Thesis Title

Cheuk Chuen Siow
School of Computer Science
McGill University, Montréal

September 2017

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Abstract

Abrégé

Acknowledgements

Preface

Table of Contents

Abstract **ii**

Abrégé **iii**

Acknowledgements **iv**

Preface **v**

List of Figures **viii**

List of Tables **ix**

1 Introduction **1**

2 Background **2**

2.1 Use Case Map (UCM) 2

2.2 Concern-oriented Reuse (CORE) 2

2.2.1 Reusable Aspect Models (RAM) 2

2.3 Review of Related Modeling Tools 2

3 Adding Support for UCM to CORE **3**

3.1 UCM Metamodel 3

3.2 Weaver 3

3.2.1 Responsibility Mapping 3

3.2.2 Connecting Point Mapping 3

4 UCM Implementation in TouchCORE 4
4.1 Scenario Model in TouchCORE 4
4.2 Model Weaving with UCMs 4
4.2.1 Model Extension 4
4.2.2 Model Reuse 4

5 Validation 5
5.1 Case Studies 5
5.1.1 Authentication 5
5.1.2 Online Payment 5
5.2 Workflow Patterns 5

6 Conclusion 6
6.1 Summary 6
6.2 Future Work 6

Appendix A Complete Metamodels 7
A.1 CORE Metamodel 7
A.2 UCM Metamodel 8

Appendix B Complete Case Study Models 9
B.1 Authentication Model 9
B.2 Online Payment Model 9

References 10

List of Figures

A.1 Abstract grammar: CORE metamodel overview 7
A.2 Abstract grammar: UCM metamodel overview 8

List of Tables

Chapter 1

Introduction

First paragraph starts off with a short history of software engineering.

Second paragraph introduces the topic of MDE.

Third paragraph discusses the current state and potential drawbacks of MDE.

Forth paragraph introduces the notion of CORE.

Fifth paragraph continues with TouchCORE and its current state.

Build on the motivation of having UCM as an additional model for TouchCORE.

Last paragraph leads the reader to the remaining chapters.

Chapter 2

Background

URN consists of two notations: GRL for modeling non-functional requirements, and UCM for modeling functional requirements. In this chapter, we describe UCM as part of the requirements engineering tool and its use in specifying scenarios in section 2.1. Then we detail about CORE as a reuse technique with the current state of development in section 2.2. Finally, we provide an overview of related work to modeling tools, applicable to the use of UCM and concern-orientation, in section 2.3.

2.1 Use Case Map (UCM)

This section describes UCM in detail.

2.2 Concern-oriented Reuse (CORE)

This section describes CORE in detail.

2.2.1 Reusable Aspect Models (RAM)

2.3 Review of Related Modeling Tools

Literature review.

Last paragraph leads to implementation chapter.

Chapter 3

Adding Support for UCM to CORE

In this chapter, we describe the integration of UCM as a new modeling language into existing modeling languages of TouchCORE. The project uses Java SE Development Kit 8 as the implementation language and Eclipse Modeling Framework (EMF) [1] as the modeling facility for building TouchCORE. To support a new language, we need to specify its metamodel through EMF. We describe the definition of UCM metamodel in section 3.1. We also present the weaving algorithm for UCM in the Context of CORE in section 3.2.

3.1 UCM Metamodel

We follow the URN specification [2] closely in designing the UCM metamodel.

Figure **??**

The basic structure of a UCM

3.2 Weaver

Again, maybe we need a short intro here. I assume there is also a common algorithm to extensions and reuses, or are they completely separate?

3.2.1 Responsibility Mapping

3.2.2 Connecting Point Mapping

Chapter 4

UCM Implementation in TouchCORE

The definition of UCM metamodel and the specification of weaving algorithm described in previous chapter provide the foundation for the implementation of UCM in TouchCORE. In this chapter, we illustrate the realization of scenario models in TouchCORE through the use of UCM notation in section 4.1. Then we describe the weaving process for extensions and reuses in section 4.2.

