

A Rule-based Toolkit for Automated Generation of Microservices Architecture

K. J. P. G. Perera
178045U

Thesis submitted in partial fulfillment of the requirements for the Degree of Master of
Science by Research

Department of Computer Science & Engineering
University of Moratuwa
Sri Lanka

July 2019

Declaration

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

Date:

Name: K. J. P. G. Perera

The above candidate has carried out research for the MSc Dissertation under my supervision.

Signature of the supervisor:.....

Date:

Name of the supervisor: Dr. G. I. U. S. Perera

Abstract

Software applications play a critical role in current business world; hence it is necessary to design a quality and a sound architecture which facilitates it to become a scalable, extensible and highly available solution. In terms of designing and developing software applications, software engineering community has started shifting towards serverless-microservices instead of building large monolith applications.

It requires high experience and expertise to understand each business scenario along with considering non-functional requirements too to design a high-level software architecture which would be the ground point for a software application. The traditional manual process of doing the above is tedious as well as can be error prone when architecture designing is done without proper experience and expertise, which could eventually degrade the quality of the software application.

We introduce TheArchitect, a rule-based system providing a tool-based support in order to design the best fitted high-level architecture containing serverless microservices, preserving the identified non-functional requirements too, for any given application. Furthermore, TheArchitect provides the ability to a software engineer also to generate a high-quality high-level architecture even without an experienced software architect. Considering the increasing tendency within the software engineering community to move away from monolith application development towards microservices-serverless based application development, TheArchitect has also been developed focusing on generating high-level application architecture designs based on serverless-microservices.

TheArchitect was used to generate architecture designs for restaurant management domain. System generated architecture designs for two real world applications and how experienced architects' modifications are incorporated as modified rules for future designs have been discussed. Further a performance evaluation is conducted on TheArchitect to provide an analysis on the time it takes to process the requirements and design the architecture for various real-world systems along with an industry user study is presented evaluating the usability of TheArchitect.

Keywords: Software Architecture, Microservices Architecture, Serverless Architecture, Domain Driven Design, Architecture Evaluation

Acknowledgments

I would like to express my sincere gratitude to my mentor Dr. Indika Perera and my supervisors for the guidance, support provided throughout my research. I would not be able to achieve all what I have achieved without your incredible mentorship and advices from the beginning.

I would like to thank all the staff members of the Department of Computer Science and Engineering for their continuous and generous support provided for me.

I am also grateful for all the software architects and technical leads within the industry who helped in evaluating the developed tool (TheArchitect), as well as spent their valuable time in helping out in terms of fine tuning TheArchitect.

Special thanks to my family and my loved ones for your encouragement and understanding throughout the past two years.

Table of Contents

Declaration.....	viii
Abstract.....	viii
Acknowledgement	iv
Table of Contents.....	v
List of Tables	vi
List of Figures.....	viii
List of Abbreviations	x
1. Introduction.....	1
1.1 Background.....	1
1.2 Motivation for the Research.....	2
1.3 Research Statement.....	2
1.4 Objectives of the Research.....	4
1.5 Research Methodology	4
1.6 Contributions	5
1.7 Organization of the Thesis	5
2. Literature Review.....	7
2.1 Microservices Architecture.....	7
2.2 Serverless Architecture	13
2.3 Rule-Based Systems.....	15
2.4 Domain Specific Software Architecture (DSSA)	16
2.5 Backend for Frontend (BFF).....	16
2.6 Service Oriented Architecture (SOA) and Micro-services	17
2.7 Architecture Description Languages (ADLs) – Model Software Architecture Based Development.....	17
2.8 Formal Process for Software Architecture Improvement	22
2.9 Scenario-Based Software Architecture Evaluation.....	24
3. Methodology.....	26
3.1 Input Wizard	26
3.2 Data Processor	28

