Service Oriented Software Framework for Rapid Application Development

Rubasinghe N.

158773F

Faculty of Information Technology

University of Moratuwa

June 2018

Service Oriented Software Framework for Rapid Application Development

Rubasinghe N

158750H

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the fulfilment of the requirements of Degree of Master of Science in Information Technology

June 2018

Declaration

We declare that this thesis is our own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Nishantha Rubasinghe ……………………………

Date: 2018-06-27 Signature of Student

Supervised by

Mr Chaman Wijesiriwardana
Senior Lecturer
Faculty of Information Technology
University of Moratuwa

…………………………………..

Date: 2018-06-27 Signature of Supervisor

Acknowledgements

I would like to express my gratitude to my supervisor Mr Chaman Wijesiriwardana, Senior Lecturer at the University of Moratuwa, Sri Lanka whose expertise. understanding. And patience added considerably to my research experience. I appreciate his vast knowledge and skill in many areas and his assistance in writing reports.

I would like to thank the other lecturers of University of Moratuwa, Sri Lanka. especially, Prof. Asoka Karunananda for the knowledge and assistance they provided at all levels of the research project.

I would also like to thank all the batch mates of the M.Sc. in IT degree program who gave their valuable feedback to improve the results of the research and my family for the support they provided me through my entire life and. I must acknowledge my wife Shashikala, without whose love, encouragement and editing assistance I would not have finished this thesis.

Abstract

This paper proposes a service-oriented rapid application development framework for application developments. It provides nine commonly use business patterns, the application under development with representations for the data model, business logic, API layer, user interfaces. Which are presented together with a diagram showing the precedence for checking the convenience of their utilization? The paper describes the layers, techniques, code generation, API testing tool and explores possible future extensions.

Many aspects of software development projects can be improved with an information architecture that provides a framework to automate the design, development, testing processes using pre-defined patterns and automate processes to ensure that the process is completed on time, within budget, and reliably with quality assurance. In this paper, we address a critical aspect of the time-consuming gaps that exists between the process of software design, implementation, and testing. The proposed pattern-based software rapid development framework is intended to fill these gaps and provide a more integrated approach in the development of a software system. We introduce the capability to orchestrate new integration technology that leverages recent advances in Web Services, including service-oriented architecture. This new approach can lead to significant improvement in software productivity and quality by providing a platform that bridges the gap between stakeholders and software engineers, as well as providing a reliable integration mechanism using pattern-based service-oriented approach to manage rapid changes in domain knowledge.

Table of Contents

Page

Declaration **i**

Acknowledgements **ii**

Abstract **iii**

Table of Contents **iv**

List of Figures **v**

List of Tables **vi**

Chapter 1 – Introduction **1**

1.1 Introduction 1

1.2 Background and motivation 1

1.3 Problem Definition 2

1.4 Aim & Objectives 2

1.5 Summary 3

Chapter 2 – Rapid Development Approaches **4**

2.1 Introduction 4

2.2 Rapid Application Development 4

2.3 Software Development Challenges 5

2.4 Software Patterns 6

2.5 Latest RAD Frameworks & Approaches 7

2.6 Code Generations 8

2.6 API Testing Tools 8

2.7 Summary 9

Chapter 3 – Approach **10**

3.1 Introduction 10

3.2 Hypothesis 10

3.3 Major Phases of the Proposed Framework 10

3.3.1 Identifying Common Business Patterns 10

3.3.2 Code Generation Module 11

3.3.3 Automate API Testing Tool 12

3.4 Advantages & Limitations 13

3.5 Summary 15

Chapter 4 – Design & Implementation 16

4.1 Introduction 16

4.2 Over roll Framework Design 16

4.3 Top Level Architecture 17

4.4 Analyzing & identifying patterns 18

4.4.1 Pattern 1 18

4.4.2 Pattern 2 20

4.4.3 Pattern 3 22

4.4.4 Pattern 4 25

4.4.5 Pattern 5 28

4.4.6 Pattern 6 30

4.4.7 Pattern 7 33

4.4.8 Pattern 8 37

4.4.9 Pattern 9 40

4.5 Common Functions of an Entity Model 43

4.6 Code Generation 43

4.7 API Testing Tool 45

4.8 Summary 47

Chapter 5 – Evaluation 48

5.1 Introduction 46

5.2 Case Study Evaluation 46

5.2.1 Case Study 1 48

5.2.2 Case Study 2 50

5.3 Case Study Outcomes 52

5.4 Summary 52

Chapter 6 – Conclusion **53**

6.1 Introduction 53

6.2 Conclusion 53

6.3 Further Works 53

6.4 Summary 54

Chapter 7 – Reference 55

List of Figures

Page

Figure 3.5.2.1: Workflow of the code generation 11

Figure 4.3.1: System Architecture 17

Figure 4.4.1.1: Data Model of Pattern 1 18

Figure 4.4.1.2: Basic UI Input form of Pattern 1 19

Figure 4.4.2.1: Data Model of Pattern 2 20

Figure 4.4.2.2: Basic UI Input form of Pattern 2 21

Figure 4.4.3.1: Data Model of Pattern 3 22

Figure 4.4.3.2: Basic UI Input form of Pattern 3 24

Figure 5.4.4.1: Sample Data Model of Pattern 4 25

Figure 4.4.4.2: Basic UI Input form of Pattern 4 27

Figure 4.4.5.1: Data Model of Pattern 5 28

Figure 4.4.5.2: Basic UI Input form of Pattern 5 29

Figure 4.4.6.1: Data Model of Pattern 6 30

Figure 4.4.6.2: Basic UI Input form of Pattern 6 32

Figure 4.4.6.3: simplest UI Input form of Pattern 6 33

Figure 4.4.7.1: Data Model of Pattern 7 34

Figure 4.4.7.2: Basic UI Input form of Pattern 7 36

Figure 4.4.8.1: Data Model of Pattern 8 37

Figure 4.4.8.2: Basic UI Input form of Pattern 8 39

Figure 4.4.9.1: Data Model of Pattern 9 40

Figure 4.4.9.2: Basic UI Input form of Pattern 9 42

Figure 6.5.3: A screenshot of a single API test mode 46

Figure 5.1.1.2: Data Model of cash study 1 49

Figure 5.1.2.2: Data Model of cash study 2 51

List of Tables

Page

Table 4.5.1: Common API/UI function list for a pattern model 42

Table 4.5.2: entity configuration settings 43

Table 4.5.3: entity property configuration settings 44

Table 5.1.1.1: Case Study 1 - Entity mapping with patterns 48

Table 5.1.2.1: Case Study 2 - Entity mapping with patterns **50**