Rapid Prototyping for Requirements Validation: A Best-Practice with RM2PT

Yilong Yang*, Xiaoshan Li*, Zhi Li‡
*Faculty of Science and Technology, University of Macau, Macau
‡College of Computer Science and Information Technology, Guangxi Normal University, China
Email: yilongyang@um.edu.mo, xsl@um.edu.mo, zhili@gxnu.edu.cn

Abstract—**Rapid prototyping is an effective and efficient way of requirements validation to avoid introducing errors in the early stage of software development. However, manually developing a prototype of a software system requires additional efforts, which would increase the overall cost of software development. This half-day tutorial introduces attendees to an approach with a CASE tool named RM2PT, which can be used for requirements modeling and analysis in UML and automatically generating MVC prototypes from requirements models. By investigating the executions of use cases in the generated prototypes, the stakeholders can easily check whether the requirements reflect their real needs. Moreover, requirements inconsistency can be automatically detected and further fixed through the provided features of the generated prototype. Participants will be given the chance to use RM2PT with the selected case studies, and learn how RM2PT has been and can be applied to real-world projects.**

Index Terms—**Requirements, Requirements Validation, UML, Prototype, Rapid Prototyping, Automatic Generation**

I. Motivation and Objectives

Requirements errors are one of the causes leading failings in software projects. Careful requirements modeling along with systematic validation helps to reduce the uncertainty about target systems. Rapid prototyping is an effective approach to requirements validation to demonstrate concepts, discover requirements errors and find possible fixing solutions [1]. In practice, it is very desirable to generate prototypes directly from requirements automatically with a CASE tool. To fill this gap, we present an approach [2] and a tool RM2PT [3] [4] to automatically generate prototypes from requirements models. The stakeholders can easily check whether the requirements reflect their real needs by investigating the executions of use cases in the generated prototypes. Besides, the conflict and contradictory of the requirements (especially between the contracts of system operations and invariants) can be automatically detected and further fixed by the features of consistency checking and state observation in the generated prototypes.

This tutorial aims to provide an introduction to RM2PT, and how it can be used to help stakeholders for requirements modeling and validation. On completion of the tutorial, participants will be able to:

- create requirements models in UML diagrams complemented by the contracts of system operations in Object Constraint Language (OCL)[1]
- automatically generate and execute prototypes from requirements models in RM2PT
- validate requirements by investigating the executions of use cases and fixing requirements errors through the generated prototypes in RM2PT

II. Outline of Topics

We would like to organize a half-day tutorial, which is structured as follows:

1) Overview of RM2PT (1 hour)
   - Requirements modeling and analysis in RM2PT
   - Automated prototype generation from requirements
   - Requirements validation and error fixing
2) A Case Study through RM2PT (1 hour)
   - Overview of the case study
   - Requirements modeling and analysis
   - Requirements validation by automatic prototyping
   - Three-round requirements modeling and validation
3) Hands-on RM2PT (1 hour)
   - Participants apply RM2PT to their projects
   - Start-up their projects through several rounds of requirements modeling and validation
   - Discussion and Q&A

In the first hour, we will present an overview of RM2PT, which includes 1) what kind of requirements model is adopted in the RM2PT. In brief, RM2PT takes UML diagram as a requirements model, which includes a use case diagram, a conceptual class diagram (no operations in the classes) with class invariants, system sequence diagrams (system events only between actors and systems) for use cases, and the contracts of their system operations formally specified by a pair of pre- and post-conditions in OCL; 2) how to generate prototypes from a requirements model; 3) how to use the generated prototype to validate the requirements by investigating the executions of use cases and how to automatically detect requirements errors and further fix them by the provided features of prototypes.

In the next hour, we will introduce a case study step by step in RM2PT. We plan to present three-rounds of requirements modeling and validation starting from scratch to achieve a validated model. Specially, we introduce 1) how to do validity

[1] http://www.omg.org/spec/OCL/

checking to confirm whether the requirements reflect the real needs of stakeholders, 2) how to do consistency checking to examine whether requirements models contain conflict and contradictory, and 3) how to locate and fix errors by using the mechanism of observing the current state of the objects in the prototype.

The remaining one hour is left to participants for hands-on RM2PT. 1) The participants can either create a requirements model from scratch or based on the provided the case studies from the Github repository[2], which includes a supermarket management system, a library management system, an automated teller machine, and a loan processing system. 2) Based on the created requirements model, participants can automatically generate prototypes in RM2PT and then validate their requirements by investigating the executions of each use case. 3) The prototype will prompt users if any requirements error causes the system into an unexpected state and help users to make further investigations to locate and fix the errors. 4) After several rounds of modeling and validation, participants can share their validated model to enhance the community by sending pull requests to the RM2PT case studies repositories.

