CORE ASSETS DEVELOPMENT APPROACH USING SERVICE ORIENTED PRODUCT LINE FOR HEALTH CARE INFORMATION SYSTEM

AIN BALQIS BINTI MD NOR BASMMI

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Master of Philosophy

School of Computing
Faculty of Engineering
Universiti Teknologi Malaysia

APRIL 2022

DEDICATION

This thesis is dedicated to my mama and daddy, with love.

And for you, Ain Balqis. You survived.

ACKNOWLEDGEMENT

In preparing this thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed towards my understanding and thoughts. In particular, I wish to express my sincere appreciation to my main thesis supervisor, Dr Shahliza binti Abd Halim, for encouragement, guidance, critics and friendship. Her immense help gave a big impact on me throughout the journey of completing this project. Words can never be enough to express my gratitude. I am also very thankful to my co-supervisor Prof. Dr. Dayang Norhayati Abang Jawawi for her guidance, advices and motivation. Without her continued support and interest, this thesis would not have been the same as presented here.

I am also indebted to Universiti Teknologi Malaysia (UTM) for funding my Master study. Librarians at UTM also deserve special thanks for their assistance in supplying the relevant literatures.

I thank my fellow postgraduate friend, Nurul Nazihah binti Jamal, for the sleepless night we were working together before deadlines, the ideas we shared, the fun we had, and the sweat and tears we face together for the past 2 years. Your existence has helped me go through the worst to get to where I am today. Meryl Nadia binti Esrail and Amylia Syazwani binti Ahamad Tamizi, who should also be recognised for their support and continuous affirmations when I needed one. My fellow postgraduate friend, Muhammad Irsyad bin Kamil Riadz for the help in the process of completing my research. My sincere appreciation also extends to my fellow friends whom I just met during my post-graduate study but has been nothing but an amazing family of mine: Dr. Fateen Nabilla binti Rasli, Zainab binti Yahya Arif, and Aimi Dayana binti Jairen. And to all my lab mates, colleagues, and others who have provided assistance at various occasions. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space. I am grateful to all of you who played a role in my academic accomplishment throughout my post-graduate journey.

ABSTRACT

Service Oriented Product Line (SOPL) has emerged as a synergy that is able to assist systematic reuse by reusing an existing asset resulting to shorter product's time to market. Since huge effort is required to adjust the demand for specific requirements and fulfil the need for production of multiple systems, a systematic approach is very much needed particularly in identifying only possible core assets to be reused and managing variability between different applications. Current development of core assets in SOPL did not adequately resolve systematic reuse in identification of services to design a Reference Architecture (RA). Variations and common assets need to be addressed in order to ensure systematic reuse of core assets. Therefore, this study proposes an enhancement of the systematic reuse approach in determining reusable core assets by concentrating on the Domain Engineering Activity. This leads to the use of Trade-off Analysis between structural stability and priority of added services depending on the probability of occurrence for multiple user preferences. The approach involves specific processes which core asset identification and architectural modelling representation. The set of identified reusable core assets is represented into RA model with conceptual and physical levels representation. Proposed approach is validated through quantitative and qualitative evaluations to measure the applicability of core assets and ensure the artifacts built are according to comparative criteria. The proposed identification technique proved that it can improve reusability of identified core assets, recording 92% Functional Commonality and 80% of Modularity, hence resulting in a more refined RA. However, the result of core assets applicability appears to be lower by 9% compared to existing approach due to the multiple user requirements. This indicates that core assets are applicable to be reused by only one application instead of various applications. Regardless, reuse of core assets need to consider preferences and requirements from multiple users who want a customized system domain suitable for their organization.

ABSTRAK

Barisan Produk Berorientasikan Perkhidmatan (SOPL) adalah sinergi yang mampu membantu guna semula sistematik dengan mengguna semula aset sedia ada, memendekkan masa diperlukan untuk pemasaran produk. Pendekatan sistematik amat diperlukan dalam mengenal pasti aset teras yang boleh diguna semula dan mengurus variasi aplikasi berbeza kerana tidak mudah untuk menentukan permintaan keperluan khusus dan memenuhi keperluan pengeluaran pelbagai sistem berbeza. Kini, pembangunan aset teras SOPL tidak berupaya menyelesaikan guna semula sistematik dan mengenal pasti servis untuk mereka bentuk Seni Bina Rujukan (RA). Aset berbeza dan serupa perlu ditangani dengan betul untuk memastikan guna semula aset teras sistematik. Kajian ini mencadangkan penambahbaikan kaedah guna semula sistematik untuk menentukan aset teras guna semula dengan menumpukan kepada Aktiviti Kejuruteraan Domain yang memerlukan penggunaan Analisis Keseimbangan antara kestabilan struktur dan keutamaan servis tambahan berpandukan kebarangkalian kejadian berdasarkan kepelbagaian pilihan pengguna. Kaedah ini melibatkan proses khusus termasuk mengenal pasti aset teras dan perwakilan model seni bina. Set aset teras guna semula yang dikenal pasti akan diwakilkan dalam bentuk model RA melalui perwakilan peringkat konseptual dan fizikal. Kaedah dicadangkan ditentusahkan melalui penilaian kuantitatif dan kualitatif untuk mengukur kebolehgunaan aset teras dan memastikan artifak yang dibina adalah berpandukan kriteria bandingan. Teknik identifikasi dicadangkan membuktikan penambahbaikan guna semula aset teras dikenal pasti dengan mencatatkan 92% Fungsi Keserupaan dan 80% Kemodularan bersamaan RA yang lebih baik. Namun, keputusan kebolehgunaan aset teras kaedah dicadangkan ialah 9% lebih rendah berbanding pendekatan sedia ada akibat kehendak pelbagai pengguna. Ini menunjukkan bahawa aset teras hanya boleh diguna semula oleh satu aplikasi sahaja dan tidak boleh diguna semula oleh lebih dari satu aplikasi. Walau bagaimanapun, penggunaan semula aset teras mengambil kira keutamaan dan keperluan pelbagai pengguna yang mahukan sistem domain yang bersesuaian dengan organisasi mereka.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BPMN	-	Business Process Model and Notation
CBSD	-	Component Based Software Development
C	-	Structural Complexity Metric
C_1	-	Subsystem Complexity
C_2	-	Interactional Complexity
C_3	-	Topological Complexity
CC	-	Cyclomatic Complexity
COTS	-	Commercial Off-The-Shelf
EHR	-	Electronic Health Record
FM	-	Feature Model
FMT	-	Feature Model Type
GIS	-	Geographic Information system
HBT	-	Heinsohn Business Type
HIS	-	Healthcare Information System
HR	-	Health Record
IC	-	Integrity Constraint
IT	-	Internet Technology
MT	-	Model Type
NFR	-	Non-Functional Requirement
PL	-	Product Line
PNP	-	Patient Navigation System
QoS	-	Quality of Service
ROI	-	Return of Investment
SC	-	Service Consumer
SCp	-	Structural Complexity
SI	-	Service Identification
SMS	-	Systematic Mapping Study
SO	-	Service Oriented
SOA	-	Service Oriented Architecture
SOPL	-	Service Oriented Product Line

SP	-	Service Provider
SPL	-	Software Product Line
SPLIT	-	Software Product Line Integration Tool
UML	-	Unified Modelling Language

LIST OF SYMBOLS

$\sum$	-	Total sum
σ	-	Sigma

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Overview

Service-Oriented Architecture (SOA) is an architectural model for building systems based on the interaction of services as it is realizable by service-orientation which allow flexible application composition quickly and cost-efficiently (Ezenwoke *et al.*, 2013). SOA also aims to modularize and develop independent service (Juan Carlos Herrera *et al.*, 2016b) as it is able to manage rapidly changing business environments via runtime (Jaejoon Lee *et al.*, 2010). To increase the productivity of business, software reuse is a critical factor that contributes in increasing the productivity of businesses (Garusinghe *et al.*, 2017b). In this case, SOA has proven to be useful architecture for increasing productivity and reusability (Garusinghe *et al.*, 2017b). To enable an effective and systematic reuse of software components which are not supported by SOA (Abu-Matar and Gomaa, 2013), variations of similar software products should be considered. This is achievable via Software Product Line (SPL) paradigm (Ezenwoke *et al.*, 2013).

