

VARIABILITY MANAGEMENT FOR CLOUD-BASED SYSTEMS:

HOW FAR ARE WE?



Wesley K. G. Assunção

<https://wesleyklewerton.github.io/>
@wesleyklewerton

PRELIMINARIES

PhD: System variants > SPLs

Roberto E. Lopez-Herrejon · Jabier Martinez · Wesley Klewerton Guez Assunção · Tewfik Ziadi · Mathieu Acher · Silvia Vergilio *Editors*

Handbook
of Re-Engineering
Software
Intensive Systems
into Software
Produ

Information and Software Technology
Volume 117, January 2020, 106198

Automatic extraction of product line architecture and feature models from UML class diagram variants

Wesley K.G. Assunção^a, Silvia R. Vergilio^b, Roberto E. Lopez-Herrejon^c

Show more

The Journal of Systems and Software 210 (2024) 111969

Contents lists available at ScienceDirect

The Journal of Systems & Software

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

In practice

Variability debt in opportunistic reuse: A multi-project field study[☆]

Daniele Wolfart^a, Jabier Martinez^b, Wesley K.G. Assunção^{c,d,e}, Thelma E. Colanzi^e, Alexander Egyed^f

^aPPGComp, Western Paraná State University (UNIOESTE), Cascavel, Brazil
^bTecnalia, Basque Research and Technology Alliance (BRTA), Berio, Spain
^cCSC, North Carolina State University (NCSU), Raleigh, USA
^dOPUS, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brazil
^eDSN, State University of Maringá (UEM), Maringá, Brazil
^fISSE, Johannes Kepler University Linz (JKU), Linz, Austria

Postdoc: Legacy systems > Microservices

Empirical Software Engineering (2022) 27: 51
<https://doi.org/10.1007/s10664-021-10049-7>



Analysis of a many-objective optimization approach for identifying microservices from legacy systems

Wesley K. G. Assunção
Alessandro Garcia¹ · Ju

2019 IEEE/ACM Joint 7th International Workshop on Conducting Empirical Studies in Industry (CESI) and 6th International Workshop on Software Engineering Research and Industrial Practice (SER&IP)

Accepted: 1 September 2021 /
© The Author(s), under exclus

Analysis of the Criteria Adopted in Industry to Extract Microservices

Luiz Carvalho^{*}, Alessandro Garcia^{*}, Wesley K. G. Assunção[†], Rafael de Mello^{*}, Maria Julia de Lima[‡]
^{*}Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brazil
{lmcvalho, afgarcia, rmaiani}@inf.puc-rio.br

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 50, NO. 3, MARCH 2024

651

On the Usefulness of Automatically Generated Microservice Architectures

Luiz Carvalho, Thelma Elita Colanzi[⊕], Wesley K. G. Assunção[⊕], Alessandro Garcia[⊕], Juliana Alves Pereira[⊕], Marcos Kalinowski[⊕], Member, IEEE, Rafael Maiani de Mello[⊕], Maria Julia de Lima[⊕], and Carlos Lucena[⊕]

Abstract—The modernization of monolithic legacy systems with microservices has been a trend in recent years. As part of this modernization, identifying microservice candidates starting from legacy code is challenging, as maintainers may consider many criteria simultaneously. Multi-objective search-based approaches represent a promising state-of-the-art solution to

previous studies are, in fact, considered by maintainers; and (iii) the maintainer profiles, i.e., the preferred granularity for microservice, highly affect design decisions. Finally, we observed the maintainers needed little effort in adjusting the automatically identified microservices to make them adoptable. In addition to indicating a promising potential of search-based approaches to

PRELIMINARIES

2019 IEEE/ACM Joint 7th International Workshop on Conducting Empirical Studies in Industry (CESI) and 6th International Workshop on Software Engineering Research and Industrial Practice (SER&IP)

Analysis of the Criteria Adopted in Industry to Extract Microservices

Luiz Carvalho*, Alessandro Garcia*, Wesley K. G. Assunção†, Rafael de Mello*, Maria Julia de Lima‡

*Pontifical Catholic University of Rio de Janeiro (PUC-Rio). Rio de Janeiro, Brazil

{lmcavalho, afgarcia, rmaiani}@inf.puc-rio.br

†Federal University of Technology - Paraná (UTFPR). Toledo, Brazil

wesleyk@utfpr.edu.br

‡Teccraf Institute, Pontifical Catholic University of Rio de Janeiro (PUC-Rio). Rio de Janeiro, Brazil

Reuse opportunities are considered on microservice extraction, albeit not unanimously. Reuse also achieved a median of 4 in the responses. One participant mentioned the practice of reuse “*is indeed a key driving factor for the migration to microservices architecture; it is important to promote the reuse of the extracted microservice by other systems*”. However, the relevance of reuse was not unanimous.

