

Towards Model Validation and Transformation Design Pattern

Muideen Ajagbe, Fiona A.C Polack and Richard Paige,
Department of Computer Science
University of York
York, UK YO10 5DD
Email: maa589@york.ac.uk

CHAPTER 1

1 Introduction

Lately, one of the major focus of software engineering is handling constriction on software systems in order to increase their productivity and reusability. Handling of this constrictions enables the implementation of a system's requirement. As such, contextual and conceptual part of a system needs to be manipulated (transformed) to understand emergent behavior and design patterns that a system may exhibit. These leads to call for the need to use software engineering practices to develop an effective approach towards software systems engineering.

As there are increased complexity and high level of abstraction involved in most software systems, a method that reduces cost with time of delivery to market and produces high quality software is needed. More so, this method is to enable the development of efficient tools and technique that can be used for developing and understanding of complex patterns in software systems.

Model-Driven Engineering (MDE) is a software development approach that seeks to use models as first-class engineering artefacts so as to improve systems productivity, reusability, disintegration and modularization. With MDE, automation of this artefacts leads efficient development and management of systems.

1.1 Synopsis of Model and Model Driven Engineering

Model is an abstraction used in understanding a concept. It is an artifact and specific instances of events in a software system therefore the models developed when using MDE are usually defined in modelling languages, which are well specified using metamodels. A good model is said to conform to its metamodel.

A metamodel provides a means to achieve a different end and it is a qualified variation of a model. Metamodel as a domain specific approach is oriented towards the representation of software systems endeavors. This endeavors may include methods, unit and processes – a focal point in software development.

MDE raises the level of a system abstraction by laying emphasis on model and model transformation. It allows for structured model to be developed for specific domain where it can be engineered by software tools (e.g. code generators). MDE enables construction, manipulation and validation of processes in models – model management.

The need for modelling stems from the ability to proffer solution to a system. The process of modelling involves showing behavior and structure of software systems in details depending on the level of abstractions involved. Abstraction is a nomenclature of complex system and a promising approach to tackling this complexity is to effectively deploy MDE for transformation of models. MDE is more focused on management and manipulation of models thereby providing the capacity to identify and avert errors.

1.2 Overview of Complex Systems

In computing world, complex systems are studied through in silico simulations. Simulation of complex systems enables the development, identification, implementation, verification and manipulation of models. Examples of complex systems includes the complex flocking of birds shown with large samples of simple boids or swarm robotics conforming to simple rules.

Most evidently, the modelling and simulation of a murine autoimmune disease called experimental autoimmune encephalomyelitis (EAE) by Mark Read is another real time scenario of a complex system. There are many approaches to modelling and simulation of complex system. The precise one that sounds as the basis of this thesis by Mark Read is referenced in other chapters below.

In Mark Read, the process of performing simulation-based investigation of complex systems using CoSMoS approach where the domain, domain model, platform model, simulation platform and result model are developed and enabled for a system is an effective process towards simulations. This process

enables the use of simulation for investigations there by performing rigorous modelling activities. A process like these enables the development of computational techniques for the exploitation of many complex systems among which are validation and transformation. Most of the benefit of simulating complex systems include:

- The capturing and merging of vast amount of data from different sources which can be used to developing a system-level synopsis of the behavioral role of the data.
- Also, simulations being a real time system, are more amenable to design, gathering and reusability of data from natural phenomenon.
- Simulation provide a platform for the devising and valuation of hypotheses concerning complex system's operation.
- Model management can be achieved through development, manipulation, verification, validation and transformation of models.
- Most importantly, wet-lab experimentation can be conducted easily with preliminary investigations pointing to specific area of systems being studied.

Specifically, EAE being a form of complex system is an immune system in the same capacity and it is modelled and simulated through computational immunology. Computational Immunology provides methods being used to study immunological systems. This thesis provides an insight to the model validation and transformation using MDE for agent based archetype mostly used in computational immunology. By dedicating model specifics to complex system, semantics like concept of metamodels and models are used to a complex systems advantage. The advantage lies in the idea that they are well refined and more fine-grained for any level of abstraction.