SPL is a software reuse paradigm for the development of a system which reduces time to market and improves quality by developing core assets that enable systematic reuse (Jaejoon Lee *et al.*, 2010). SPL aims at identifying variabilities and commonalities in a product family developed by one producer alone (Juan Carlos Herrera *et al.*, 2016b). Since 1990, software engineer have been paying close attention to SPL due to the benefits it offers, such as reuse in software development, time to market reduction, cost reduction, improved software quality and customer satisfaction (Chacón-Luna *et al.*, 2020). Studies on the synergy of SPL and SOA is considered as an emerging topic in recent years (Garusinghe *et al.*, 2017a; Khoshnevis and Shams, 2017a; Lu *et al.*, 2019). The fusion of SOA and SPL can be regarded as Service Oriented Product Line (SOPL) (Ezenwoke *et al.*, 2013).

Rather than designing everything from scratch which will consume more time and cost, SOPL provides a more systematic way to reuse core assets to meet specific stakeholder's need that are common throughout application (Garusinghe *et al.*, 2017b). However, to adequately identify the particular core assets to be reused might be one of the biggest challenges for most researchers. SOPL accommodates these challenges by modifying the software to be reused or composing new orchestration of services (Castelluccia and Boffoli, 2014b). Reusing services can be strengthened by developing a product line that meets several criteria from various customer's applications, and variabilities is included in the design of the product line following SOA principles and implemented using services or components (Ribeiro *et al.*, 2011). From the concept of core asset development, SOPL enables systematic reuse of core assets and satisfies the needs for mass customization (Lee, 2012).

This study will focus on the adaptation of SPL development life cycle into SOA is being focussed. SPL process is divided into two activities which are Domain Engineering (DE) and Application Engineering (AE). DE defines the development of core assets while AE defines the development of product using the core assets (Ezenwoke *et al.*, 2013; Achour, Labed and Ben Ghezala, 2015; Ali, 2018). There are four processes commonly involved in DE which are: Domain Scoping (DS), Domain Analysis (DA), Domain Design (DD), and Domain Implementation (DI) (Ezenwoke *et al.*, 2013; Losavio *et al.*, 2016). However, this study only covers the development of core assets in DE that aims to define variability and commonality of a system to establish a set of services to be reused. (Alférez and Pelechano, 2011). Several researches have applied systematic approach for developing core assets via DE activities (Galster and Eberlein, 2011; Ezenwoke *et al.*, 2013; Parra and Joya, 2015). However, only a limited number of existing works offer a complete DE activity.

There exist certain challenges when adapting SPL development life cycle into SOA. One of the main challenges is the role of service and how SOPL can be extended to make it applicable for a family of products instead of a single system (Serajzadeh and Shams, 2012). SOPL necessitates the management of variability within service composition, variation among products requirement, identifying service variability, and defining architectural views (Castelluccia and Boffoli, 2014a). Furthermore, managing variability is a large part of SPL since variability tends to provide requisite flexibility for service identification and diversification (Castelluccia and Boffoli, 2014a). In this case, SOPL is considered as a systematic approach that provides guidance in solving the challenges aforementioned which are not sufficiently being researched by existing studies. Therefore, an approach that addresses and deals with variability during core asset development is crucial and need to be further investigated.

1.2 Problem Background

The combination of SPL and SOA paradigms is said to promise better reuse with systematic and large-scale reuse (Benabdelali *et al.*, 2015). SOA refers to an architectural model where the logic of information is broken down into smaller, distinct logic units that are used collectively to construct a broader system of market automation (Tirapathi, 2016). However, SOA alone is not able to provide a systematic reuse, and a way to design SOA that is highly customizable and that supports planned reused is by integrating SPL into SOA paradigm (Galster and Eberlein, 2011; Mohabbati *et al.*, 2014; Garusinghe *et al.*, 2017). SOA takes advantage of SPL which aims at providing systematic reuse (Abu-matar, 2014) by improving reusability and remove wasteful generic development of components (Mohabbati *et al.*, 2014). Many approaches have proposed integrating systematic reuse of SPL in facilitating SOA development (Mohabbati *et al.*, 2014; Losavio *et al.*, 2015; Garusinghe *et al.*, 2017a; Lu *et al.*, 2019) which was then called SOPL.

Although SOPL offers systematic reuse towards development of core assets, there is an uncovered challenge arises on how to identify only potential core assets to be reused (Galster and Eberlein, 2011; Zhu *et al.*, 2013). Lack of systematic reuse approach has been addressed in managing variability in requirement and architecture which causes inadequate identification of core assets to be reused, in turn results to low reusability of core assets (Kang and Baik, 2010; Serajzadeh and Shams, 2012; Gu and Lago, 2013; Garusinghe *et al.*, 2017a; Ali, 2018). This challenge needs to be solved since the primary outcome of core assets development is a set of highly reusable core assets (Zhu *et al.*, 2013). Furthermore, since variability in SOA is addressed through loose coupling of service and dynamic retrieval, facilitating variability can lead to different instances of SOA (Galster and Eberlein, 2011). Therefore, a systematic process that can manage variability during core assets development is needed to treat different instances of a service-based system as a member of SPL as it shares commonalities but vary in certain aspects.

There exist quite a number of research on systematic reuse through SOPL (Ezenwoke *et al.*, 2013; Parra and Joya, 2015; Imran Abbasi and M. Mackenzie, 2017). In the scope of core assets development during DE, four sub-processes—DS, DA, DD, and DI—are commonly being applied by researchers to achieve better core assets reusability (Ezenwoke *et al.*, 2013; Losavio *et al.*, 2015). However, only a limited number of existing works offers a complete DE activity of identifying reusable core assets and most research that tackle DE activity starts with DA whilst neglecting the first process, DS (Imran Abbasi and M. Mackenzie, 2017; Kamoun, Kacem and Kacem, 2017; Lu *et al.*, 2019). DS is the process where information used in developing software systems within the domain is identified, captured and organized with the purpose of making it reusable when building new products (America *et al.*, 2001). The absence of DS process means there is lack of planning in obtaining highly reusable core assets (Jihyun Lee *et al.*, 2010).

DS is often represented in a form of Product Map (Schmid, 2002; Ezenwoke *et al.*, 2013) and proven to support reusability (Jihyun Lee *et al.*, 2010). However, reuse approaches that is being proposed by most researchers are solely based on user's requirements without incorporating other aspect of reuse such as the priorities of the core assets (Ezenwoke *et al.*, 2013; Mohabbati *et al.*, 2014; Juan Carlos Herrera *et al.*, 2016a). Priorities are strategies used by developers which may help in specifying intended resolution of feature interactions (Soares *et al.*, 2018). Structure of the system that is generated should be considered in the earlier stages of development (Schmid, 2002). However, existing studies of DS only focus on reuse without concerning on high reusability of core assets (Jihyun Lee *et al.*, 2010).

Furthermore, there is a lack of focus on structural system is being considered at the early stage of development. Consequently, as more core assets are being reused, structure of the system may be affected thus leading to an increase in cost and effort for implementing and maintaining architecture (Galster and Eberlein, 2011). This shows that the structure of the system really depends on the core assets identified during the early stages. However, current studies may have disregarded this problem as most of them are only focusing on reusing core assets without concerning on the quality and architectural aspect of the system. Therefore, instead of only focusing on reuse of core assets, deriving core assets that can survive evolutionary changes or remain stable while accommodating the development and fulfilling user requirements is crucial to bring positive impact on the architecture (Dias *et al.*, 2010).