SHORT-PAPER



Extraction of Configurable and Reusable Microservices from Legacy Systems: An Exploratory Study

Authors: Luiz Carvalho, Alessandro Garcia, Wesley K. G. Assunção, Rodrigo Bonifácio, Leonardo P. Tizzel,

Thelma Flita Colanzi [Authors info & Claims](#)

SPLC '19: Proceedings of the 23rd International Systems and Software Product Line Conference - Volume A • September 2019 • Pages 26–31 • <https://doi.org/10.1145/3336294.3336319>

Published: 09 September 2019 [Publication History](#)

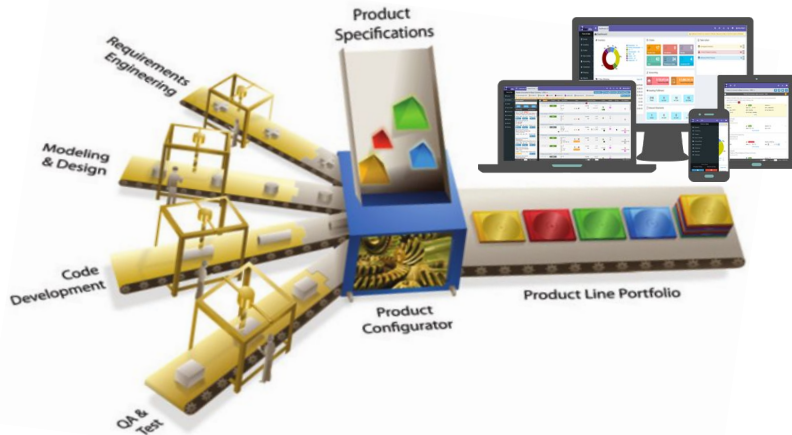


20 583

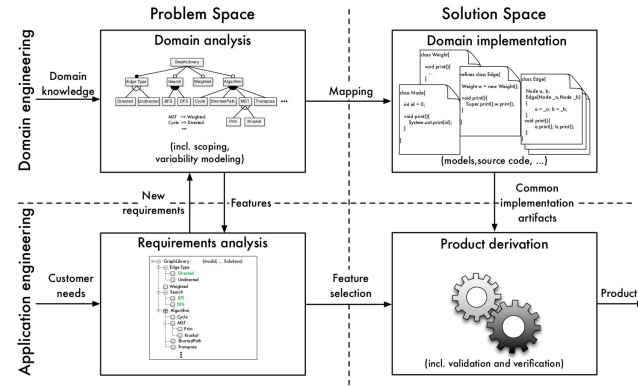


FEATURE-ORIENTED SW DEVELOPMENT

Variability Management



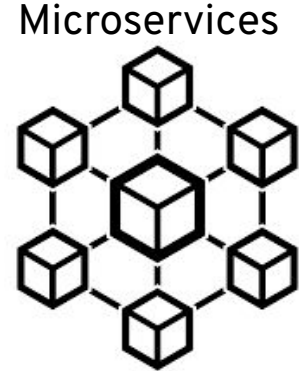
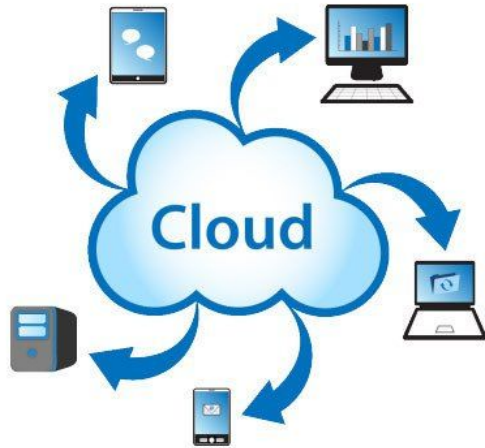
Features as building blocks



Feature: A prominent or distinctive **user-visible aspect, quality, or characteristic** of a software system or systems.