4.1 Scenario Model in TouchCORE

Lots of figures here.

4.2 Model Weaving with UCMs

Intro

4.2.1 Model Extension

4.2.2 Model Reuse

Chapter 5

Validation

UCM allows us to describe precise sequences graphically display a workflow model. In this chapter, we attempt to validate our proposed approach of concern-oriented UCMs by means of case studies in section 5.1. Then we demonstrate that concern-oriented UCMs are able to cover the workflow patterns in section 5.2.

5.1 Case Studies

5.1.1 Authentication

5.1.2 Online Payment

5.2 Workflow Patterns

Chapter 6

Conclusion

6.1 Summary

Recap of thesis: what I did so far and what can the tool achieve.

6.2 Future Work

Remaining tasks such as components, path drawing, validation, semantics, etc.

Appendix A

Complete Metamodels

A.1 CORE Metamodel

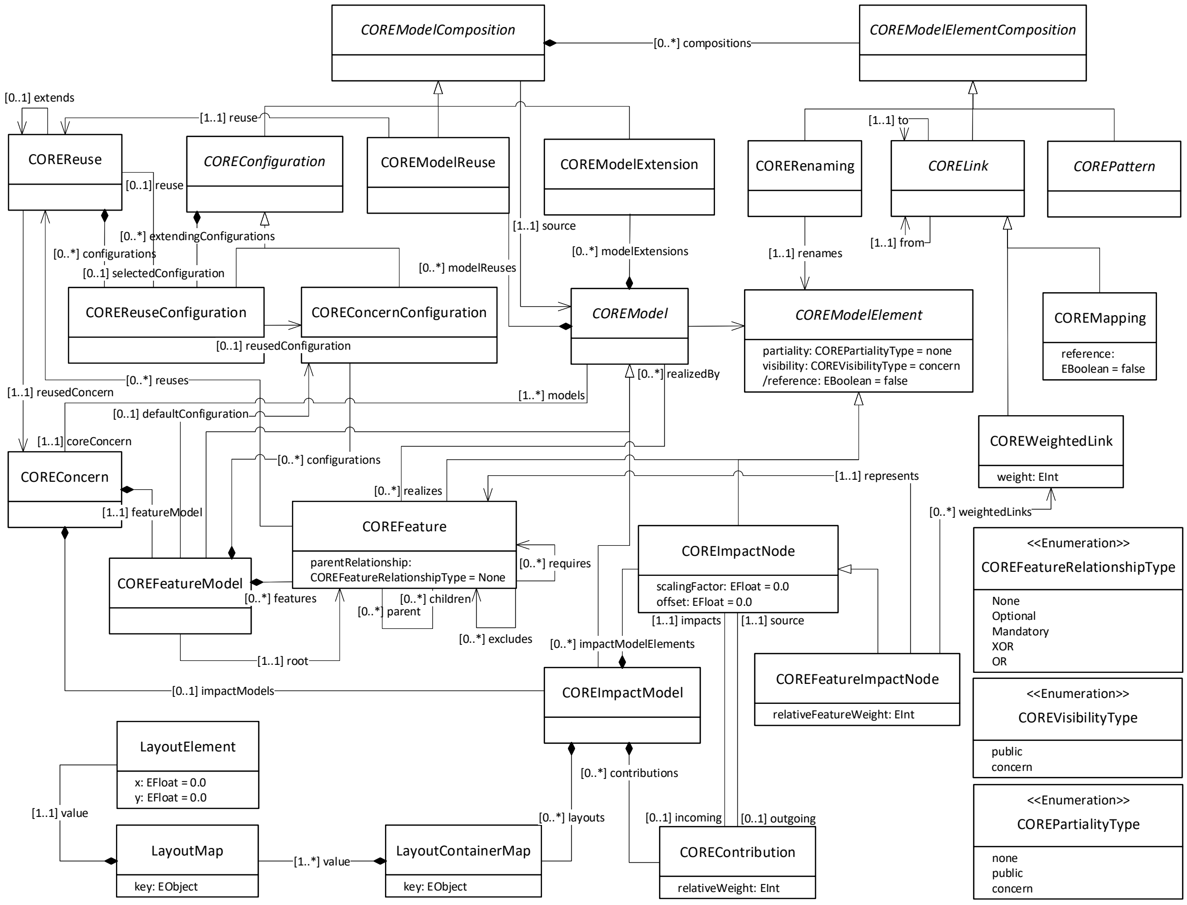

Figure A.1: Abstract grammar: CORE metamodel overview

A.2 UCM Metamodel

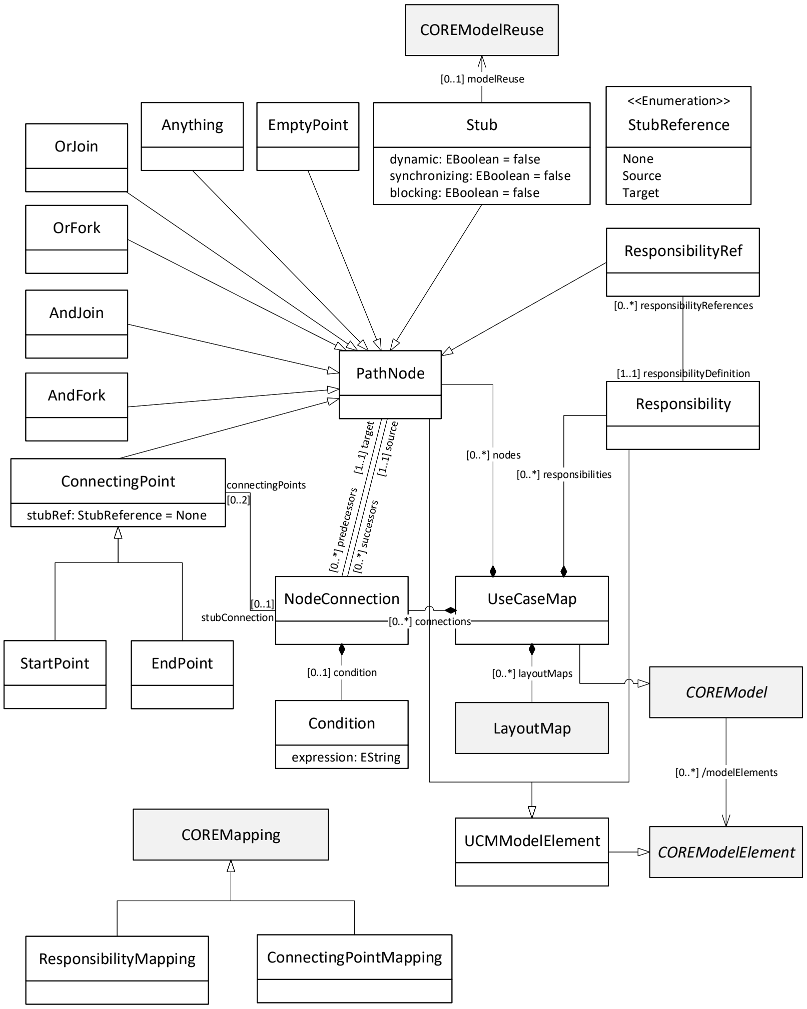

Figure A.2: Abstract grammar: UCM metamodel overview

Appendix B

Complete Case Study Models

B.1 Authentication Model

B.2 Online Payment Model

References

[1] Dave Steinberg et al. *EMF: eclipse modeling framework*. Pearson Education, 2008.

[2] Z ITU-T. “151 User requirements notation (URN)–Language definition”. In: *ITU-T, Oct* (2012).