Apart from that, in developing reusable core assets, existing studies have discussed on how to accommodate core assets development in SPL by focusing on grouping core assets into feature model during DA (Kang and Baik, 2010; Alférez and Pelechano, 2011; Zhu *et al.*, 2013) which can aid service identification activity at certain extend. However, there is lack of detailed explanation on the approach of grouping core assets into commonalities and variabilities. This challenge needs to be tackled as it serves as the fundamental of system development and main output for core asset development (Zhu *et al.*, 2013). Therefore, a study is required on how to extend these existing approaches so that they are applicable for a new product family (Galster and Eberlein, 2011) and can be represented in Reference Architecture (RA) during DD.

DD is where an architecture is created with the variability of various services and is presented through various views for different stakeholders (Ezenwoke *et al.*, 2013; Achour, Labed and Ben Ghezala, 2015). RA is designed as a concrete architecture system in achieving a system with high structural stability and low structural complexity during DS (Nakagawa *et al.*, 2014). However, since SOPL is closely related with managing variability systematically throughout the development of core assets, most studies still show lack of contribution on these aspects when designing RA (Galster and Avgeriou, 2011; Galster *et al.*, 2013; Guessi *et al.*, 2014). Variability has been one of the common issues tackled by most researchers during the process of designing RA (Palanivel and Kuppuswami, 2011; Galster *et al.*, 2013; Guessi *et al.*, 2014; Losavio *et al.*, 2015; Juan Carlos Herrera *et al.*, 2016b; Abu-Matar and Mizouni, 2019). Despite the idea of addressing variability in an architecture, there is still lack of understanding about how variability can be handled properly in RA (Nakagawa *et al.*, 2014).

RA are usually designed in an unsystematically manner (Galster *et al.*, 2013; Mistrik *et al.*, 2019) thus making the RA complex (Martínez-Fernández *et al.*, 2013). One way to handle RA design systematically is by documenting the architecture (Phillippe Kruchten, 1995). However, there only exist a few number of studies that documented the architectural design in a proper manner such as studies from Flávio M. Medeiros *et al.* (2010). Without a proper documentation, variability description for constructing architectural views can't be supported (Guessi *et al.*, 2014; Bedjeti *et al.*, 2017) and it is hard to maintain the systematic manner of designing architecture. Therefore, in order to design RA while considering variability description systematically, a sufficient documentation should be provided.

Other than that, in adapting user's requirements, most software developers strive to provide views for different stakeholders (Flávio M. Medeiros *et al.*, 2010; Gomaa and Hashimoto, 2012; Abu-Matar and Gomaa, 2013). Different views are needed to define solution at many levels of abstraction and for different purposes (US Dept. of Defence / Office of the DoD CIO, 2010; Heuser *et al.*, 2017). To describe the entire software architecture of a system, multiple and concurrent diagrams are a common feature of graphical documentation to overcome issues such as crowded diagram, inconsistent notation, mixing architectural style, overemphasizing one element, and ignoring individual stakeholder concerns (Philippe Kruchten, 1995; May, 2005). Therefore, it is important to provide a basis for organization that can maximize the benefit for certain group of stakeholders in handling these issues. This provides additional challenge to DD process because most studies only focus on the development of conceptual level and forsake the physical level where mapping of software onto hardware and deployment process are being handled. Especially when the development involves web service composition, the ability to rapidly design and deploy service composition during physical level are very much needed (Ben Hadj Yahia and Laurent, 2017). Due to this, the architectural reusability cannot be maximized as not all stakeholder perspective is being considered (Suvelayutnan, 2018).

1.3 Problem Statement

Both SPL and SOA offer the concepts of reusability rather than redeveloping the system from scratch. Both methods may have different outlooks on reusability and as this research proposes the combination of these two approaches, a lot of critical things need to be taken into account. In identifying reusable assets, services that may be the foundation of SOPL need to be defined in the form of commonalities and variabilities. The selection mechanism should be based on a thorough examination of the context in which the service would actually be used (Garusinghe *et al.*, 2017a).

For DS, the main problems discovered involve the technique for identification of reusable core assets that concern the architectural of the system from the early phase. In addition, most studies do not open to incorporate multiple user requirements during DS. Besides, although SOPL offers reuse towards development of core assets, there are still uncovered challenges arises on how to identify only potential assets to be reused through SOPL as existing studies shows lack of systematic manner for reuse. Based on the problem related to DS, intended resolution of feature interactions is specified (Soares *et al.*, 2018) while considering the structural architecture of the developed system. Therefore, in emphasizing the long-term effects of the architectural structure while meeting user's requirements (Herrera et al., 2016b), an expressive identification technique of highly reusable core assets during DS is needed. The identified reusable core assets are then used as an input to design RA during DE.

Subsequently, in terms of DD, RA is usually complex (Martínez-Fernández *et al.*, 2013) because it is often designed in an unsystematically manner (Galster *et al.*, 2013; Mistrik *et al.*, 2019). Another main problem is the ability to provide a physical view that is able to maximize the benefit for certain group of stakeholders while managing architecture variability. It is critical to have a basis for organization that can optimize the advantage for a certain group of stakeholders (Philippe Kruchten, 1995;

May, 2005). Especially when web service composition is involved, the ability to build and deploy service compositions quickly at physical level is critical (Ben hadj yahia and Laurent, 2017) since the involvement of both conceptual and physical levels in designing RA is very much needed. Last but not least, to ensure that the enhancement made in this study is suitable to be used for developing core assets, the overall process is evaluated to analyze the artifacts built and to identify whether the end product is applicable to be reused across various applications.

From the previous discussion, the primary research question that this research is trying to address is:

"How to enhance systematic reuse approach of Domain Engineering Activity which comprises of Domain Scoping (DS), Domain Analysis (DA), and Domain Design (DD) in identifying reusable core assets for Service Oriented Product Line (SOPL)?"

A set of research questions are derived to support the primary research question as below:

1) To what extend does the core assets development through SOPL is being studied?
2) How do other research perform identification technique of reusable core assets for DS and grouping of commonality and variability for DA?
3) How do previous research handle variability management and architecture representation for identified reusable core assets?
4) What is the process used to evaluate applicability of core asset and artifacts built from the proposed systematic reuse approach in identifying reusable core asset is evaluated?

1.4 Research Aim

This research aims to enhance an existing systematic approach of Domain Engineering Activity which comprises of DS, DA, and DD in identifying reusable core assets for Service Oriented Product Line.

1.5 Research Objectives

In order to achieve the goal of this research as aforementioned, the following objectives are set:

1) To enhance an identification technique for highly reusable core assets for DS.
2) To enhance the variability representation for RA during DD based on the identified reusable core assets.
3) To evaluate the applicability of core asset and artifacts built from the proposed systematic approach in Core Assets Development for SOPL.

1.6 Research Scope

This research focuses on the following research scopes:

1) Static service composition during Domain Engineering Activity that consist of DS, DA, and DD only.

2) Architectural representation of multiple view RA to define solution at many levels of abstraction and for different purposes.

1.7 Research Contribution

The significance of this research is the enhancement of existing approaches on identifying and developing reusable core assets for Domain Specific Application through the implementation of SOPL. SOPL will involve three activities in DE which are DS, DA, and DD. SOPL is beneficial because it saves time (does not require a new system to be developed from scratch) and a reusable system can also help in enhancing the quality of a system. This research may also provide a guidance for other researchers on a systematic process to identify and represent the core assets to be reused by multiple stakeholders and on ways to evaluate the identified core assets.

The approach proposed in this research helps in deriving highly reusable core assets with a comprehensive identification technique during DS and provides a thorough view of RA with the addition of physical level view. Likewise, quality of the artifacts built and applicability of product from this study attract the interest of other researchers to use this approach for potential core asset development approach that involves multiple user preferences and requirements. To conclude, two main contributions addressed in this study are in terms of core assets identification technique and RA representation that includes conceptual and physical levels representation.

1.8 Research Organization

This thesis is organized into six chapters. The first chapter gives an introduction of the proposed research project and briefly explains the problem background that this study is trying to tackle. Chapter two discusses existing studies to examine the gaps and potential rooms for improvement. Chapter three describes the research methodology that consists of research framework, research process, and case study. The fourth chapter demonstrates the proposed systematic approach. Chapter five illustrates the evaluation of outcome for each objective and chapter six concludes this research by discussing the contributions and suggestions for future works.