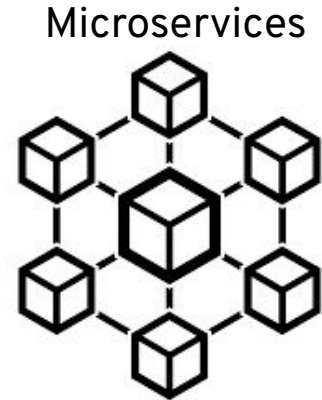
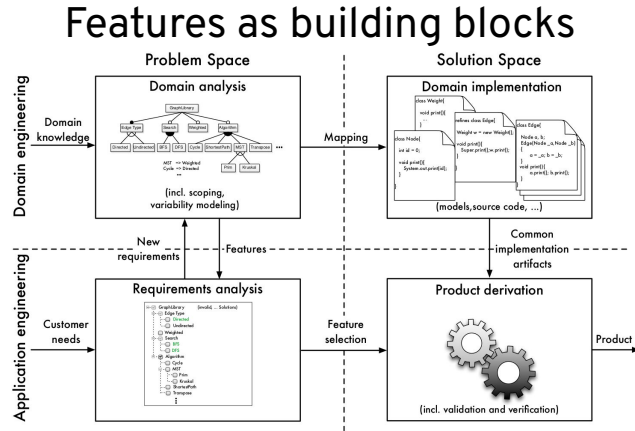
A feature is a **unit of functionality** of a software system that satisfies a requirement, represents a design decision, and provides a **potential configuration option**.

CLOUD-BASED SOFTWARE



With a microservices architecture, an application is built as **independent components** that run each application process as a service. These services communicate via a **well-defined interface** using lightweight APIs. Services are built for **business capabilities** and each service performs a **single function**.

FEATURES AND MICROSERVICES



- user-visible aspect, quality, or characteristic
- unit of functionality
- potential configuration option
- independent components
- well-defined interface
- business capabilities
- single function

VARIABILITY MANAGEMENT + MICROSERVICES

RESEARCH-ARTICLE



Variability management meets microservices: six challenges of re-engineering microservice-based webshops

Authors: Wesley K. G. Assunção, Jacob Krüger, Willian D. F. Mendonça [Authors Info & Claims](#)

SPLC '20: Proceedings of the 24th ACM Conference on Systems and Software Product Line: Volume A - Volume A • October 2020 • Article No.: 22 • Pages 1–6 • <https://doi.org/10.1145/3382025.3414942>

Published: 19 October 2020 [Publication History](#)



16 394



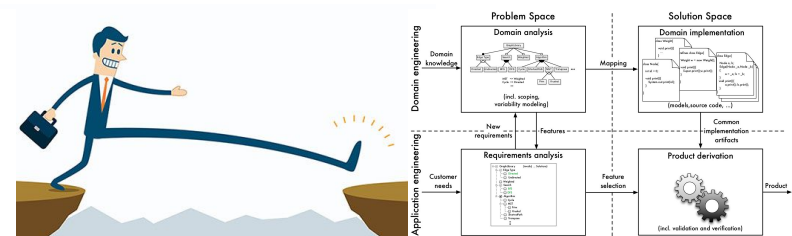
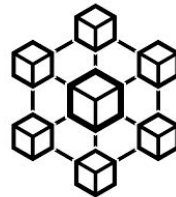
ABSTRACT

A microservice implements a small unit of functionality that it provides through a network using lightweight protocols. So, microservices can be combined to fulfill tasks and implement features of a larger software system—resembling a variability mechanism in the context of a software product line (SPL). Microservices and SPLs have similar goals, namely facilitating reuse and customizing, but they are usually employed in different contexts. Any developer who has access to the network can provide a microservice for any task, while SPLs are usually intended to implement features of a specific domain. Due to their different concepts, using microservices to implement an SPL or adopting SPL practices (e.g., variability management) for microservices is a challenging cross-area



both techniques can complement each other, and thus tackling this or organizations that employ either technique. In this paper, we advancing in this direction, and sketch six concrete challenges to feature identification, (2) variability modeling, (3) variable microservice ability, (5) deep customization, and (6) re-engineering an SPL. We reive as a starting point for future research in this cross-area research concepts of one area are reinvented in the other.

