The Research Monograph Series in Computing, Electrical & Communication Networks
Vol. 1 (01), April 2023, pp. 42–43

# CHAPTER 6

# CONCLUSION AND FUTURE ENHANCEMENTS

## 6.1 OVERVIEW

Many problems which are inherent in engineering large, complex, distributed, long lived software systems have been doubted with software architecture. In the emergence of many architectural styles, modeling notations and analysis techniques from the software architecture research community many developers are provided with conceptual tools to tackle these problems. But these approaches do not address the relationship between the abstract architectural models and concrete system implementations.

## 6.2 CONCLUSION

The automated generation of performance feedback in software architectures is dealt in this thesis. In order to keep track of the performance knowledge which tends to be fragmented and quickly lost a methodology is devised. This is done with the objectives of interpreting the performance analysis results and with the suggestion of the most suitable architectural reconfigurations. This aims at integration of different forms of data (e.g. architectural model elements, performance indices), to maintain relationships between them and manage the data over time.

Although there exists, many approaches in view of evaluating software architecture, implementation of these in software product line is expensive and lack of focusing on the reusability of the products. On the contrary, the proposed approach which has a comprehensive product line architecture modeling methods is set to change its basic approach. The advantages are using reusable, built-in change sets, and the surface evaluation.

The research work exposes a model for implementation of refactoring the software architecture automatically into a performance model. The research work is concluded by identifying

- The principles and concept of automatic transforming architecture model are discovered and integrated.
- Recognize the translation of software model into performance model.
- To simplify development process for performing automated analysis of Component Based Architecture, a methodology was derived.
- The approach is demonstrated as a framework which utilizes the methodology designed, implemented, utilized and evaluated.
- Provides a feedback at the design level to Refactor and improvise the design.

The detailed research of automatic transforming architecture design through UML diagram is offered.

## 6.3 FURTHER ENHANCEMENTS

The validation and lack of model parameters are the two recommendations considered for future research work. The approach needs to be more extensively validated to determine the extent of its support to the user activities. The two dimensions in validation are: (i) it has to be exposed to a set of target user's e.g. model-driven developers, graduates in a software engineering course towards analyzing its scope and usability. (ii) Its application in complex case studies with the involvement of industry partners towards analyzing its scalability. It is worth to have thus experimentation. By analyzing the model piecewise, beginning from sub models, lack of information or even uncertainty, about model parameter values can be tackled. It is believed that these two recommendations are appropriate for the present research work.

The VSD tool can be improved in various ways. First random properties of the static objects are allowed to utilize the class diagram which can be achieved by attaching a "note" to the appropriate class. Thus classes can be used towards indicating the multiplicity of the objects involved in the system.