REFERENCES

Abu-Matar, M., & Gomaa, H. (2011, June). Feature based variability for service oriented architectures. In 2011 *Ninth Working IEEE/IFIP Conference on Software Architecture* (pp. 302-309). IEEE.

Abu-Matar, M., Center, E. B. T. I., & Dhabi, A. (2013). Towards a Variability Modeling Approach for Service-Oriented Cloud Architectures.

Abu-Matar, M., & Gomaa, H. (2011, August). Variability modeling for service oriented product line architectures. In 2011 *15th International Software Product Line Conference* (pp. 110-119). IEEE.

Abu-Matar, M., & Gomaa, H. (2013, March). An automated framework for variability management of service-oriented software product lines. In 2013 *IEEE Seventh International Symposium on Service-Oriented System Engineering* (pp. 260-267). IEEE.

Abu-Matar, M., Gomaa, H., Kim, M., & Elkhodary, A. M. (2010, July). Feature Modeling for Service Variability Management in Service-Oriented Architectures. In *Software Engineering and Knowledge Engineering*. (pp. 468-473).

Abu-Matar, M., & Mizouni, R. (2018, September). Variability modeling for smart city reference architectures. In 2018 *IEEE International Smart Cities Conference (ISC2)* (pp. 1-8). IEEE.

Achour, I., Labed, L., & Ghezala, H. B. (2015, July). Formalization of secure service oriented product line. In 2015 *10th International Joint Conference on Software Technologies (ICSOFT)* (Vol. 1, pp. 1-8). IEEE.

Achour, I., Labed, L., & Ghezala, H. B. (2015, February). Proposition of secure service oriented product line. In 2015 *6th International Conference on Information Systems and Economic Intelligence (SIIE)* (pp. 52-59). IEEE.

Adjoyan, S., & Seriai, A. D. (2015, July). An architecture description language for dynamic service-oriented product lines. In *SEKE: Software Engineering and Knowledge Engineering* (pp. 231-236).

Adjoyan, S., & Seriai, A. D. (2017, April). Reconfigurable service-based architecture based on variability description. In *Proceedings of the Symposium on Applied Computing* (pp. 1154-1161).

Akhtar, U., Khattak, A. M., & Lee, S. (2016, May). Challenges in managing real-time data in health information system (HIS). In *International Conference on Smart Homes and Health Telematics* (pp. 305-313). Springer, Cham.

Al-Rawahi, N., & Baghdadi, Y. (2005, June). Approaches to identify and develop Web services as instance of SOA architecture. In *Proceedings of ICSSSM'05. 2005 International Conference on Services Systems and Services Management, 2005*. (Vol. 1, pp. 579-584). IEEE.

Pelechano, V. (2011, September). Systematic reuse of web services through software product line engineering. In *2011 IEEE Ninth European Conference on Web Services* (pp. 192-199). IEEE.

Salinas, G. H. A., & Salinas, E. M. A. (2011). An Aspect-Oriented Framework to Model Non-Functional Requirements in Software Product Lines of Service-Oriented Architectures. In *Non-Functional Properties in Service Oriented Architecture: Requirements, Models and Methods* (pp. 246-267). IGI Global.

Ali, A. R. S. (2018). A Framework for Evaluating Reusability of Core Assets using SPL and SOA (Doctoral dissertation, Sudan University of Science & Technology).

Altintas, N. I., Cetin, S., Dogru, A. H., & Oguztuzun, H. (2011). Modeling product line software assets using domain-specific kits. *IEEE transactions on software engineering*, 38(6), (pp 1376-1402).

Alzahmi, S. M., Abu-Matar, M., & Mizouni, R. (2014, April). A practical tool for automating service oriented software product lines derivation. In *2014 IEEE 8th International Symposium on Service Oriented System Engineering* (pp. 90-97). IEEE.

Amoud, M., & Roudies, O. (2017, June). Dynamic adaptation and reconfiguration of security in mobile devices. In *2017 International Conference On Cyber Incident Response, Coordination, Containment & Control* (Cyber Incident) (pp. 1-6). IEEE.

Arango, G. (1989). Domain analysis: From art form to engineering discipline. *ACM Sigsoft software engineering notes*, 14(3), (pp. 152-159).

Abbaspour Asadollah, S., Inam, R., & Hansson, H. (2015, November). A survey on testing for cyber physical system. In *IFIP International Conference on Testing Software and Systems* (pp. 194-207). Springer, Cham.

Azouzi, S., Ghannouchi, S. A., & Brahmi, Z. (2017, September). Software product line to express variability in e-learning process. In *European, Mediterranean, and Middle Eastern Conference on Information Systems* (pp. 173-185). Springer, Cham.

Baghdadi, Y. (2014). Modelling business process with services: towards agile enterprises. *International Journal of Business Information Systems*, 15(4), 410-433.

Bagheri, E., & Gasevic, D. (2011). Assessing the maintainability of software product line feature models using structural metrics. *Software Quality Journal*, 19(3), 579-612.

Bashari, M., Bagheri, E., & Du, W. (2018). Automated composition and optimization of services for variability-intensive domains. *Journal of Systems and Software*, 146, 356-376.

Caldiera, V. R. B. G., & Rombach, H. D. (1994). The goal question metric approach. *Encyclopedia of software engineering*, 528-532.

Bass, L., Clements, P., Kazman, R., & Klein, M. (2008, February). Evaluating the software architecture competence of organizations. In *Seventh Working IEEE/IFIP Conference on Software Architecture* (WICSA 2008) (pp. 249-252). IEEE.

Bedjeti, A., Lago, P., Lewis, G. A., De Boer, R. D., & Hilliard, R. (2017, April). Modeling context with an architecture viewpoint. In *2017 IEEE International Conference on Software Architecture* (ICSA) (pp. 117-120). IEEE.

Beloglazov, A., Banerjee, D., Hartman, A., & Buyya, R. (2015). Improving productivity in design and development of information technology (IT) service delivery simulation models. *Journal of Service Research*, 18(1), 75-89.

Benabdelali, M., Achour, I., Jilani, L. L., & Ghezala, H. B. (2015). Toward a new classification of Non-Functional Requirements for Service Oriented Product Line. *28th International Conference on Computer Applications in Industry and Engineering, CAINE*, 307-312.

Berger, T., Steghöfer, J. P., Ziadi, T., Robin, J., & Martinez, J. (2020). The state of adoption and the challenges of systematic variability management in industry. *Empirical Software Engineering*, 25(3), 1755-1797.

Böckle, G., Muñoz, J. B., Knauber, P., Krueger, C. W., Leite, J. C. S. D. P., Linden, F. V. D., ... & Weiss, D. M. (2002, August). Adopting and institutionalizing a product line culture. In *International Conference on Software Product Lines* (pp.

49-59). Springer, Berlin, Heidelberg.

Buzurovic, I., Podder, T. K., Fu, L., & Yu, Y. (2010, May). Modular software design for brachytherapy image-guided robotic systems. In 2010 *IEEE International Conference on BioInformatics and BioEngineering* (pp. 203-208). IEEE.

Capilla, R., Bosch, J., Trinidad, P., Ruiz-Cortés, A., & Hinchey, M. (2014). An overview of Dynamic Software Product Line architectures and techniques: Observations from research and industry. *Journal of Systems and Software*, 91, 3-23.

Cardona, V., & Duarte, H. (2013). Approach for the Model Driven Development of Business Processes Lines Based on Service Oriented Architectures. *American International Journal of Contemporary Research*, 3(7), 78-87.

Castelluccia, D., & Boffoli, N. (2014). Service-oriented product lines: A systematic mapping study. *ACM SIGSOFT Software Engineering Notes*, 39(2), 1-6.

Leite, J. C. S. D. P., Yu, Y., Liu, L., Yu, E. S., & Mylopoulos, J. (2005, June). Quality-based software reuse. In *International Conference on Advanced Information Systems Engineering* (pp. 535-550). Springer, Berlin, Heidelberg.