1. Feature Identification and Mapping
2. Variability Modeling
3. Microservice-Based SPL Architecture
4. Microservice Interchanging
5. Deep Customization of Microservices
6. Re-Engineering a Microservice-Based SPL



VM + MS CHALLENGE SOLUTIONS

RESEARCH-ARTICLE



Can microservice-based online-retailers be used as an SPL?: a study of six reference architectures

Authors: Benjamin Benni, Sébastien Mosser, Jean-Philippe Caissy, Yann-Gaël Guéhéneuc [Authors Info & Claims](#)

SPLC '20: Proceedings of the 24th ACM Conference on Systems and Software Product Line: Volume A - Volume A • October 2020 • Article No.: 24 • Pages 1–6 • <https://doi.org/10.1145/3382025.3414979>

Published: 19 October 2020 [Publication History](#)



4 110



ABSTRACT

Microservices are deployable software artifacts that combine a set of business features and expose them to other microservices. Ideally, the reuse and interchanging of microservices should be easy as they are supposed to be independent of each other, both conceptually and technologically. Selecting a service to fulfill a given feature (*e.g.*, managing a cart in a website) recalls the way *Software Product Lines* (SPL) allow variability. However, in practice, interchanging microservices requires knowing the features that the services propose, how they communicate with other services and their types. In this work, we propose to analyze service dependencies as feature dependencies, at the feature, structural, technological, and versioning level, to assess the interchangeability of services. We analyze six community-selected use-cases and report that services are non-interchangeable systematically.

RESEARCH-ARTICLE



Variability management: re-engineering microservices with delta-oriented software product lines

Authors: Maya R.A. Setyautami, Hafiyyan S. Fadhillah, Daya Adianto, Ichlasul Affan, Ade Azurat [Authors Info & Claims](#)

SPLC '20: Proceedings of the 24th ACM Conference on Systems and Software Product Line: Volume A - Volume A • October 2020 • Article No.: 23 • Pages 1–6 • <https://doi.org/10.1145/3382025.3414981>

Published: 19 October 2020 [Publication History](#)



3 159



ABSTRACT

Combining microservices and software product line engineering (SPLE) is a challenge in variability management. This paper proposes a solution to that challenge by re-engineering existing webshop systems into a product line application. We first perform feature identification to analyze the features of subject systems. We introduce a mechanism that models the variability and designs a software product line architecture based on existing features. We use a UML diagram with the UML-DOP profile to abstract microservice variability in SPLE. Then, a transformation into a product line application is conducted to generate running applications based on selected features. We utilize a preliminary framework of microservice variability based on delta-oriented programming.

VM FOR CLOUD-BASED SYSTEMS

- How far are we?
 - Do we (this community) have the solutions for dealing with variability in cloud-based systems?
 - Design methods, tools, variability mechanisms, etc.
 - Can the industry benefit of having cloud-based software product lines?
 - E.g., reuse, mass customization, reduce cost, etc.
 - Are there new challenges due to the cloud environment (large space, distributed, asynchronous)?
 - E.g. chaos engineering
 - What are the best practices (what to teach)?

FEATURE FLAGS

Why You Should Use Feature Flags in a Microservices Architecture

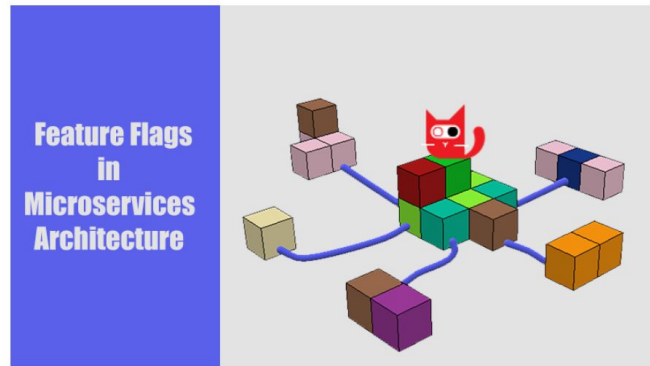
October 20, 2023 · 7 min read



Emil Kovačević

Code hard, debug harder.

Microservices Architecture is a revolutionary approach to software development that empowers applications to scale and evolve with unparalleled flexibility. But: How do feature flags help in this dynamic landscape of microservices?



Benefits of Using Feature Flags in a Microservices Architecture

- Independent Development of Features
- Risk Reduction and Releases Control
- Enhanced User Experience and Testing Capabilities
- ? Customization ?

What are Feature Flags?

Feature flags, also known as feature toggles, are a technique used in software development to enable or disable certain features in an application. By using feature flags, developers can easily turn features on or off at runtime without code re-deployment.

