotyped <<variant>>, or <<optional>>, the default value being variant.

2.2 The Quality-based PLA Design

It is evident that there are several architecture design methods available for Software Product Line. Among all the current popular design methods, there are a few quality oriented architecture design method. In the next lines we briefly present and discuss some works.

(Len Bass et al., November 2001)[20] Their works is related to the quality attributes and design of software architecture. They presented an approach to characterizing quality attributes and capturing architectural patterns that are used to achieve these attributes. For each pattern, it is important not only how the pattern achieves a quality attribute goal but also what impact the pattern has on other attributes. They embodied this investigation of quality into the Attribute Driven Design Method for designing software architecture.

They have embarked on an effort to identify and codify architectural patterns that are primitive with respect to the achievement of quality attributes. They called this set of architectural patterns attribute primitives. They embodied this relationship in a design method for software architecture. Their work in brief, they have a characterization for six important attributes, they have a list and an organization for attribute primitives to achieve these attributes, and they have modified the Attribute Driven Design method (ADD) to utilize both the attribute characterizations and the attribute primitives.

However, this work concern with how to use or modify the architectural patterns in term of achievement of quality attributes at the architectural design level, No new method is develop with consideration of specific attribute.

(Bosch et al., 2000) [21] Presents a design method that elevates quality attributes from being almost totally ignored to an important player in the design process. This method, however, still places functional requirements as primary and quality requirements as secondary. The method begins with a design that achieves the functional requirements and this design is iteratively transformed into one that achieves quality requirements.

(Lei Tan, Yuqing Lin, Huilin Ye, 2012) [22]. Quality-driven Architecture Design and quality Analysis (QADA) is a traceable quality based method to design and evaluate software architecture. QADA contains scenario-based quality analysis to evaluate if the architecture design options meet the quality requirements. QADA consists of three viewpoints: structural view, behavior view, and deployment view at two levels of abstractions: conceptual level and concrete level. It contains several views at different levels to separate concerns and it provides a quality-driven link between software requirement and architecture. This work extended QADA method by adding an extra view to improve this quality based PLA design method.

In this framework, the quality attributes of a software system will be taken into account in the early stage of architecture design and the reference architecture of SPL will be elicited based on quality-related consideration.

However, their work is just extending to QADA method by adding an extra view to improve this quality based PLA design method. This work may be a direction of more open researches, especially in field of product line, that by focusing on a specific quality attribute or other architectural attributes.

2.3 The safety-based design

Software safety assurance refers to a series of quality assurance activities during software development life cycle, which aims to eliminate the potential dangers.

The specification of safety constraints is the first step of the safety-constraint centered design approach [29].

2.3.1 The design and safety

While modeling software safety it is important to note that no software works in isolation. The entire system must be designed to be safe. The system components may be software, hardware, users, and the environment. All must be given consideration when developing software. All parts of the system must be safe. Functional and operational safety starts at the system level. Safety cannot be assured if efforts are focused only on software. The software can be totally free of 'bugs' and employ numerous safety features, yet the equipment can be unsafe because of how the software and all the other parts interact in the system [27].

2.3.2 Safety analysis at the architectural level

“From a safety viewpoint, the software architecture is where the basic safety strategy is developed in the software.” It is very significant to study how the non-functional attribute “safety” to be described, analyzed and verified during the architecture construction process [10].

Although a considerable number of safety analysis techniques have been proposed to aid software design such as Software Hazard Analysis and Resolution in Design (SHARD) [28], there is little analysis work focusing on an architectural level to aid software architecture design. In particular, safety is the entire property of a system; it is almost impossible to analyze software safety effectively without considering system or platform safety. We thus need a safety analysis approach that is able to model the integration of software with hardware or other system components and to characterise the architectural elements at different architectural levels.

2.3.3 Safety attribut and Product line Architecture design

As product-line engineering becomes more widespread, more safety-critical software product lines are being built [23].

The study shown that, Nowadays, there are various design methods available and each is focusing on certain perspective of architecture design. Especially, safety-based methods have received a lot of attentions and have been well developed for single system architecture design. However, the use of safety-based design methods is limited in software product line (SPL) because of the complexity and variabilities existing in SPL architecture. In the next lines we briefly present and discuss some of the related works.

(Donald Firesmith, 2004) [24]. His work concerned with the Engineering Safety Requirements, Safety Constraints, and Safety-Critical Requirements. He used the concept of a quality model to define safety as a quality factor. Thus, safety (like security and survivability) is a kind of defensibility, which is a kind of dependability, which is a kind of quality. Next, He discussed the structure of quality requirements and showed how safety requirements can be engineered based on safety's numerous quality subfactors. Then, He defined and discussed safety constraints (i.e., mandated safeguards) and safety-critical requirements (i.e., functional, data, and interface requirements that can cause accidents if not implemented correctly).

However, no tasks or attentions related to how design the software product line architecture based on safety analysis.

(David C. Jensen, Irem Y. Tumer, 2013) [25]. their work presented a method of explicit inclusion of safety into a model-based design for cyber physical systems. This approach enables an analysis where component-level failures can be mapped to potential system-level hazards. This work presented a method of representing the safety property of a system by the introduction of the concept termed "safety function". Further, the function of achieving safety is mapped to the performance functions of the system. They presented a process of concurrently developing a system concept from the safety and functional perspective. The end result of this process is a system architecture where components of the system are explicitly mapped to both the functions they perform and the role they play in ensuring safe system operation. The benefit of this approach is having a system representation that allows for analysis of critical events and off-nominal component behavior to identify potential losses in function and safety constraint violations. The perspective of these approaches is that safety should be a driver for design. Thus the objective of this work is to introduce a safety-centric method of developing a design based on the functional modeling paradigm.

However, this work does not address the design of software architecture. The proposed method focus on inclusion of safety into design level in general without focusing on a specific design activity.