Chacón-Luna, A. E., Gutiérrez, A. M., Galindo, J. A., & Benavides, D. (2020). Empirical software product line engineering: a systematic literature review. *Information and Software Technology*, 128, 106389.

Choi, S. W., & Kim, S. D. (2008, July). A quality model for evaluating reusability of services in soa. In 2008 *10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services* (pp. 293-298). IEEE.

Clements, P. C., Jones, L. G., Northrop, L. M., & McGregor, J. D. (2005). Project management in a software product line organization. *IEEE software,* 22(5), 54-62.

Cognini, R., Corradini, F., Polini, A., & Re, B. (2015, June). Extending feature models to express variability in business process models. In *International Conference on Advanced Information Systems Engineering* (pp. 245-256). Springer, Cham.

Cognini, R., Corradini, F., Polini, A., & Re, B. (2016). Business process feature model: an approach to deal with variability of business processes. In *Domain-Specific Conceptual Modeling* (pp. 171-194). Springer, Cham.

Cohen, S., & Krut, R. (2010). Managing variation in services in a software product line context. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE

ENGINEERING INST.

Cubo, J., Gamez, N., Fuentes, L., & Pimentel, E. (2013, June). Composition and self-adaptation of service-based systems with feature models. In *International Conference on Software Reuse* (pp. 326-342). Springer, Berlin, Heidelberg.

Cubo, J., Gamez, N., Pimentel, E., & Fuentes, L. (2015, June). Reconfiguration of service failures in damasco using dynamic software product lines. In 2015 *IEEE International Conference on Services Computing* (pp. 114-121). IEEE.

Khoshnevis, S., & Shams, F. (2015, April). Linear Evolution of Domain Architecture in Service-Oriented Software Product Lines. In *International Conference on Fundamentals of Software Engineering* (pp. 275-291). Springer, Cham.

Dehmouch, I. (2014, May). Towards an agile feature composition for a large scale software product lines. In 2014 *IEEE Eighth International Conference on Research Challenges in Information Science* (RCIS) (pp. 1-6). IEEE.

Dias, M. O., Tizzei, L. P., Rubira, C. M., Garcia, A. F., & Lee, J. (2010). Leveraging Aspect-Connectors to Improve Stability of Product-Line Variabilities. In *VaMoS* (pp. 21-28).

Dieste, O., Grimán, A., & Juristo, N. (2009). Developing search strategies for detecting relevant experiments. *Empirical Software Engineering*, 14(5), 513-539.

Etedali, A., Lung, C. H., Ajila, S., & Veselinovic, I. (2017, July). Automated constraint-based multi-tenant saas configuration support using XML filtering techniques. In 2017 *IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 2, pp. 413-418). IEEE.

Evers, S., Ernsting, J., & Majchrzak, T. A. (2016, January). Towards a reference architecture for model-driven business apps. In 2016 *49th hawaii international conference on system sciences (hicss)* (pp. 5731-5740). IEEE.

Ezenwoke, A., Misra, S., & Adigun, M. O. (2013). An approach for e-commerce on-demand service-oriented product line development. *Acta Polytechnica Hungarica*, 10(2), 69-87.

Fægri, T. E., & Hallsteinsen, S. (2006). A software product line reference architecture for security. In *Software Product Lines* (pp. 275-326). Springer, Berlin, Heidelberg.

Fajar, A. N., & Shofi, I. M. (2016, April). Development of SPL government system with ontology Web language. In *2016 4th International Conference on Cyber and IT Service Management* (pp. 1-4). IEEE.

Fajar, A. N., Utama, D. N., & Wang, G. (2018, September). Intelligent Software Product Line For Supply Chain. In *Journal of Physics: Conference Series* (Vol. 1090, No. 1, p. 012043). IOP Publishing.

Falvo, V., Duarte Filho, N. F., Oliveira, E., & Barbosa, E. F. (2014, October). A contribution to the adoption of software product lines in the development of mobile learning applications. In 2014 *IEEE Frontiers in Education Conference (FIE) Proceedings* (pp. 1-8). IEEE.

Hillebrand, C., & Coetzee, M. (2015, September). The design of a configurable reputation service. In *International Conference on Trust and Privacy in Digital Business* (pp. 60-70). Springer, Cham.

Freeman, H. P., & Rodriguez, R. L. (2011). History and principles of patient navigation. Cancer, 117(S15), 3537-3540.

Gallagher, B. P. (2000). Using the architecture tradeoff analysis methodsm to evaluate a reference architecture: a case study. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

Galster, M., & Avgeriou, P. (2011, June). Empirically-grounded reference architectures: a proposal. In *Proceedings of the joint ACM SIGSOFT conference--QoSA and ACM SIGSOFT symposium--ISARCS on Quality of software architectures--QoSA and architecting critical systems--ISARCS* (pp. 153-158).

Galster, M., Avgeriou, P., & Tofan, D. (2013). Constraints for the design of variability-intensive service-oriented reference architectures–An industrial case study. *Information and Software Technology*, 55(2), 428-441.

Galster, M., Avgeriou, P., Weyns, D., & Männistö, T. (2011). Variability in software architecture: current practice and challenges. *ACM SIGSOFT Software Engineering Notes*, 36(5), 30-32.

Galster, M., & Eberlein, A. (2011, April). Identifying potential core assets in service-based systems to support the transition to service-oriented product lines. In 2011 *18th IEEE International Conference and Workshops on Engineering of Computer-Based Systems* (pp. 179-186). IEEE.

Gamez, N., El Haddad, J., & Fuentes, L. (2015, June). SPL-TQSSS: a software product line approach for stateful service selection. In 2015 *IEEE International Conference on Web Services* (pp. 73-80). IEEE.

Garusinghe, A., Perera, I., & Meedeniya, D. (2016, September). Managing Service Level Agreements in Service Oriented Product Lines. In 2016 *Sixteenth*

International Conference on Advances in ICT for Emerging Regions (ICTer) (pp. 274-280). IEEE.

Garusinghe, A. I., Perera, I., & Meedeniya, D. (2017). Service oriented product lines-managed service level agreements for better quality of service. *ICTer*, 10(2).

Paliwal, G., & Kiwelekar, A. W. (2015). A product line architecture for mobile patient monitoring system. In *Mobile Health* (pp. 489-511). Springer, Cham.

Godse, M., Sonar, R., & Mulik, S. (2008, November). The analytical hierarchy process approach for prioritizing features in the selection of web service. In 2008 *Sixth European Conference on Web Services* (pp. 41-50). IEEE.

Goknil, A., Kurtev, I., & van den Berg, K. (2016, November). A rule-based approach for evolution of AADL models based on changes in functional requirements. In *Proccedings of the 10th European Conference on Software Architecture Workshops* (pp. 1-7).

Gomaa, H., Albassam, E., & Menascé, D. A. (2017, September). Run-time Software Architectural Models for Adaptation, Recovery and Evolution. In *MODELS* (Satellite Events) (pp. 193-200).

Gomaa, H., & Hashimoto, K. (2012, June). Dynamic self-adaptation for distributed service-oriented transactions. In 2012 *7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)* (pp. 11-20). IEEE.

Granell, C., Díaz, L., & Gould, M. (2010). Service-oriented applications for environmental models: Reusable geospatial services. *Environmental Modelling & Software*, 25(2), 182-198.

Gu, Q., & Lago, P. (2010, December). Service identification methods: a systematic literature review. In *European Conference on a Service-Based Internet* (pp. 37-50). Springer, Berlin, Heidelberg.

Guessi, M., Oquendo, F., & Nakagawa, E. Y. (2014). Variability viewpoint to describe reference architectures. In *Proceedings of the WICSA 2014 Companion Volume* (pp. 1-6).

Guizzardi, G. (2013). Ontology-based evaluation and design of visual conceptual modeling languages. In *Domain engineering* (pp. 317-347). Springer, Berlin, Heidelberg.