FEATURE FLAGS

SHORT-PAPER OPEN ACCESS

Capture the Feature Flag: Detecting Feature Flags in Open-Source

Authors: Jens Meinicke, Juan Hoyos, Bogdan Vasilescu, Christian Kästner [Authors Info & Claims](#)

MSR '20: Proceedings of the 17th International Conference on Mining Software Repositories • June 2020 • Pages 169–173
• <https://doi.org/10.1145/3379597.3387463>

Published: 18 September 2020 [Publication History](#) [Check for updates](#)

Empirical Software Engineering (2021) 26: 1
<https://doi.org/10.1007/s10664-020-09901-z>

Software development with feature toggles: practices used by practitioners

Rezvan Mahdavi-Hezaveh¹ · Jacob Dremann¹ · Laurie Williams¹

Accepted: 23 September 2020 / Published online: 8 January 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

Abstract
Background Using feature toggles is a technique that allows developers to either turn a feature on or off with a variable in a conditional statement. Feature toggles are increasingly used by software companies to facilitate continuous integration and continuous delivery. However, using feature toggles inappropriately may cause problems which can have a severe impact, such as code complexity, dead code, and system failure. For example, the erroneous

RESEARCH-ARTICLE PUBLIC ACCESS

Exploring differences and commonalities between feature flags and configuration options

Authors: Jens Meinicke, Chu-Pan Wong, Bogdan Vasilescu, Christian Kästner [Authors Info & Claims](#)

ICSE-SEIP '20: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Practice • June 2020 • Pages 233–242 • <https://doi.org/10.1145/3377813.3381366>

Published: 18 September 2020 [Publication History](#) [Check for updates](#)

20 515

RESEARCH-ARTICLE

From feature models to feature toggles in practice

Authors: Jean-Marc Jézéquel, Jörg Kienzle, Mathieu Acher [Authors Info & Claims](#)

SPLC '22: Proceedings of the 26th ACM International Systems and Software Product Line Conference - Volume A • September 2022 • Pages 234–244 • <https://doi.org/10.1145/3546932.3547009>

Published: 12 September 2022 [Publication History](#) [Check for updates](#)

1 131

ABSTRACT

Feature Toggles (often also referred to as Feature Flags) are a powerful technique, providing an alternative to maintaining multiple feature branches in source code. A condition within the code enables or disables a feature at runtime, hence providing a kind of runtime variability resolution. Several works have already identified the proximity of this concept with the notion of *Feature* found in *Software Product Lines*. In this paper, we propose to go one step further in unifying these concepts to provide a seamless transition between design time and runtime variability resolutions. We propose to model all the variability using a feature model. Then this feature model can be partially resolved at design time (yielding an incomplete product derivation), the unresolved variability being used to generate feature toggles that can be enabled/disabled at runtime. We first demonstrate these ideas on the toy example of the *Expression Product Line*, and then show how it can scale to build a configurable authentication system, where a partially resolved feature model can interface with popular feature toggle frameworks such as *Togglz*.

CALL TO ACTION ?!

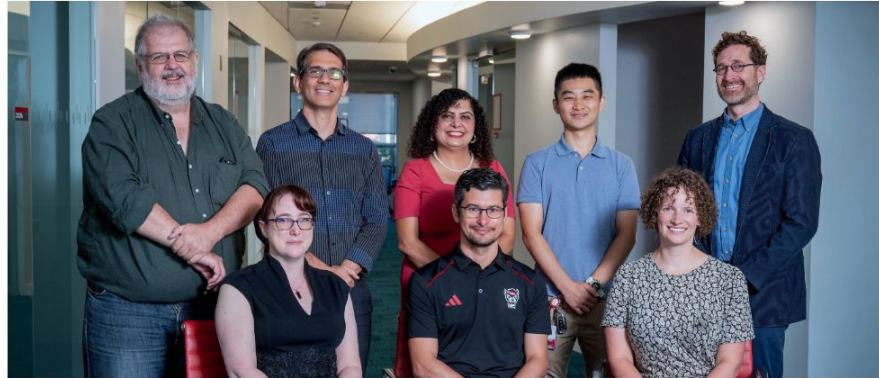
- Establish(or define) the role of reuse/customization in cloud-based system
- Collaboration: cross-area research between cloud + SPL engineering
- What else?



SOFTWARE ENGINEERING AT NCSU

NC STATE

[Home](#) [Contact](#)



Software Engineering at NCSU

Accelerate your SE career, in industry, in research.



MAKING SOFTWARE DEVELOPMENT BETTER

RESEARCH OVERVIEW



Wesley K. G. Assunção

<https://wesleyklewerton.github.io/>
@wesleyklewerton