(Yuling Huang, 2013) [26]. Safety design at the architecture level can effectively improve software or system safety. This work address the problem of how to consider safety in software architecture design phase and proposed a safety-oriented software architecture design approach. Through the system hazard analysis, this design approach uses the selected combination of safety tactics to effectively improve the software or system safety, providing a new way of thinking for software safety architecture design.

However, this work does not take in account the concepts of a family of architectures, namely, Product line Architectures.

3. RESEARCH METHOD

A survey is a method for collecting data about features, behavior or opinions of a specific group of people, pointed out as representative from a target population (Pinsonneault and Kraemer, 1993).

The stages involved in our literature review are structured into three phases: planning, conducting, and reporting the review, based on the guidelines proposed by Kitchenham [30].

A systematic mapping study is launched to find as much literature as possible, and the 22 papers found are classified with respect to focus, research type and contribution type.

Based on the guidelines, Kitchenham [30], this section details the research questions, the performed research steps, and the protocol of the literature review. First, Section 2.1 describes the research questions underlying our survey. Then, Section 2.2 derives the research tasks we conducted, and thus describes our procedure. Section 2.3 then details the literature search step and highlights the inclusion and exclusion criteria. Finally, Section 2.4 discusses threats to the validity of our study.

However, the reported results are fragmented over different research communities, multiple system domains, and multiple

quality attributes. Based on this survey, a taxonomy has been created which is used to classify the existing research. Furthermore, the systematic analysis of the research literature provided in this review aims to help the research community in consolidating the existing research efforts and deriving a research agenda for future developments.

3.1 Research Questions

Based on the objectives described in the introduction, the following research questions have been derived, which form the basis for the literature review:

- RQ1 How can the current research on software architecture design be classified?
- RQ2 What is the current state of the art of software architecture design research with respect to this classification? And the SPLA design, Quality-driven SPLA design, Safety-driven SPLA design methods in the existing methods?
- RQ3 What can be learned from the current research results that will lead to topics for further investigation?

3.2 Research Tasks

To answer the three research questions RQ1-3, numbers of tasks have been conducted: one task to set up the literature review, and others research tasks dedicated to the identified research questions.

3.3 Literature Search Process

The search strategy for the review was primarily directed toward finding published papers in journals and conference proceedings via the widely accepted literature search engines and databases Google Scholar, IEEE Explore, and Elsevier ScienceDirect. For the search we focused on selected keywords, based on the aimed scope of the literature review. Examples of the keywords are: Software Architecture, Quality attributes, Safety analysis, Architectural Design, Software Product Line Architectures, Safety-driven software product line architecture design.

The keywords were refined and extended during the search process. In the subsequent phase, we reviewed the abstracts (and keywords) of the collected papers with respect to the defined set of inclusion and exclusion criteria (Sections 2.3.1 and 2.3.2 below), and further extended the collection with additional papers based on an analysis of the cited papers and the ones citing it (forward and backward citation search).

3.3.1 Inclusion Criteria

The focus of this literature review is on software architecture quality attributes, safety analysis, architectural design, software product line architectures, and safety-driven software product line architecture design. A summary of the inclusion and criteria is: Peer reviewed publications with a clear focus on some aspect of software product line architecture design.

3.3.2 Exclusion Criteria

We excluded papers that: (a) design a software with no relation to software architecture, (b) focus on an architecture-irrelevant problem, (c) focus on software architecture design for single program without considering any quality attribute, (d) focus on a product line-irrelevant problem.

We did not exclude papers for quality reasons, because the quality of the papers was generally acceptable. A summary of the exclusion criteria is: Publications where either architecture design focus or software product line focus is lacking.

3.4 Threats to Validity

One of the main threats to the validity of this literature review is the incompleteness. The risk of this threat highly depends on the selected list of keywords and the limitations of the employed search engines. To decrease the risk of an incomplete keyword list, we have used an iterative approach to keyword-list construction. A well-known set of papers was used to build the initial taxonomy which evolved over time. New keywords were added when the keyword list was not able to find the state-of-the-art in the respective area of study. Another important issue is whether our taxonomy is robust enough for the analysis and classification of the papers. To avoid a taxonomy with insufficient capability to classify the selected papers, we used an iterative content analysis method to continuously evolve the taxonomy for every new concept encountered in the papers. New concepts were introduced into the taxonomy and changes were made in the related taxonomy categories. Furthermore, in order to make the taxonomy a better foundation for analyzing the selected papers, we allowed multiple abstraction levels for selected taxonomy concepts.

4. TAXONOMY AND CLASSIFICATION

The quality of a literature review project highly depends on the selected taxonomy scheme, which influences the depth of knowledge recorded about each studied approach [31]. This section, present the identification of the taxonomy categories and provide an answer to the first research question (RQ1).

In this article we provided a basic classification framework in form of taxonomy to classify existing architecture design approaches. As mentioned in the previous sections, here we present two schemes of broad classifications of software architecture design approaches, the two sections bellow are illustrate that.

4.1 First classification scheme

As the first step of our survey, we classify existing approaches for software architecture design into three broad categories depending on whether they attempt to address the architecture of single product or product line or quality attributes of architecture.

Each of these categories contains one or more subcategories based on the high-level strategies used to realize its goal. Some of these sub-categories are further divided indicating the specific intention adopted. Fig.2 and Fig.3 illustrates this classification framework through which the results of the survey are presented.