Jadeja, Y., & Modi, K. (2012, March). Cloud computing-concepts, architecture and challenges. In 2012 *international conference on computing, electronics and* *electrical technologies (ICCEET)* (pp. 877-880). IEEE.

Yahia, E. B. H. (2017). A language-based approach for web service composition (Doctoral dissertation, Universite de Bordeaux).

Hassan, A. E., & Holt, R. C. (2000, November). A reference architecture for web servers. In *Proceedings Seventh Working Conference on Reverse Engineering* (pp. 150-159). IEEE.

Helu, M., Hedberg Jr, T., & Feeney, A. B. (2017). Reference architecture to integrate heterogeneous manufacturing systems for the digital thread. *CIRP journal of manufacturing science and technology*, 19, 191-195.

Her, J. S., Kim, J. H., Oh, S. H., Rhew, S. Y., & Kim, S. D. (2007). A framework for evaluating reusability of core asset in product line engineering. *Information and Software Technology*, 49(7), 740-760.

Carlos Herrera, J., Losavio, F., & Ordaz, O. (2016). QuaDRA: Quality-oriented Design of Reference Architecture for Software Product Lines based on ISO/IEC 26550. *Revista Antioqueña de las Ciencias Computacionales*, 6(1).

Herrera, J. C., Losavio, F., & Ordaz, O. (2016). Web-services reference architecture for software product lines: A quality-driven approach Arquitectura de referencia de servicios web para líneas de productos de software: Un enfoque dirigido por la calidad.

Heuser, L., Scheer, J., Den Hamer, P., & De Lathouwer, B. (2017). Reference architecture & design principles. *EIP-SCC Work Stream*, *2*.

Abbasi, M. I., & Mackenzie, L. M. (2017). A Flexible Approach for Modelling and Analysis of Feature Interactions in Service-Oriented Product Lines. *J. Softw.*, 12(10), 823-830.

John, I. (2010). Using documentation for product line scoping. *IEEE software*, 27(3), 42-47.

Jula, A., Sundararajan, E., & Othman, Z. (2014). Cloud computing service composition: A systematic literature review. *Expert systems with applications*, 41(8), 3809-3824.

Kamoun, A., Kacem, M. H., & Kacem, A. H. (2014, November). Feature model for modeling compound SOA design patterns. In 2014 *IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)* (pp. 381-388). IEEE.

Kamoun, A., Kacem, M. H., & Kacem, A. H. (2014, November). Feature model for modeling compound SOA design patterns. In 2014 *IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)* (pp. 381-388). IEEE.

Kamoun, A., Kacem, M. H., & Kacem, A. H. (2016, December). Feature model for the service provider in the service oriented architecture. In 2016 *17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)* (pp. 705-710). IEEE.

Kamoun, A., Kacem, M., Kacem, A., & Drira, K. (2017, April). Feature model based on design pattern for the service provider in the service oriented architecture. In *The 19th International Conference on Enterprise Information Systems (ICEIS'2017)* (Vol. 2, pp. 111-120).

Kamoun, A., Kacem, M. H., Kacem, A. H., & Drira, K. (2019). Feature models as service contracts in service oriented architecture. *International Journal of Services Technology and Management*, 25(3-4), 267-288.

Kang, D., & Baik, D. K. (2010, August). Bridging software product lines and service-oriented architectures for service identification using bpm and fm. In *2010 IEEE/ACIS 9th International Conference on Computer and Information Science* (pp. 755-759). IEEE.

Karthikeyan, T., & Geetha, J. (2012). A Study and Critical Survey on Service Reusability Metrics. *International Journal of Information Technology and Computer Science (IJITCS)*, 4(5), 25-31.

Katz, S., Dabrowski, C., Miles, K., & Law, M. (1994). *Glossary of Software Reuse Terms*. Gaithersburg: National Institute of Standards and Technology.

Keswani, R., Joshi, S., & Jatain, A. (2014, February). Software reuse in practice. In 2014 *Fourth International Conference on Advanced Computing & Communication Technologies* (pp. 159-162). IEEE.

Khoshkbarforoushha, A., Jamshidi, P., Gholami, M. F., Wang, L., & Ranjan, R. (2014). Metrics for BPEL process reusability analysis in a workflow system. *IEEE Systems Journal*, 10(1), 36-45.

Khoshnevis, S., & Shams, F. (2017). Automating identification of services and their variability for product lines using NSGA-II. *Frontiers of Computer Science*, 11(3), 444-464.

Kruchten, P. (1995). Architectural Blueprints-The. The, 4(1), 42-50.

Kruchten, P. (2001). Architecture blueprints: The "4+ 1" view model of software architecture. Retrieved August 19, 2004.

Kugele, S., Obergfell, P., Broy, M., Creighton, O., Traub, M., & Hopfensitz, W. (2017, April). On service-orientation for automotive software. In 2017 *IEEE International Conference on Software Architecture (ICSA)* (pp. 193-202). IEEE.

La, H. J., Her, J. S., & Kim, S. D. (2013, May). Framework for evaluating reusability of component-as-a-service (caas). In 2013 *5th International Workshop on Principles of Engineering Service-Oriented Systems (PESOS)* (pp. 41-44). IEEE.

Lee, J., Muthig, D., Naab, M., Kim, M., & Park, S. (2008, May). Identifying and specifying reusable services of service centric systems through product line technology. In Proceedings of the First Workshop on Service-Oriented Architectures and Software Product Lines (pp. D1-D11).

Lee, J., Kang, S., & Lee, D. (2010). A comparison of software product line scoping approaches. International Journal of Software Engineering and Knowledge Engineering, 20(05), 637-663.

Lee, J., & Kotonya, G. (2010). Combining service-orientation with product line engineering. *IEEE software*, 27(3), 35-41.

Lee, J., Muthig, D., & Naab, M. (2010). A feature-oriented approach for developing reusable product line assets of service-based systems. *Journal of Systems and Software*, 83(7), 1123-1136.

Leite, A. F., Alves, V., Rodrigues, G. N., Tadonki, C., Eisenbeis, C., & De Melo, A. C. M. A. (2016, September). Autonomic provisioning, configuration, and management of inter-cloud environments based on a software product line engineering method. In 2016 *International Conference on Cloud and Autonomic Computing (ICCAC)* (pp. 72-83). IEEE.

Lotufo, R., She, S., Berger, T., Czarnecki, K., & Wąsowski, A. (2010, September). Evolution of the Linux kernel variability model. In *International Conference on Software Product Lines* (pp. 136-150). Springer, Berlin, Heidelberg.

Refactoring-Based Design of Reference Architecture Diseño Basado en Refactorización de la Arquitectura de Referencia.

Lu, X., Yin, J., He, G., Yu, H., & Xiong, N. N. (2019). An Architecture-Centric Development Approach for Service-Oriented Product Lines. *Journal of Internet Technology*, 20(4), 999-1012.

Luo, A., Zhang, M., Mao, Y., Kou, Y., & Zhang, X. (2019). A Structural Complexity Metric Method for Complex Information Systems. *J. Softw.*, 14(7), 332-339.

Maksuti, S., Tauber, M., & Delsing, J. (2019, October). Generic autonomic management as a service in a soa-based framework for industry 4.0. In *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society* (Vol. 1, pp. 5480-5485). IEEE.

Martínez-Fernández, S., Ayala, C. P., Franch, X., & Martins Marques, H. (2013, July). Benefits and drawbacks of reference architectures. *In European Conference on Software Architecture* (pp. 307-310). Springer, Berlin, Heidelberg.

Martinez, J., Ziadi, T., Bissyandé, T. F., Klein, J., & Le Traon, Y. (2015, July). Bottom-up adoption of software product lines: a generic and extensible approach. In *Proceedings of the 19th International Conference on Software Product Line* (pp. 101-110).

Mason, T. A., Thompson, W. W., Allen, D., Rogers, D., Gabram-Mendola, S., & Jacob Arriola, K. R. (2013). Evaluation of the Avon Foundation community education and outreach initiative Community Patient Navigation Program. *Health promotion practice*, 14(1), 105-112.