3.3 Architecture Generator.....	28
3.4 Visual Representation	29
4. Implementation	31
4.1 Knowledge Base	31
4.2 Architecture Generation Algorithm	32
4.3 Visual Representation	40
5. Experiments	41
5.1 Experiment Design.....	41
5.2 Experiment Type A.....	42
5.3 Experiment Type B	45
5.4 Experiment Type C	45
6. Results and Discussion	46
6.1 Order Receive Application	46
6.2 Inventory Management Application	50
6.3 Performance Evaluation.....	54
6.4 User Study Statistics	55
7. Conclusion	57
7.1 Study Limitations.....	58
7.2 Future Directions	59
8. References.....	60

List of Tables

Table 1: ADL Facilitation for Modeling Components [14].....	18
Table 2: ADL Facilitation for Modeling Connectors [14].....	19
Table 3: ADL Facilitation for Modeling Architectural Configurations [14].....	20
Table 4: ADL Tool-Based Support [14]	21
Table 5: Weights of evaluation metrics – Restaurant management domain.....	44
Table 6: Order Receive Application - Services based metrics evaluation on system generated architecture.....	50
Table 7: Order Receive Application - Services based metrics evaluation on user modifications accepted architecture.....	50
Table 8: Inventory Management Application - Services based metrics evaluation on system generated architecture	54
Table 9: Inventory Management Application - Services based metrics evaluation on user modifications accepted architecture.....	54
Table 10: Number of high-level system epics vs processing times	54

List of Figures

Figure 1: Monolith System vs Microservice architecture-based systems [5]	8
Figure 2: Work Organization for Monolith vs Microservices architecture systems [5]	9
Figure 3: Basic Build Pipeline [5]	12
Figure 4: One BFF per user interface	16
Figure 5: Software Architecture Improvement Cycle [9]	22
Figure 6: High-level architecture of TheArchitect.....	26
Figure 7: TheArchitect - Base rule set for any application domain.....	27
Figure 8: Commission Calculator Application - High-level architecture design diagram	29
Figure 9: Commission Calculator - Serverless technology analysis.....	30
Figure 10: Commission Calculator - Updated rules set for finance domain.....	31
Figure 11: Algorithm 1 – High-level architecture generation algorithm.....	32
Figure 12: Algorithm 2 – Path builder algorithm	33
Figure 13: Algorithm 3 – Clear path algorithm	33
Figure 14: Flow chart – High-level architecture generation algorithm based on API availability.....	36
Figure 15: Flow chart - High-level architecture generation algorithm based on database availability.....	37
Figure 16: Flow chart - Incorporating domain specific rules set in generating high-level architecture.....	38
Figure 17: Order Receive Application - System generated high-level architecture design diagram	47
Figure 18: Order Receive Application - High-level architecture design diagram with accepted user modifications	48
Figure 19: Order Receive Application - Component based metrics evaluation on system generated architecture diagram	49
Figure 20: Order Receive Application - Component based metrics evaluation on user modifications accepted architecture diagram.....	49
Figure 21: Inventory Management Application - System generated high-level architecture design diagram.....	51
Figure 22: Inventory Management Application - High-level architecture design diagram with accepted user modifications.....	52

Figure 23: Inventory Management Application - Component based metrics evaluation for system generated architecture diagram.....	53
Figure 24: Inventory Management Application - Component based metrics evaluation on user modifications accepted architecture diagram.....	53
Figure 25: Number of problem elements against the number of modifications for the rule set	55
Figure 26: User preference statistics for TheArchitect.....	56

List of Abbreviations

HTTP	Hyper Text Transfer Protocol
API	Application Programming Interface
AWS	Amazon Web Services
ADL	Architecture Description Language
DDD	Domain Driven Design
DSSA	Domain Specific Software Architecture
BAAS	Backend-as-a-Service
MBAAS	Mobile Backend-as-a-Service
FAAS	Function-as-a-Service
BFF	Backend-for-Frontend
SOA	Service Oriented Architecture