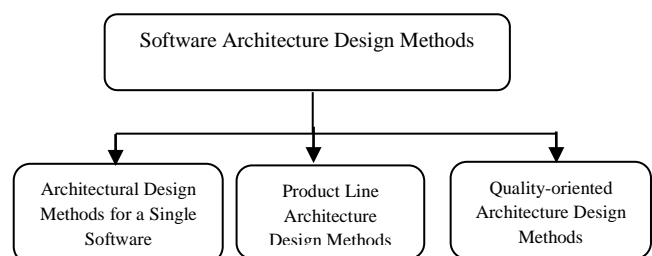


Figure2: High level taxonomy of Architecture Design Approaches

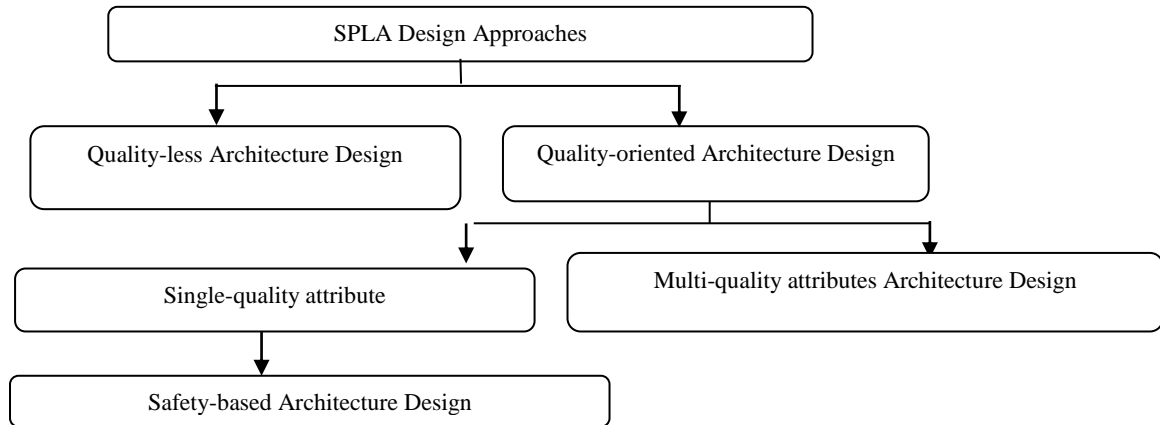


Figure 3. the taxonomy Related to SPLA Design Approaches

The papers are published between 1998 and 2014, and summarized in Table 1. Tables 2 and 3. The total number of classification items in Table 1 is 28.

Table 1 lists all papers on term of architecture design approaches. Table 2 lists all papers on quality-oriented architecture design approaches. Table 3 lists all papers on software product line architecture design approaches.

4.2 The second classification scheme in term of SPLA design

Here, the publications are classified into categories in three different dimensions: research focus, type of contribution and research type. This structure is presented by Petersen et al.

[52], [53]. However we adopt different categories in our study. We established a scheme and mapped publications iteratively and added them as new primary studies. When the scheme was finally set, we reviewed all classifications again.

We identified Three categories of research focus: (i) SPLA design, (ii) quality-based SPLA design, (iii) safety-based SPLA design, Contribution type is classified into five categories: Tool, Method, Model, Metric, and Open Items.

The classification of research types is based on a scheme proposed by Wieringa et al. [53] [54]. And we classified the research into six categories: (i) validation research, (ii) evaluation research, (iii) solution proposals, (iv) conceptual proposals, (v) opinion papers, and (vi) experience papers.

Table 1.Papers on architecture design approaches (high level taxonomy/a broad classification)

No	Authors [Ref]	Paper Title	year	Architectural Design for a Single Software	Product Line Architecture Design	Quality-oriented Architecture Design
1	David C. Jensen, Irem Y. Tumerb[25]	Modeling and Analysis of Safety in Early Design	2013	✓		✓
2	Lei Tan, Yuqing Lin and Huilin Ye [3]	Modeling Quality Attributes in Software Product Line Architecture	2012		✓	✓
3	Yuling Huang [26]	Safety-Oriented Software Architecture Design Approach	2013	✓		✓
4	Len Bass, Mark Klein, and Felix Bachmann [20]	Quality Attribute Design Primitives and the Attribute Driven Design Method	2001	✓	✓	✓
5	Made Murwantara Tangerang, Indonesia [19]	Hybrid ANP: Quality Attributes Decision Modeling of a Product Line Architecture Design	2012		✓	✓
6	Qian Feng, Robyn R. Lutz [23]	Bi-Directional Safety Analysis of Product Lines	2005		✓	✓
7	Joachim Bayer, Oliver Flege, and Cristina Gacek [32]	Creating Product Line Architectures	2000		✓	
8	Lei Tan, Yuqing Lin, Huilin Ye [22]	Quality-Oriented Software Product Line Architecture Design	2012		✓	✓
9	Weihang Wu, Tim Kelly [33]	Safety Tactics for Software Architecture Design	2004	✓		✓
10	Liliana Dobrica, EILA Niemela [9]	Attribute-based product-line architecture development for embedded systems	2003		✓	✓
11	Bass, L.; Clements, P.; & Kazman, R. [34]	Software Architecture in Practice. Reading , Attribute Driven Design method (ADD)	2003	✓		✓
12	John Ryan O'Farrell [35]	Development of A Software Architecture Method for Software Product Families and its Application to the AubieSat Satellite Program	2009		✓	
13	J'urgen Meister, Ralf Reussner, Martin Rohde [36]	Applying Patterns to Develop a Product Line Architecture for Statistical Analysis Software	2004		✓	
14	P. America, H. Obbink, J. Muller, and R. van Ommering [37]	COPA: A Component-Oriented Platform Architecting Method for Families of Software Intensive Electronic Products	2000		✓	
15	D. Weiss, C. Lai, and R. Tau [38]	a family-based software development process.	1999		✓	
16	K. C. Kang, S. Kim, J. Lee, K. Kim, E. Shin, and M. Huh [39]	FORM: A Feature-Oriented Reuse Method with Domain- Specific Reference Architectures	1998		✓	
17	C. Atkinson et al [40]	Component-based product line engineering with UML	2002		✓	
18	Mikael Svahnberg, Claes Wohlin, Lars Lundberg, Michael Mattsson [41]	A Quality-Driven Decision-Support Method for Identifying Software Architecture Candidates	2003	✓		✓
19	M. Matinlasi, E. Niemelä, and L. Dobrica[42]	Quality-driven architecture design and quality analysis method	2002		✓	✓
20	F. Bachmann, L. Bass, G. Chastek, P. Donohoe, and F. Peruzzi [43]	The Architecture Based Design Method	2000	✓	✓	
21	Hassan Gomaa[44]	Designing Software Product Lines with UML 2.0: From Use Cases to Pattern-Based Software Architectures	2006		✓	
22	Jianli Dong, Jianzhou Wang, Donghuai Sun, Haiyan Lu [45]	The Research of Software Product Line Engineering Process and Its Integrated Development Environment Model	2008		✓	
23	Jiayi Zhu, Xin Peng, Stan Jarzabek, Zhenchang Xing, Yinxing Xue, Wenyun Zhao [46]	Improving Product Line Architecture Design and Customization by Raising the Level of Variability Modeling	2011		✓	
24	M.Sharafi, S.Dadollahi [47]	A Scenario-Based Approach for Architecture Reconstruction of Product Line	2013		✓	
25	Hataichanok Unphon [48]	Making Use of Architecture throughout the Software Life Cycle – How the Build Hierarchy can Facilitate Product Line Development	2009		✓	