May, N. (2005, March). A survey of software architecture viewpoint models. In *Proceedings of the Sixth Australasian Workshop on Software and System Architectures* (pp. 13-24).

Medeiros, F. M., de Almeida, E. S., & de Lemos Meira, S. R. (2010, September). Designing a set of service-oriented systems as a software product line. In 2010 *Fourth Brazilian Symposium on Software Components, Architectures and Reuse* (pp. 70-79). IEEE.

Medeiros, F. M., Almeida, E. S. D., & Meira, S. R. (2010, September). SOPLE-DE: An approach to design service-oriented product line architectures. In *International Conference on Software Product Lines* (pp. 456-460). Springer, Berlin, Heidelberg.

Mistrik, I., Galster, M., & Maxim, B. R. (Eds.). (2019). Software Engineering for Variability Intensive Systems: Foundations and Applications. CRC Press.

Moens, H., Dhoedt, B., & De Turck, F. (2016, April). Management of customizable software-as-a-service in cloud and network environments. In *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium* (pp. 955-960). IEEE.

Moha, N., Palma, F., Nayrolles, M., Conseil, B. J., Guéhéneuc, Y. G., Baudry, B., & Jézéquel, J. M. (2012, November). Specification and detection of soa antipatterns. In *International Conference on Service-Oriented Computing* (pp. 1-16). Springer, Berlin, Heidelberg.

Mohabbati, B. (2013). Quality Model and Evaluation Method for Service-Oriented Software Product Line and Configurable Business Processes.

Mohabbati, B. (2013). Quality-aware Service-Oriented Software Product Lines: Feature-Driven Process Configuration and Optimization (Doctoral dissertation, Communication, Art & Technology: School of Interactive Arts and Technology).

Mohabbati, B., Asadi, M., Gašević, D., & Lee, J. (2014). Software product line engineering to develop variant-rich web services. In *Web Services Foundations* (pp. 535-562). Springer, New York, NY.

Mohamed, F., Mizouni, R., Abu-Matar, M., Al-Qutayri, M., & Whittle, J. (2017, August). An Integrated Platform for Dynamic Adaptation of Multi-Tenant Single Instance SaaS Applications. In *2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)* (pp. 257-264). IEEE.

Mohsin, A., & Janjua, N. K. (2018). A review and future directions of SOA-based software architecture modeling approaches for System of Systems. *Service Oriented Computing and Applications*, 12(3), 183-200.

Moon, M., Yeom, K., & Chae, H. S. (2005). An approach to developing domain requirements as a core asset based on commonality and variability analysis in a product line. *IEEE transactions on software engineering*, 31(7), 551-569.

Muchandi, V. (2007). Applying 4+ 1 view architecture with uml 2. *FCGSS White Paper*.

Murugesupillai, E., Mohabbati, B., & Gašević, D. (2011, August). A preliminary mapping study of approaches bridging software product lines and service-oriented architectures. In *Proceedings of the 15th International Software Product Line Conference, Volume 2* (pp. 1-8).

Nagamine, M., Nakajima, T., & Kuno, N. (2016, September). A case study of applying software product line engineering to the air conditioner domain. In *Proceedings of the 20th International Systems and Software Product Line Conference* (pp. 220-226).

Nakagawa, E. Y., Antonino, P. O., & Becker, M. (2011, August). Exploring the use of reference architectures in the development of product line artifacts. In *Proceedings of the 15th International Software Product Line Conference,* *Volume 2* (pp. 1-8).

Nakagawa, E. Y., Oliveira Antonino, P., & Becker, M. (2011, September). Reference architecture and product line architecture: A subtle but critical difference. In *European conference on software architecture* (pp. 207-211). Springer, Berlin, Heidelberg.

Nascimento, A. S., Rubira, C. M., Burrows, R., & Castor, F. (2013, September). A model-driven infrastructure for developing product line architectures using cvl. In *2013 VII Brazilian Symposium on Software Components, Architectures and Reuse* (pp. 119-128). IEEE.

Nguyen, T., Colman, A., & Han, J. (2011, October). A web services variability description language (WSVL) for business users oriented service customization. In *Web Information Systems Engineering–WISE 2011 and 2012 Workshops* (pp. 321-334). Springer, Berlin, Heidelberg.

Northrop, L., Clements, P., Bachmann, F., Bergey, J., Chastek, G., Cohen, S., ... & Little, R. (2007). A framework for software product line practice, version 5.0. *SEI.–2007–http://www. sei. cmu. edu/productlines/index. html*.

OpenEMR (2017) OpenEMR, *OpenEMR website*.

Ouali, S., Kraïem, N., Al-Khanjari, Z., & Baghdadi, Y. (2013, June). A model driven software product line process for developing applications. In *International Conference on Advanced Information Systems Engineering* (pp. 447-454). Springer, Berlin, Heidelberg.

Ouni, A., Gaikovina Kula, R., Kessentini, M., & Inoue, K. (2015, July). Web service antipatterns detection using genetic programming. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation* (pp. 1351-1358).

Palanivel, K., & Kuppuswami, S. (2011). Service-oriented reference architecture for personalized e-learning systems (SORAPES). *International Journal of Computer Applications*, 24(5), 35-44.

Pang, L. Y., Zhong, R. Y., & Huang, G. Q. (2013). Agent-based service-oriented architecture for heterogeneous data sources management in ubiquitous enterprise. In *Advances in Sustainable and Competitive Manufacturing Systems* (pp. 367-378). Springer, Heidelberg.

Parra, C., & Joya, D. (2015). SPLIT: An Automated Approach for Enterprise Product Line Adoption Through SOA. *J. Internet Serv. Inf. Secur.*, 5(1), 29-52.

Parra, C., Joya, D., Giral, L., & Infante, A. (2014, March). An SOA approach for automating software product line adoption. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing* (pp. 1231-1238).

Braga, R. T. V., Feloni, D., Pacini, K., & Gottardi, T. (2016, June). Aires: An architecture to improve software reuse. In *International Conference on Software Reuse* (pp. 231-246). Springer, Cham.

Bashari, M., Bagheri, E., & Du, W. (2016, June). Automated composition of service mashups through software product line engineering. In *International Conference on Software Reuse* (pp. 20-38). Springer, Cham.

Tiihonen, J., Raatikainen, M., Myllärniemi, V., & Männistö, T. (2016, June). Carrying ideas from knowledge-based configuration to software product lines. In *International Conference on Software Reuse* (pp. 55-62). Springer, Cham.

Kienzle, J., Mussbacher, G., Alam, O., Schöttle, M., Belloir, N., Collet, P., ... & Rumpe, B. (2016, June). VCU: the three dimensions of reuse. In *International Conference on Software Reuse* (pp. 122-137). Springer, Cham.

Pereira, A. F., David, J. M. N., Braga, R., & Campos, F. (2016, January). An architecture to enhance collaboration in scientific software product line. In *2016 49th Hawaii International Conference on System Sciences (HICSS)* (pp. 338-347). IEEE.

Perrouin, G., Amrani, M., Acher, M., Combemale, B., Legay, A., & Schobbens, P. Y. (2016, May). Featured model types: towards systematic reuse in modelling language engineering. In *Proceedings of the 8th International Workshop on Modeling in Software Engineering* (pp. 1-7).

Ponnalagu, K., Narendra, N. C., & Ghose, A. (2013, June). Formalizing service variability modeling in SOA-based solutions. In *International Conference on Advanced Information Systems Engineering* (pp. 402-416). Springer, Berlin, Heidelberg.

Ponnalagu, K., Narendra, N. C., Ghose, A., Chiktey, N., & Tamilselvam, S. (2013, December). Goal oriented variability modeling in service-based business processes. In *International Conference on Service-Oriented Computing* (pp. 499-506). Springer, Berlin, Heidelberg.

Pressman, R. S. (2005). Software engineering: a practitioner's approach. Palgrave macmillan.

