The Research Monograph Series in Computing, Electrical & Communication Networks
Vol. 1 (01), April 2023, pp. 16–19

CHAPTER 3

REFRACTORING OF SOFTWARE MODEL

3.1 INTRODUCTION

The process of predicting and evaluating the software system in satisfying performance of goals of the User is software performances. A suitable abstract model of the final software is required to predict the performance of a software system. It should also provide a suitable description of the run-time behavior of the software system to estimate that performance. Measuring i.e. the activity of measuring the performances of the system's implementation is referred to performance evaluation. The performance-intensive software systems quality, like high-performance scientific computing systems and distributed real-time and embedded (DRE) systems is largely dependent on their infrastructure platforms which include the platform, operating system, middleware, and language processing tools. But the infrastructures need to be tuned by the developers along with the software applications in order to accumulate the often changing platform environments and performance requirements.

3.2 PROPOSED SYSTEM

Unified Model Language has been accepted as a standard for designing new systems. System designers capture their ideas through its array of notations and thereby making it expressive to understand easily. Thus a novel scheme is proposed by the research work to develop a performance model by building a software model. In this research work, UML profile is represented defining tagged values, constraints and new stereo types to trace design patterns in UML diagram. And this approach visualizes design patterns in UML diagrams.

3.3 PROPOSED ARCHITECTURE DESIGN

The transformation from software model to performance model is described and detailed by the proposed architecture design in Figure 3.1,

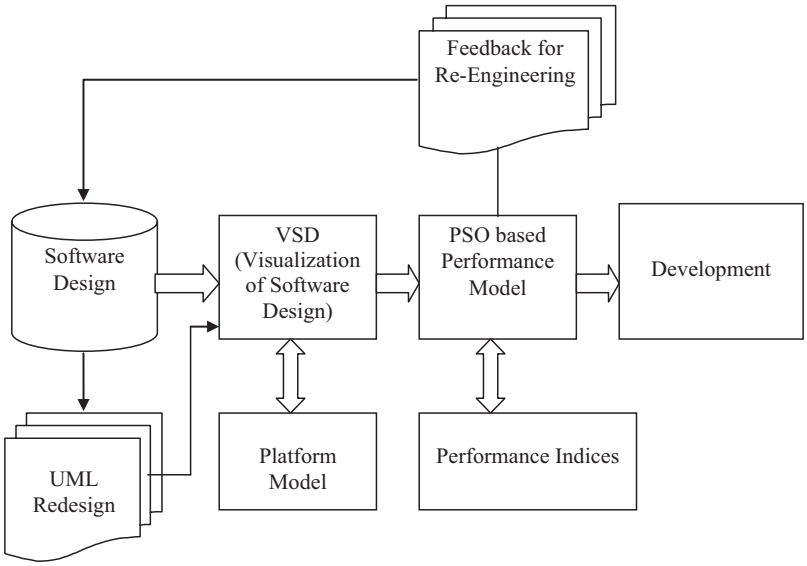

Figure 3.1 Architecture Diagram.

As the architecture satisfies all needs of the user, because the architecture designs contains a feedback collecting method.

The design which is made by UML for a system is mapped on the QN with the help of Visualization of Software Design (VSD) which is created with the help of java tool. Visualizing, specifying, constructing and documenting the artifacts of software intensive system is done by UML, which is a graphical language. An interaction diagram is provided by UML to model the dynamic aspects of a system. Besides including the messages which might be sent from one object to another.

It consists of objects, and their relationship, constructing an executable system through forward engineering with regard to achieve automatic performance generation, made possible by an interaction diagram.

Software is designed with UML. This UML diagram uses VSD tool to map the UML diagram into QN in which QN and platform model supports it in conversion. The evaluation of indices is generated by performance model such as Response Time, Throughput and resource utilization, refer to performance indices. Particle Swarm Optimization (PSO) helps the performance indices to evaluate the generated outcome.

Analyzing performance indices is done successfully, with the help of PSO and it is verified with the help of Spearman correlation coefficient. If the values are acceptable then it is directed to software development else a feedback is sent to the UML redesign. The redesign is made with the help of the rule which has been proposed by the researcher.

3.3.1 Phases of Research

The Research work is done in three different phases:

Phase 1: Software model is designed using the UML diagrams, annotated with the help of VSD by using the performance information such as Schedulability, Performance and Time specification (SPT) profile and mapped on to the QN.

Phase 2: QN solved using QNsolver, the quantitative results are optimized using the PSO algorithm.

Phase 3: Functional attributes are checked using the PSO, if the design needs a change, apply rules and give the feedback making changes, until the customer requirement is satisfied iterate from Phase 2.

3.4 ARCHITECTURE FRAMEWORK

The representation of the architecture flow is shown in the Figure 3.2. The input is given as a UML diagram.

The UML diagrams are annotated and are mapped to the QN using the VSD, a tool that is created using the java. This tool, maps with the help of the QNBE elements shown in the Figure 3.2, with these elements the QN is solved using the QNsolver.

The attributes are evaluated using the PSO for the optimization and checked for the customer requirements, if the qualitative attributes are satisfied then we propose the diagram for development. If the attributes are not satisfied, then the flow proposes a feedback for refactoring the software design with the help of the proposed rules framed by the researcher.

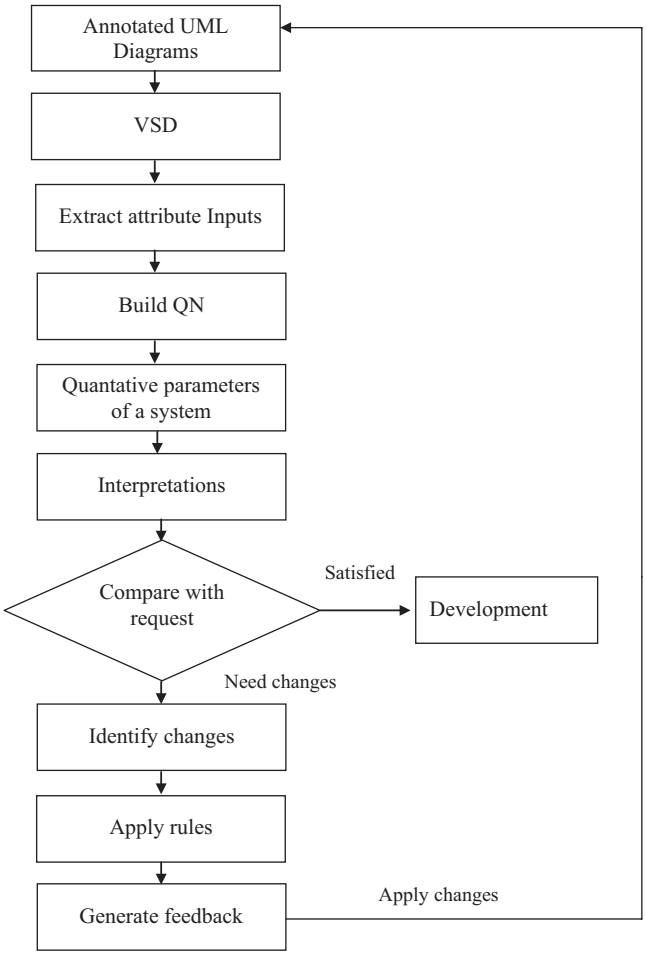

Figure 3.2 Architecture activities.