26	Jing Liu, Josh Dehlinger, Robyn Lutz [49]	Safety analysis of software product lines using state-based modeling	2007		✓	✓
27	Jan Bosch [8]	Software Product Lines and Software Architecture Design	2001		✓	✓
28	Thelma Elita Colanzi, Silvia Regina Vergilio[50]	Representation of Software Product Line Architectures for Search-Based Design	2013		✓	
29	Broerse, C., Riva, C., Gall, H., Wijnstra, J.g., Girard, J.F., Knodel, J., Pinzger, M., Pasman, W. [58]	Architecture Recovery for Product Family	2004		✓	

Table 2.Papers on quality-oriented architecture design approaches

No	Authors [Ref]	Paper	year	Single/Product line
1	M. Matinlassi, E. Niemel, and L. Dobrica[42]	Quality-driven architecture design and quality analysis method	2002	Product line
2	L. Bass, M. Klein, and F. Bachmann [51]	Quality Attribute Primitives and the Attribute Driven Design Method	2002	Support all
3	David C. Jensen, Irem Y. Tumerb[25]	Modeling and Analysis of Safety in Early Design	2013	Single
4	Lei Tan, Yuqing Lin and Huilin Ye [3]	Modeling Quality Attributes in Software Product Line Architecture	2012	Product line
5	Yuling Huang [26]	Safety-Oriented Software Architecture Design Approach	2013	Single
6	Len Bass, Mark Klein, and Felix Bachmann [20]	Quality Attribute Design Primitives and the Attribute Driven Design Method	2001	Single
7	Made Murwantara Tangerang, Indonesia [19]	Hybrid ANP: Quality Attributes Decision Modeling of a Product Line Architecture Design	2012	Product line
8	Qian Feng, Robyn R. Lutz [23]	Bi-Directional Safety Analysis of Product Lines	2005	Product line
9	Lei Tan, Yuqing Lin, Huilin Ye [22]	Quality-Oriented Software Product Line Architecture Design	2012	Product line
10	Weihang Wu, Tim Kelly [33]	Safety Tactics for Software Architecture Design	2004	Single
11	LILIANA DOBRICA, EILA NIEMELÄ [9]	Attribute-based product-line architecture development for embedded systems	2003	Product line
12	Mikael Svahnberg, Claes Wohlin, Lars Lundberg, Michael Mattsson [41]	A Quality-Driven Decision-Support Method for Identifying Software Architecture Candidates	2003	Single
13	Jing Liu, Josh Dehlinger, Robyn Lutz [49]	Safety analysis of software product lines using state-based modeling	2007	Product line
14	Jan Bosch [8]	Software Product Lines and Software Architecture Design	2001	Product line

Table 3.Papers on software product line architecture design approaches

No	Authors [Ref]	Paper Title	Date	Quality-less Architecture Design	Quality-oriented Architecture Design	Single-quality attribute	Multi-quality attributes Architecture Design
1	P. America, H. Obbink, J. Muller, and R. van Ommering [37]	COPA: A Component-Oriented Platform Architecting Method for Families of Software Intensive Electronic Products	2000	✓			
2	D. Weiss, C. Lai, and R. Tau [38]	Software product-line engineering: a family-based software development process.	1999	✓			
3	K. C. Kang, S. Kim, J. Lee, K. Kim, E. Shin, and M. Huh [39]	FORM: A Feature-Oriented Reuse Method with Domain- Specific Reference Architectures	1998	✓			
4	C. Atkinson et al. [40]	Component-based product line engineering with UML	2002	✓			
5	Lei Tan, Yuqing Lin and Huilin Ye [3]	Modeling Quality Attributes in Software Product Line Architecture	2012		✓		✓
6	Len Bass, Mark Klein, and Felix Bachmann [20]	Quality Attribute Design Primitives and the Attribute Driven Design Method	2001		✓		✓
7	Made Murwantara Tangerang, Indonesia [19]	Hybrid ANP: Quality Attributes Decision Modeling of a Product Line Architecture Design	2012		✓		✓
8	Qian Feng, Robyn R. Lutz [23]	Bi-Directional Safety Analysis of Product Lines	2005		✓	✓	
9	Joachim Bayer, Oliver Flege, and Cristina Gaek [32]	Creating Product Line Architectures	2000	✓			
10	Lei Tan, Yuqing Lin, Huilin Ye [22]	Quality-Oriented Software Product Line Architecture Design	2012		✓		✓
11	Liliana Dobrica, Eila NIEMELÄ [9]	Attribute-based product-line architecture development for embedded systems	2003		✓		✓
12	John Ryan O'Farrell [35]	Development of A Software Architecture Method for Software Product Families and its Application to the AubieSat Satellite Program	2009	✓			
13	J'urgen Meister, Ralf Reussner, Martin Rohde [36]	Applying Patterns to Develop a Product Line Architecture for Statistical Analysis Software	2004	✓			
14	Hassan Gomaa [44]	Designing Software Product Lines with UML 2.0: From Use Cases to Pattern-Based Software Architectures	2006	✓			
15	Jianli Dong, Jianzhou Wang, Donghuai Sun, Haiyan Lu [45]	The Research of Software Product Line Engineering Process and Its Integrated Development Environment Model	2008	✓			
16	Jiayi Zhu, Xin Peng, Stan Jarzabek, Zhenchang Xing, Yinxing Xue, Wenyun Zhao [46]	" Improving Product Line Architecture Design and Customization by Raising the Level of Variability Modeling	2011	✓			
17	M.Sharafi, S.Dadollahi [47]	A Scenario-Based Approach for Architecture Reconstruction of Product Line	2013	✓			
18	Hataichanok Unphon [48]	Making Use of Architecture throughout the Software Life Cycle – How the Build Hierarchy can Facilitate Product Line Development	2009	✓			
19	M. Matinlassi, E. Niemelä, and L. Dobrica [42]	Quality-driven architecture design and quality analysis method	2002		✓		✓
20	F. Bachmann, L. Bass, G. Chastek, P. Donohoe, and F. Peruzzi [43]	The Architecture Based Design Method	2000	✓			
21	Jing Liu, Josh Dehlinger, Robyn Lutz [49]	Safety analysis of software product lines using state-based modeling	2007		✓	✓	
22	Jan Bosch [8]	Software Product Lines and Software Architecture Design	2001		✓		✓
23	Thelma Elita Colanzi, Silvia Regina Vergilio [50]	Representation of Software Product Line Architectures for Search-Based Design	2013	✓			
24	Broerse, C., Riva, C., Gall, H., Wijnstra, J.g., Girard, J.F., Knodel, J., Pinzger, M., Pasman, W. [58]	Architecture Recovery for Product Family	2004	✓			