III. Target Audience

Attendees are assumed to have some familiarity with object-oriented requirements modeling and analysis in UML. This tutorial will benefit three participants groups.

- *Industry practitioners*. The tutorial provides hands-on experience using RM2PT. Practitioners can easily leverage on their projects based on the exercises and case studies.
- *Researchers*. The tutorial provides useful knowledge on requirements engineering and several directions for future research based on RM2PT.
- *Educators*. All materials used in this tutorial will be available on RM2PT website[3] along with guidance on incorporating it into the undergraduate and postgraduate courses such as requirements engineering. RM2PT can give students the intuition of the importance of requirements engineering and how requirements relate to implementations through automatic generating prototypes and investigating executions of use cases in the requirements modeling stage.

IV. Tutorials History

It is the first time we offer such a tutorial of RM2PT in a conference setting, but the parts of this tutorial have been taught in the Bachelor and Master courses at the University of Macau as well as the seminars at the universities such as Beihang University, Central South University, Guangxi Normal University, and Huawei Technologies Co., Ltd.

V. Presenter'Bios

Yilong Yang is a post-doctoral fellow at the University of Macau, where he received his Ph.D. degree in Software Engineering. His research interests are Automated and Intelligent Software Engineering. He has published over 20 papers in international conferences and journals. He has also been a fellow at the UNU-IIST and visiting research fellow at the Guangxi Normal University. He serves as the PC member for IJCAI'20 and ECAI'20.

Xiaoshan Li received his Ph.D. degree from the Institute of Software, the Chinese Academy of Sciences. Currently, he is an Associate Professor in the Department of Computer and Information Science at the University of Macau. His research interests include Automated Software Engineering, Formal Specification and Verification, and Formal Semantics of UML.

Zhi Li is Professor at Guangxi Normal University, China. His research interests are Problem-oriented Requirements Engineering for big data analytics, modeling, and verification of cyber-physical systems. His research has been sponsored by grants from the National Natural Science Foundation of China and he has published over 20 research papers.

VI. Publicity

The authors have been involved in requirements and software engineering communities [2] [3] [4] [5] [6] and collaborations with software companies. They will intensively promote the tutorial through their networks and extend RM2PT to support other requirements specifications and models such as Easy Approach to Requirements Syntax (EARS), Goal Model, and Problem Frames (PF).

VII. Equipment

RM2PT is a free CASE tool. Any audience is advised to bring her laptop and download RM2PT before the tutorial. The minimum system requirements are Windows 7 / MacOS 9 / Linux, 1GHz CPU, 2GB memory, 1GB free disk space with JDK8 (64 bit) installed.

Acknowledgment

This work was supported by Macau Science and Technology Development Fund (FDCT) (No. 103/2015/A3) and University of Macau (No. 2017-00141-FST), and National Natural Science Foundation of China (NSFC) (No. 61562011 and 61862009)

References

[1] F. Kordon and Luqi, "An introduction to rapid system prototyping," *IEEE Transactions on Software Engineering*, vol. 28, no. 9, pp. 817–821, Sep. 2002.

[2] Y. Yang, X. Li, W. Ke, and Z. Liu, "Automated prototype generation from formal requirements model," *IEEE Transactions on Reliability*, pp. 1–25, 2019.

[3] Y. Yang, X. Li, Z. Liu, and W. Ke, "RM2PT: A tool for automated prototype generation from requirements model," in *Proceedings of the 41th International Conference on Software Engineering: Companion Proceedings (ICSE'19)*, May. 2019, pp. 59–62.

[4] Y. Yang, W. Ke, and X. Li, "RM2PT: Requirements validation through automatic prototyping," in *2019 IEEE 27th International Requirements Engineering Conference (RE'19)*, Sep. 2019, pp. 484–485.

[5] Z. Li, J. G. Hall, and L. Rapanotti, "On the systematic transformation of requirements to specifications," *Requirements Engineering*, vol. 19, no. 4, pp. 397–419, Nov 2014.

[6] N. Gao and Z. Li, "Generating testing codes for behavior-driven development from problem diagrams: A tool-based approach," in *2016 IEEE 24th International Requirements Engineering Conference (RE'16)*, Sep. 2016, pp. 399–400.

[2] https://github.com/RM2PT/CaseStudies

[3] http://www.rm2pt.com