1.3 Model Use and Simulation Confidence

A major focus of model application to complex systems involve the use of diagrammatic approach to capture, connect and think around simulation. Models are noted with scientific and engineering assumptions thereby capturing the understanding of a system. However, models are sometimes generated and updated during simulation-centered research.

For confidence on simulation, there needs to be a conceptual format to the process where simulations are engineered and manipulated using MDE. Confidence is built when there is a modelling process that exposes how scientific

facts are rendered into simulation who in turn helps mitigate unwarranted assumptions. A confidence in processes towards simulation helps strengthened structured models that can be validated and transformed – a pivot of our thesis.

1.4 Motivation and Research Hypothesis

Previous work from YCIL uses similar modelling diagrams to but handcraft code for simulation. The handcrafted code can be clearly written once and easily reused. However, handcrafted code might contain errors and may not reflect the actual representation of the models in contrast to a generated code. MDE can both properly underpin the diagram notations and support transformation to code which is more fit-for-purpose.

From a theoretical view, there is need to elucidate the approaches to support different level of abstraction contained in complex system (hence the need to use various model management processes of MDE) without compromising the systems in play due to the various transformations being carried out by the developers. However, from practical point of view, work carried out by Mark Read of YCIL group explore EAE, an immune system disease that causes paralysis to immune system. By using agent based simulation, the handcrafted code is developed for the models and its simulation. This doesn't give much room for model to be handled in a more distinct way regardless of their respective domains and different levels of abstraction.

The focus of this research is on building solutions and approach to an effective validation and transformation of diagrammatic complex system models used in Mark Reads thesis. This gives us automation capability to modelling the system thereby generating code which can be used to understand the emergent behavior observed in the model in contrast to the handcrafted code techniques employed by the YCIL group. The proposed approach involves transformation of models from behavioral to structural state.

At the core of MDE is model transformation, the method of transforming models overtly between domains. Model transformations experienced in different categories underpin many engineering processes (and automated tools) for MDE. In achieving this, automated transformation and validation of models is used to develop an efficient way of model management without altering their state or losing the big picture of a resulting simulation. A hypothesis of this research questions the process used in model to model of complex systems and how our proposed approach can be used effectively in a non-altering model conformance to transformations shown in complex systems (specifically EAE).

1.5 Research Results

This thesis put forward, a novel approach to model to model transformation of complex systems in different abstraction level using MDE; the CoSMoS framework used by YCIL group is tailored around the model being designed from Mark Reads EAE immunology system model. The work is validated through the deployment of the approach to more complex systems model. Extensive empirical valuation by writing validation test cases with result supporting the research hypothesis of effective transformation is buttressed.

1.6 Thesis Structure

Chapter 2 provides an overview of EAE immunology disease as a complex systems with the sections presenting the low level, top level of the systems model and introducing the reader to the domain of the model and the roles of the classes. Section 2.2 reviews the CoSMoS framework used in the modelling and how it correlates to the individual domain of the system. Section 2.3 provides an overview of MDE and how it helps in the development of an approach towards our transformation of models from one level of abstraction to the other.

Chapter 3 presents the hypothesis this research aims to prove and the breakdown of the approach being used to build on Mark Reads work. Research objectives and the scope the work is restricted to is detailed.

Chapter 4 details the software engineering development process used in building the model, the different models used in each level of abstraction .

Chapter 5 analyzes the object, validation and transformation languages used in building test cases for the transformed model. Strategy used for evaluating the system is presented, automation effort, the result and their significance to the modelling and simulation of EAE model is detailed. Alternative solutions that can be used as part of this work and process for evaluating the whole system is discussed.

Chapter 6 blends a summary of the understanding acquired from the research to a conclusion of the direction of this research in future. It also identify its contribution to the field of software-engineering and model-driven software development.