Table 4. Distribution over research focus

Research focus	1998-2008	2009-2014	Total
SPLA design	10	5	15
Quality-based SPLA design	4	3	7
Safety-based SPLA design	2	-	2
Total	16	8	24

Table 5. Papers on SPLA design.

Authors [Ref]	Title	Paper Type	Contribution type
John Ryan O'Farrell [35]	Development of A Software Architecture Method for Software Product Families and its Application to the AubeSat Satellite Program	Conceptual proposal	Method
Joachim Bayer, Oliver Flege, and Cristina Gacek [32]	Creating Product Line Architectures	Solution proposal	Method
Jürgen Meister, Ralf Reussner, Martin Rohde [36]	Applying Patterns to Develop a Product Line Architecture for Statistical Analysis Software	Experience report	Tool
Hassan Gomaa [44]	Designing Software Product Lines with UML 2.0: From Use Cases to Pattern-Based Software Architectures	Solution proposal	Model
Jianli Dong, Jianzhou Wang, Donghuai Sun, Haiyan Lu [45]	The Research of Software Product Line Engineering Process and Its Integrated Development Environment Model	Opinion paper	Model
Jiayi Zhu, Xin Peng, Stan Jarzabek, Zhenchang Xing, Yinxing Xue, Wenyun Zhao [46]	Improving Product Line Architecture Design and Customization by Raising the Level of Variability Modeling	Conceptual proposal	Open items
M.Sharafi, S.Dadollahi [47]	A Scenario-Based Approach for Architecture Reconstruction of Product Line	Conceptual proposal	Method
Hataichanok Unphon [48]	Making Use of Architecture throughout the Software Life Cycle – How the Build Hierarchy can Facilitate Product Line Development	Opinion paper	Open items
P. America, H. Obbink, J. Muller, and R. van Ommering [37]	COPA: A Component-Oriented Platform Architecting Method for Families of Software Intensive Electronic Products	Conceptual proposal	Model
D. Weiss, C. Lai, and R. Tau [38]	Software product-line engineering: a family-based software development process.	Conceptual proposal	Open items
K. C. Kang, S. Kim, J. Lee, K. Kim, E. Shin, and M. Huh [39]	FORM: A Feature-Oriented Reuse Method with Domain- Specific Reference Architectures	Conceptual proposal	Model
C. Atkinson et al. [40]	Component-based product line engineering with UML	Conceptual proposal	Tool
F. Bachmann, L. Bass, G. Chastek, P. Donohoe, and F. Peruzzi [43]	The Architecture Based Design Method	Solution proposal	Model
Thelma Elita Colanzi, Silvia Regina Vergilio [50]	Representation of Software Product Line Architectures for Search-Based Design	Experience report	Open items
Broerse, C., Riva, C., Gall, H., Wijnstra, J.g., Girard, J.F., Knodel, J., Pinzger, M., Pasman, W. [58]	Architecture Recovery for Product Family	Solution proposal	Method

Table 6. Papers on Quality-based SPLA design.

Authors [Ref]	Title	Paper Type	Contribution Type
Lei Tan, Yuqing Lin and Huilin Ye [3]	Modeling Quality Attributes in Software Product Line Architecture	Conceptual proposal	Method
I Made Murwantara Tangerang, Indonesia [19]	Hybrid ANP: Quality Attributes Decision Modeling of a Product Line Architecture Design	Conceptual proposal	Model
Lei Tan, Yuqing Lin, Huilin Ye [22]	Quality-Oriented Software Product Line Architecture Design	Solution proposal	Method
Liliana Dobrica, Eila Niemela [9]	Attribute-based Product-line Architecture Development for Embedded Systems	Solution proposal	Method
Len Bass, Mark Klein, and Felix Bachmann [20]	Quality Attribute Design Primitives and the Attribute Driven Design Method	Opinion paper	Open items
M. Matinlassi, E. Niemel, and L. Dobrica [42]	Quality-driven architecture design and quality analysis method	Conceptual proposal	Method
Jan Bosch [8]	Software Product Lines and Software Architecture Design	Experience report	Method

Table 7. Papers on Safety-based SPLA design.

