

Modular Conformance Testing and Verification for Evolving Component-Based Software

Student: Pham Ngoc Hung

Supervisor: Prof. Takuya Katayama

1. Research Objective

The research focuses on model-based techniques for modular verification of component-based software (CBS), i.e., verifying global system properties by checking components separately in the context of “*component refinement*”. Currently, the model-based techniques for software verification generally assume the availability of a model that describes the behavior of the software under checking. However, the assumption may not always hold in practice. Even the assumption holds the model could be invalidated when the software is evolved. This initiates the study of adaptive model checking (AMC) which necessitates an iterative construction of a model for software by learning. Nonetheless, the model in AMC describes the behavior of the whole software. The “*state space explosion*” may occur when checking large-scale software. Furthermore, when system is changed, the model is required to update including comparing software with the new candidate model. Updating on the whole model is not necessary because the changes often focus on a few software components. The main purpose of the research is to combine AMC and the assume-guarantee verification method as a modular model checking technique to deal with the “*state space explosion*” and to reduce the expensiveness of the conformance testing in the AMC framework.

2. Research Approach

We propose a framework about adaptive modular model checking for evolving component-based software. The key idea of this framework is to apply the ideas in the AMC and modular model checking to verify evolving CBS with a new “*component refinement*” concept which means that only adding some behaviors into the old component. For each existing component of CBS, suppose that we obtain a corresponding model by applying some modeling techniques or manual processes. When a software component is changed after adapting some refinements, if the corresponding model of the component is inaccurate then the model is updated by a learning algorithm. We use black box testing to check

conformance between the evolved component and its model. This framework also uses the assume-guarantee method to verify the evolved software by checking the components separately. The proposed framework not only tries to reuse the previous verification result, but also tries to reuse the previous models to reduce some steps of the verification process.

3. 2007's Progress

In 2007, the first task was survey researches in verification for evolving software, i.e., Black Box Testing [Peled'99], Adaptive Model Checking [Groce'06], and Verification of Evolving Software [Chaki'04&05]. The main task was proposing a framework about adaptive modular model checking for evolving component-based software with our “*component refinement*” concept. We found a modeling method for software components, studied a learning method for updating the evolved component, and proposed a technique called “*modular conformance testing*” to compare each software component and its model. Finally, the research goal has been solved by combining the assume-guarantee method, the “*modular conformance testing*” and the learning method into the proposed framework.

4. Future Direction

The proposed framework mentioned in the section 2 still is in progress. What the research is aiming to carry out in 2008 is to deploy the proposed framework more details and finding some case studies to illustrate the proposed techniques and the proposed framework. Correctness, termination and effectiveness of the proposed framework will be evaluated by some case studies.

5. Publications

- [1] P. N. Hung, N. T. Thang, T. Katayama, *An Assume-Guarantee Method for Modular Verification of Evolving Component-Based Software*, In Supplemental Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2007), Workshop on Architecting Dependable Systems (WADS), pages 160-165, 2007.
- [2] P. N. Hung and T. Katayama, *Adaptive Modular Model Checking for Evolving Component-Based Software*, In the Proceeding of JVSE'07 Workshop, 2007.