Ribeiro, H. B. G., de Almeida, E. S., & de Lemos Meira, S. R. (2011, August). An approach for implementing core assets in service-oriented product lines. In

Proceedings of the 15th International Software Product Line Conference, Volume 2 (pp. 1-4).

Rincón, L., Munoz, E., Martinez, J. C., Pabón, M. C., & Alvarez, G. (2016, October). Extractive spl adoption applied into a small software company. In 2016 *XLII Latin American Computing Conference (CLEI)* (pp. 1-8). IEEE.

Ripon, S., Piash, M. M., Hossain, S. M. A., & Uddin, M. S. (2014). Semantic web based analysis of product line variant model. *International Journal of Computer and Electrical Engineering*, 6(1), 1-6.

dos Santos Rocha, R., Fantinato, M., Thom, L. H., & Eler, M. M. (2015). Dynamic product line for business process management. *Business Process Management Journal*.

Saleh, Z. I., Obeidat, R. A., & Khamayseh, Y. (2013). A Framework for an E-government Based on Service Oriented Architecture for Jordan. *International Journal of Information Engineering & Electronic Business,* 5(3).

Schmid, K. (2002, May). A comprehensive product line scoping approach and its validation. In *Proceedings of the 24th international conference on Software engineering* (pp. 593-603).

Schroeder, J., Holzner, D., Berger, C., Hoel, C. J., Laine, L., & Magnusson, A. (2015, May). Design and evaluation of a customizable multi-domain reference architecture on top of product lines of self-driving heavy vehicles-an industrial case study. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering* (Vol. 2, pp. 189-198). IEEE.

Serajzadeh, H., & Shams, F. (2012). An approach for discovering services for a service-oriented product line. *Global Journal on Technology*, 1.

da Silva, I. F., Neto, P. A. D. M. S., O'Leary, P., de Almeida, E. S., & de Lemos Meira, S. R. (2014). Software product line scoping and requirements engineering in a small and medium-sized enterprise: An industrial case study. *Journal of Systems and Software*, 88, 189-206.

da Silva, J. R. F., de Melo Filho, A. S., & Garcia, V. C. (2014). Toward a QoS Based Run-time Reconfiguration in Service-oriented Dynamic Software Product Lines. *In ICEIS (2)* (pp. 460-465).

Silva Marcolino, A., & Francine Barbosa, E. (2017, January). Towards a software product line architecture to build m-learning applications for the teaching of programming. In *Proceedings of the 50th Hawaii international conference on* *system sciences.*

Singh, A. P., & Tomar, P. (2016). Web service component reusability evaluation: a fuzzy multi-criteria approach. *International Journal of Information Technology and Computer Science (IJITCS),* 8(1), 40-47.

Sion, L., Van Landuyt, D., Joosen, W., & de Jong, G. (2016, September). Systematic quality trade-off support in the software product-line configuration process. In *Proceedings of the 20th International Systems and Software Product Line Conference* (pp. 164-173).

Soares, L. R., Schobbens, P. Y., do Carmo Machado, I., & de Almeida, E. S. (2018). Feature interaction in software product line engineering: A systematic mapping study. *Information and Software Technology*, 98, 44-58.

Souza Filho, E. D. D., Oliveira Cavalcanti, R. D., Neiva, D. F., Oliveira, T. H., Lisboa, L. B., Almeida, E. S. D., & Lemos Meira, S. R. D. (2008, September). Evaluating domain design approaches using systematic review. In *European Conference on Software Architecture* (pp. 50-65). Springer, Berlin, Heidelberg.

Hunter, M. G. (Ed.). (2009). Strategic Information Systems: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications. IGI Global.

Strîmbei, C., Dospinescu, O., Strainu, R. M., & Nistor, A. (2015). Software architectures-present and visions. *Informatica Economica,* 19(4), 13.

Suvelayutnan, U. (2018) 'Reusable Patient Navigation Software Architecture using Software Product Line Approach', *UTM Thesis*.

Taher, Y., & Sarker, A. (2015). Service Oriented Product Line: Analysis and Verification. *East West University*.

Tan, L., Lin, Y., & Ye, H. (2012, May). Modeling quality attributes in software product line architecture. In 2012 *Spring Congress on Engineering and Technology* (pp. 1-5). IEEE.

Tanhaei, M., Habibi, J., & Mirian-Hosseinabadi, S. H. (2016). A feature model based framework for refactoring software product line architecture. *Journal of Computer Science and Technology,* 31(5), 951-986.

Wang, H. B., Zhang, H., Han, Y. C., Fan, Z., Li, J. F., & Fan, Z. M. (2009). Zhonghua Er Bi Yan Hou Tou Jing Wai Ke Za Zhi. *Surgical management of jugular foramen tumors Department of Otorhinolaryngology Head and Neck Surgery, Shandong University Shandong Provincial Hospital, Eye Ear Nose & Throat*

Hospital, Shandong Provincial Hospital Group, Jinan, 250021, 70-6.

US Dept. of Defence / Office of the DoD CIO (2010) 'Reference Architecture Description', (June), p. 22.

Widadi, S., & Fajrin, H. R. (2021, February). Design of Conceptual Architecture of Maintenance Medical Facilities Information Systems. In *4th International Conference on Sustainable Innovation 2020–Technology, Engineering and Agriculture (ICoSITEA 2020)* (pp. 209-212). Atlantis Press.

Xing, Z., Xue, Y., & Jarzabek, S. (2013, May). A large scale linux-kernel based benchmark for feature location research. In *2013 35th International Conference on Software Engineering (ICSE)* (pp. 1311-1314). IEEE.

Yuan, X., & Tripathi, S. (2016, August). An approach of dynamically combining ontologies for interactive Requirements Elicitation. In 2016 *7th IEEE International Conference on Software Engineering and Service Science (ICSESS)* (pp. 11-16). IEEE.

Zaki, S. M., Mohamad, R., Abd Halim, S., & Ahmad, N. B. (2017). Variability Management in Software Product Lines Online Learning Applications. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9(3-5), 157-162.

Zhang, H., Lin, R., Zou, H., Yang, F., & Zhao, Y. (2013, June). The collaborative configuration of service-oriented product lines based on evolutionary approach. In *2013 IEEE International Conference on Services Computing* (pp. 751-752). IEEE.

Zhu, J., Cai, H., & Bu, F. (2013, December). Identifying Restful Web Services in Service-Oriented Software Product Line. In 2013 *International Conference on Cloud Computing and Big Data* (pp. 487-492). IEEE.

Zimmermann, A., Sandkuhl, K., Pretz, M., Falkenthal, M., Jugel, D., & Wissotzki, M. (2013, August). Towards an integrated service-oriented reference enterprise architecture. In *Proceedings of the 2013 International Workshop on Ecosystem Architectures* (pp. 26-30).

Zimmermann, O. (2017). Microservices tenets. Computer Science-Research and Development, 32(3), 301-310.

LIST OF PUBLICATIONS

Index

Basmmi, A. B. M. N., Abd Halim, S., & Saadon, N. A. (2020, July). Comparison of Web Services for Sentiment Analysis in Social Networking Sites. In *IOP Conference Series: Materials Science and Engineering* (Vol. 884, No. 1, p. 012063). IOP Publishing.

Basmmi, A. B. M. N., Abd Halim, S., & Jawawi, D. N. A. (2020). Trade-off Analysis in Identifying Potential Reusable Core Assets through Service Oriented Product Line. In International Journal of Advanced Research in Engineering and Technology (IJARET) (pp. 1215-1226).

Basmmi, A. B. M. N., Abd Halim, S., & Jawawi, D. N. A. (2021). Evaluating the Reusability of Core Assets for Service Oriented Product Line: Case Study of Patient Navigation Program (PNP). In International Medical Devices and Technology Conference (iMEDITEC 2021).

Non-Index

Basmmi, A. B. M. N., Abd Halim, S., & Jawawi, D. N. A. (2020). Business Process Feature Model in Identifying Potential Core Assets. In The 12th International Conference on Internet (ICONI 2020) (pp. 300-305).