Authors [Ref]	Title	Paper Type	Contribution type
Qian Feng, Robyn R. Lutz [23]	Bi-Directional Safety Analysis of Product Lines	Conceptual proposal	Method
Jing Liu, Josh Dehlinger, Robyn Lutz [49]	Safety analysis of software product lines using state-based modeling	Solution proposal	Method

5. MAPPING

In Fig. 4, we show, based on the second classification in section 4.2, a mapping between the research focus and the contribution type and the research type. The research focus items include SPLA design, Quality-based SPLA design, and Safety-based SPLA design. The contributions types include

tools, methods, models, and metrics. The research types include experience reports, opinion papers, conceptual or solution proposals, and validation and evaluation research. The Research focus is on the Y axis, the contribution type is on the left side of the X axis, and research type on the right side of the X axis.

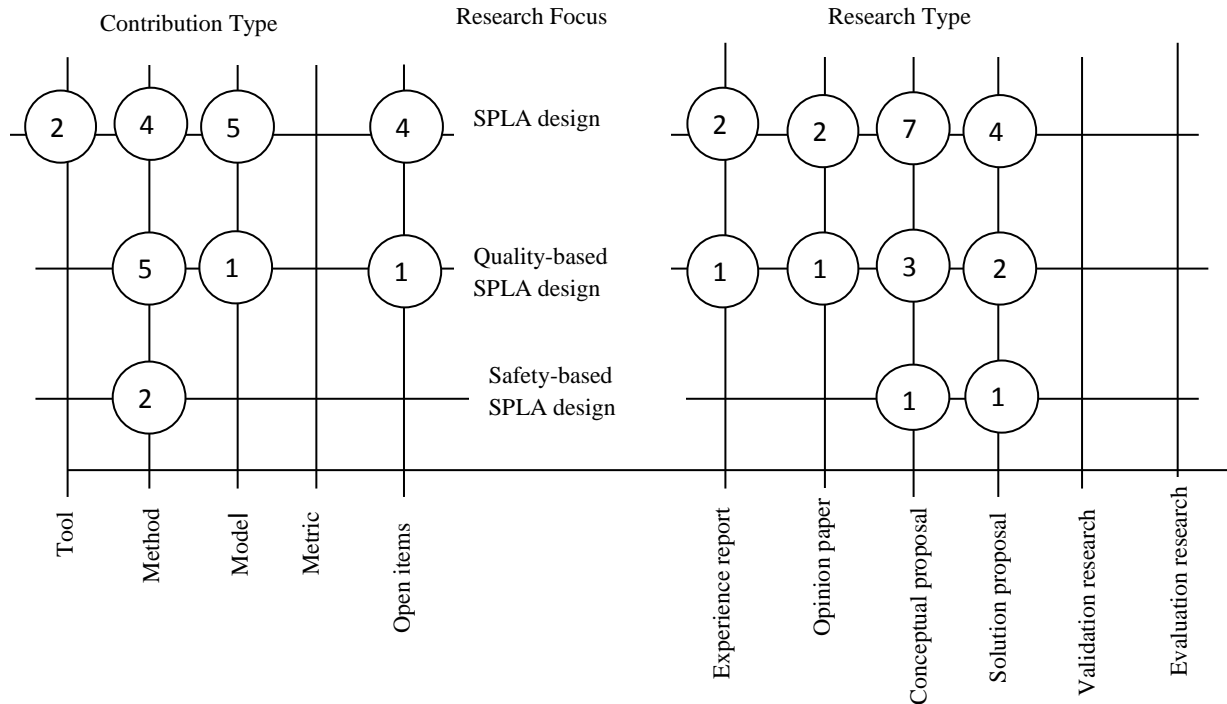


Figure 4. Map of research focus on software product line design. Research focus on the Y axis; contribution type on the left side of the X axis, and research type on the right side of the X axis.

6. Discussion and Results

The surveyed research work indicates safety based software product line design being rather an immature area.

Safety-driven software product line architecture design seems to be a “discussion” topic. There is a well established understanding about challenges. However, when looking for solutions to these challenges, we mostly find proposals. The mapping shows that 74% of the papers found include proposals, which contain ideas for solutions of the identified challenges.

The study shows that there are various design methods available and each is focusing on certain perspective of architecture design and there is a large number of SPLA design methods. Especially, quality-based methods have received a lot of attention. However, the use of safety-based design methods is limited in software product lines (SPL) due to the variability property that can potentially result in a large number of possible systems.

7. CONCLUSION AND FUTURE WORKS

Because a product line reference software architecture is the central artifact in product line engineering which provides the framework for developing and integrating shared assets [47], the creation and validation of such architecture are inherently much more complex. This study aims at surveying existing research on Software Product Line Architecture (SPLA) based on quality attributes in order to identify useful approaches and needs for future research. We investigated safety analysis at

the architectural level, and Safety-driven Software Product Line Architecture Design (SSPLAD) approaches.

The paper show that there are various design methods available and each is focusing on certain perspective of architecture design. The quality-based methods have received a lot of attentions and have been well developed for single system architecture design. However, the use of quality-based design methods is limited in software product line (SPL) because of the complexity and variability existing in the SPL reference architecture [22]. With the increasing attention to software safety, improving software safety has already become a much more important issue, especially for safety-critical systems [26]. We identify the following open issues, open research issues as related to specific key publications surveyed in this paper.

- The methodology presented by Jensen and Tumer [25] focuses on the inclusion of safety into design level in general without focusing on a specific design activity such as the design of reference architectures.
- In [22], Lei Tan et al, presented a framework for Quality-Oriented Software Product Line Architecture Design. The framework is defined at a high level without specifying any modeling techniques or tools.
- The work of Huang in [26] provides a rather complex process for safety-oriented design using Fault Tree Analysis techniques and safety tactics

that would be difficult or intractable to use with the concept of variability in product lines where the space of possible faults is very large. It remains an open issue to research if the FTA techniques combined with safety tactics can be used with fault classification techniques for safety-driven design of reference architectures.

