Karlsruhe Institute of Technology

Software Design and Quality

From Monolith to Microservices: A not yet defined Approach

Bachelor's Thesis of

Niko Benkler

at the Department of Informatics
Institute for Program Structures and Data Organization (IPD)

Reviewer:	Prof. Dr. Ralf H. Reussner
Second reviewer:	Prof. B
Advisor:	Dr. Robert Heinrich

xx. Month 20XX – xx. Month 20XX

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

PLACE, DATE

. . . .

(Niko Benkler)

Bla Sbtrakt

Abstract

English abstract.

Zusammenfassung

Deutsche Zusammenfassung

Contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation

1.2. Problem Statement

1.3. Challenges

2. CoCoME

2.1. Introduction to CoCoME

[7]

3. State of the Art

3.1. Literature Review

Link	Titel	Author (Year)	Origin	Search String
[6]	Extraction of Microservices from Monolithic Software Architectures	G. Matzlami et. al. (2017)	Google Scholar	*microservice identification*
[1]	Object-Aware Identification of Microservice	M. J. Amiri (2018)	IEEE	*identification microservices*
[2]	Microservices Identification Through Interface Analysis	L. Baresi et. al. (2017)	google scholar	*microservice identification*
[9]	Identifying Microservices Using Functional Decomposition	S. Tyszberowicz et. al. (2018)	*provided*	*n/a*
[8]	Partitioning Microservices: A Domain Engineering Approach	I. J. Munezero et. al. (2018)	IEEE	*identify microservices*
[3]	From Monolith to Microservices: A Dataflow-Driven Approach	R.Chen et. al	IEEE	monolith to microservice
[4]	Function-Splitting Heuristics for Discovery of Microservices in Enterprise Systems	A. De Alwis et. al. (2018)	Google Scholar	identify microservices
[5]	Service Cutter: A Systematic Approach to Service Decomposition	M. Gysel et. al. (2016)	[2]	*n/a*

3.2. Comparison and applicability of the approaches

* bedeuted INFO
\+ beudeted PRO
\- bedeuted CONTRA

3.2.1. Extraction of Microservices from Monolithic Software Architectures

* informal migration patterns exists. Lack of Formal Models
* small and recent body of work on how to migrate monolith to MS
* construct graph, process by clustering algorithm
* references Service Cutter (Pros and Cons)
* 2 phases: Construction (monolith to graph), clustering (decompose graph to cluster)
* starts with code base/repo from VCS
* each class is a node, edges have weights according to coupling strategy (classes that are not coupled are discarded)
* Logical Coupling Strategy(LC): Single Responsibility principle (Software has only one reason to change), enforce strong module boundaries (concept of MS) −> developers only make changes to the module (found in Change History, Class Files changed together belong togehter) ==> Weight is : for each pair of class look how often they changed together
* Semantic Coupling Strategy(SC): each MS correspond to one bounded context (DDD) from domain, examine contents/semantics of source code, term-frequency invese-documents-frequency method (tf-idf), compute relation of two classes regarding domain concepts
* tf-idf: Compute scalar vector for each class and compute cosine similarity between pairwise distinct classes
* tf-idf: Tokenize class, set of words, filter stop words, compare two classes regarding their common words with tf-idf formula
* Main Concern: Well organized teams, cross-functional but also reduce communication overhead to external teams while maximize internal
* Contributor Coupling: team/orga info used to recover relationship among sw artifacts ==> Ownership architecture read from VSC history by identifying how many developers worked on the same pair of classes (weight!)
——————————

+ algorithmic recommendation of ms candidates implemented in web-based prototype
+ unites traditional decomposition techniques and microservice extraction approaches/design principles
+ algorithm uses 3 different coupling strategies: Can be combined for better results

———————————

- rely on (meta-)data extracted from codebase
- needs VCS (proper change history)
- 2 (independent) changes in one commit destroy SRP
- SW must have gone through evolution process for LC
- Naming of class, methods, attributes needs to reflect domain language to make tf-idf possible
- Contributor Coupling: requires more developers

4. Solution Overview

5. Evaluation Planning

5.1. Applicability to CoCoME

5.2. Comparison to Functional Decomposition Approach

6. Timetable

6.1. Milestones

Bibliography

[1] M. J. Amiri. “Object-Aware Identification of Microservices”. In: (July 2018), pp. 253–256. ISSN: 2474-2473. DOI: 10.1109/SCC.2018.00042.

[2] Luciano Baresi, Martin Garriga, and Alan De Renzis. “Microservices Identification Through Interface Analysis”. In: (2017). Ed. by Flavio De Paoli, Stefan Schulte, and Einar Broch Johnsen, pp. 19–33.

[3] R. Chen, S. Li, and Z. Li. “From Monolith to Microservices: A Dataflow-Driven Approach”. In: (Dec. 2017), pp. 466–475. DOI: 10.1109/APSEC.2017.53.

[4] Adambarage Anuruddha Chathuranga De Alwis et al. “Function-Splitting Heuristics for Discovery of Microservices in Enterprise Systems”. In: (2018). Ed. by Claus Pahl et al., pp. 37–53.

[5] Michael Gysel et al. “Service Cutter: A Systematic Approach to Service Decomposition”. In: (2016). Ed. by Marco Aiello et al., pp. 185–200.

[6] G. Mazlami, J. Cito, and P. Leitner. “Extraction of Microservices from Monolithic Software Architectures”. In: (June 2017), pp. 524–531.

[7] Frank Mittelbach. “How to influence the position of float environments like figure and table in LaTeX?” In: *TUGboat* 35 (2014), pp. 248–254. URL: https://www.latex-project.org/publications/tb111mitt-float.pdf.

[8] I. J. Munezero et al. “Partitioning Microservices: A Domain Engineering Approach”. In: (May 2018), pp. 43–49.

[9] Shmuel Tyszberowicz et al. “Identifying Microservices Using Functional Decomposition”. In: (2018). Ed. by Xinyu Feng, Markus Müller-Olm, and Zijiang Yang, pp. 50–65.

A. Appendix

A.1. First Appendix Section

Figure A.1.: A figure

...