8. ACKNOWLEDGMENTS

This work was made possible by NPRP grant # [7-662-2-247] from Qatar Research Fund (a member of Qatar Foundation). The findings achieved herein are solely the responsibility of the authors. We also thank "Department of Computer Science, Sudan University of Science and Technology" for its support which helped us to complete this paper successfully.

9. REFERENCES

- [1] L. Bass, P. Clements and R. Kazman, *Software Architecture in Practice*, 2nd ed. Addison-Wesley, 2003.
- [2] IEEE-Std-1471-2000. IEEE recommended practice for architectural description of software-intensive systems. 1471-2000.
- [3] Lei Tan, Yuqing Lin and Huilin Ye, *Modeling Quality Attributes in Software Product Line Architecture*, 978-1-4577-1964-6/12/\$31.00 IEEE, 2012.
- [4] Nenad Medvidovic, Sam Malek, Marija Mikic-Rakic, *Software Architectures and Embedded Systems*, 2003.
- [5] Peter Wallin , Stig Larsson, Joakim Fröberg, Jakob Axelsson, *Problems and their mitigation in system and software architecting*, *Information and Software Technology* 54 (2012) 686–700, 2012.
- [6] Emelie Engström, Per Runeson, *Software product line testing – A systematic mapping study*, *Information and Software Technology* 53 (2011) 2–13, 2010.
- [7] Mari Matinlassi, *Comparison of Software Product Line Architecture Design Methods: COPA, FAST, FORM, KobrA and QADA*, in 26th International Conference on Software Engineering (ICSE'04) 0270-5257/04 \$20.00, IEEE, 2001.
- [8] Jan Bosch, *Software Product Lines and Software Architecture Design*, *Proceedings of the 23rd International Conference on Software Engineering (ICSE'01)* 0270-5257/01 2001.
- [9] Liliana Dobrica, Eila Niemea, *Attribute-Based Product-Line Architecture Development for Embedded Systems*, 2003.
- [10] Rafael Capilla, Jan Bosch, Pablo Trinidad, Antonio Ruiz-Cortés, Mike Hinchey, *An overview of Dynamic Software Product Line architectures and techniques: Observations from research and industry*, *The Journal of Systems and Software* 91 (2014) 3–23, 2013.
- [11] Pär J Ågerfalk, Brian Fitzgerald, Brian Lings, Björn Lundell, Liam O'Brien, and Steffen Thiel, *Open Source in the Software Product Line: An Inevitable Trajectory*, 2006.
- [12] Razieh Behjati, Tao Yue, Lionel Briand, Bran Selic, *SimPL: A product-line modeling methodology for families of integrated control systems*, *Information and Software Technology* 55 (2013) 607–629, 2013.
- [13] Jules White, José A. Galindo, Tripti Saxena, Brian Dougherty ,David Benavidesb, Douglas C. Schmid, *Evolving feature model configurations in software product lines*, *The Journal of Systems and Software* 87 (2014) 119– 136, 2013.
- [14] J. Bosch, *Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach*, Addison-Wesley, 2000.
- [15] Andreas Pleuss, Goetz Botterweck, Deepak Dhungana, Andreas Polzer, Stefan Kowalewski, *Model-driven support for product line evolution on feature level*, *The Journal of Systems and Software* 85 (2012) 2261– 2274, 2011.
- [16] D. Parnas, *On the Design and Development of Program Families*, *IEEE Transactions on Software Engineering*, vol. 2, 1-9, 1976.
- [17] K. C. Kang, S. Kim, J. Lee, K. Kim, E. Shin and M. Huh, *FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures*, *Annals of Software Engineering*, vol. 5, 143-168, 1998.
- [18] Prof Hany H. Ammar, *Lectures :The Software Product Line Architectures*, PhD Graduate Program, Sudan University of Science and Technology, Khartoum, Sudan, 2013.
- [19] I Made Murwantara, *Hybrid ANP: Quality Attributes Decision Modeling of a Product Line Architecture Design*, 2012 International Conference on Uncertainty Reasoning and Knowledge Engineering, 2012.
- [20] Len Bass, Mark Klein, and Felix Bachmann, *Quality Attribute Design Primitives and the Attribute Driven Design Method*, Software Engineering Institute Carnegie Mellon University Pittsburgh, Pa 15213 USA, supported by the U.S. Department of Defense and Robert Bosch, GmbH 2001.
- [21] Bosch, J. *Design & Use of Software Architectures*, Addison Wesley, 2000.
- [22] Lei Tan, Yuqing Lin, Huilin Ye, *Quality-Oriented Software Product Line Architecture Design*, *Journal of Software Engineering and Applications*, 5, 472-476, , 2012.
- [23] Qian Feng, Robyn R. Lutz, *Bi-Directional Safety Analysis of Product Lines*, JSSF, 2005.
- [24] Donald Firesmith, *Engineering Safety Requirements, Safety Constraints, and Safety-Critical Requirements*, *JOURNAL OF OBJECT TECHNOLOGY JOURNAL OF OBJECT TECHNOLOGY*, Vol. 3, No. 3, 2004.
- [25] David C. Jensen, Irem Y. Tumer, *Modeling and Analysis of Safety in Early Design*, *Conference on Systems Engineering Research (CSER' 13)*, Eds.: C.J.J. Paredis, C. Bishop, D. Bodner, Georgia Institute of Technology, Atlanta, GA, March 19-22, 2013.
- [26] Yuling Huang, *Safety-Oriented Software Architecture Design Approach*, *International Conference on Information Science and Computer Applications (ISCA 2013)*, Published by Atlantis Press, 2013.
- [27] M. Ben Swarup and P. Seetha Ramaiah, *A Software Safety Model for Safety Critical Applications*, *International Journal of Software Engineering and Its Applications* Vol. 3, No.4, October 2009.

- [28] Pumfrey, D. J., Fenelon, P., McDermid, J.A. & Nicholson, M., Towards Integrated Safety Analysis and Design, ACM Computing Reviews, Vol. 2, No. 1, pp. 21-32, 1994.
- [29] Ramakrishna, Satish et. al., “Run time Assertion Schemes for Safety Critical Systems”, Ninth IEEE Symposium on Computer Based Medical Systems, Ann Arbor, Michigan, 1996.
- [30] B. Kitchenham, Procedures for performing systematic reviews, Keele University, Department of Computer Science, Keele University, UK, Technical Report TR/SE-0401, 2004.
- [31] Aldeida Aleti, Barbora Buhnova, Lars Grunske, Member, IEEE, Anne Koziolk, Member, IEEE, and Indika Meedeniya , Software Architecture Optimization Methods: A Systematic Literature Review, 2012.
- [32] Joachim Bayer, Oliver Flege, and Cristina Gacek, Creating Product Line Architectures, Fraunhofer Institute for Experimental Software Engineering (IESE) Sauerwiesen 6, D-67661 Kaiserslautern, Germany, 2000.
- [33] Weihang, Wu Tim Kelly, Safety Tactics for Software Architecture Design, Department of Computer Science, University of York, York YO10 5DD, UK {weihang.wu, tim.kelly}@cs.york.ac.uk, 2004.
- [34] Bass, L.; Clements, P.; & Kazman, R. Software Architecture in Practice. Reading, MA: Addison-Wesley, 2003.
- [35] John Ryan O’Farrell, Development of A Software Architecture Method for Software Product Families and its Application to the AubieSat Satellite Program, A thesis submitted to the Graduate Faculty of Auburn University, in partial fulfillment of the requirements for the Degree of Master of Science Auburn, Alabama, 2009.
- [36] J’urgen Meister, Ralf Reussner, Martin Rohde, Applying Patterns to Develop a Product Line Architecture for Statistical Analysis Software, Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA’04) 0-7695-2172-X/04 \$ 20.00 © 2004 IEEE, 2004.
- [37] P. America, H. Obbink, J. Muller, and R. van Ommering, COPA: A Component-Oriented Platform Architecting Method for Families of Software Intensive Electronic Products, Denver, Colorado: The First Conference on Software Product Line Engineering, 2000.
- [38] D. Weiss, C. Lai, and R. Tau, Software product-line engineering: a family-based software development process. Addison-Wesley, Reading, MA, 1999.
- [39] K. C. Kang, S. Kim, J. Lee, K. Kim, E. Shin, and M. Huh, FORM: A Feature-Oriented Reuse Method with Domain- Specific Reference Architectures, Annals of Software Engineering, vol. 5, 1998, pp. 143 - 168.
- [40] C. Atkinson et al., Component-based product line engineering with UML. Addison-Wesley, London, New York, 2002.
- [41] Mikael Svahnberg, Claes Wohlin, Lars Lundberg, Michael Mattsson, A Quality-Driven Decision-Support Method for Identifying Software Architecture Candidates, 2003.
- [42] M. Matinlassi, E. Niemel, and L. Dobrica, Quality-driven architecture design and quality analysis method, A revolutionary initiation approach to a product line architecture, VTT Technical Research Centre of Finland, Espoo, 2002.
- [43] F. Bachmann, L. Bass, G. Chastek, P. Donohoe, and F. Peruzzi, The Architecture Based Design Method, CMU/SEI, Technical report 2000-TR-001, 2000.
- [44] Hassan Gomaa, Designing Software Product Lines with UML 2.0: From Use Cases to Pattern-Based Software Architectures, 10th International Software Product Line Conference (SPLC’06) 0-7695-2599-7/06 \$20.00 © 2006 IEEE, 2006.
- [45] Jianli Dong, Jianzhou Wang, Donghuai Sun, Haiyan Lu, The Research of Software Product Line Engineering Process and Its Integrated Development Environment Model, International Symposium on Computer Science and Computational Technology, 978-0-7695-3498-5/08 \$25.00 © 2008 IEEE DOI 10.1109/ISCSCCT.2008.100, 2008.
- [46] Jiayi Zhu, Xin Peng, Stan Jarzabek, Zhenchang Xing, Yinxing Xue, Wenyun Zhao, Improving Product Line Architecture Design and Customization by Raising the Level of Variability Modeling, 2011.
- [47] M.Sharafi, S.Dadollahi, A Scenario-Based Approach for Architecture Reconstruction of Product Line, 2013.
- [48] Hataichanok Unphon, Making Use of Architecture throughout the Software Life Cycle – How the Build Hierarchy can Facilitate Product Line Development, ICSE’09 Workshop , SHAR K’09, May 16, 2009, Vancouver, Canada 978-1-4244-3726-9/09/\$25.00 , 2009.
- [49] Jing Liu, Josh Dehlinger, Robyn Lutz, Safety analysis of software product lines using state-based modeling, 2007.
- [50] Thelma Elita Colanzi, Silvia Regina Vergilio, Representation of Software Product Line Architectures for Search-Based Design, CMSBSE 2013, San Francisco, CA, USA, 978-1-4673-6284-9/13/\$31.00 c 2013 IEEE, 2013.
- [51] L. Bass, M. Klein, and F. Bachmann, Quality Attribute Primitives and the Attribute Driven Design Method, in 4th International Workshop on Software Product-Family Engineering, F. van der Linden, Ed. Springer, Berlin Heidelberg, pp. 163 – 176,2002.
- [52] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, in: 12th International Conference on Evaluation and Assessment in Software Engineering (EASE). University of Bari, Italy, 26–27 June 2008.
- [53] Emelie Engström, Per Runeson, Software product line testing – A systematic mapping study, Information and Software Technology 53 (2011), 2011.
- [54] R. Wieringa, N. Maiden, N. Mead, C. Rolland, et al., Requirements engineering paper classification and evaluation criteria: a proposal and a discussion, Requirements Engineering 11 (1) (2006) 102–107.