

**REFACTORING OF  
SOFTWARE ARCHITECTURAL  
DESIGN FOR PERFORMANCE  
OPTIMIZATION**

**R. AROUL CANESSANE**

The Research Monograph Series in *Computing, Electrical & Communication Networks*

**Refactoring of Software Architectural Design for Performance Optimization**

Author: R. Aroul Canessane

**Published by**

**Bohr Publishers, Chennai, India**

Old Door No. 4, New Door No. 3, Rishilaya Building, 2nd Street, Golden Avenue,  
Devi Karumariamman Nagar, Velachery, Chennai – 600 042, India

©2022 Bohr Publishers

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or storage in any information or retrieval system, without the prior permission of the copyright owner.

ISBN: 978-93-92892-04-2 (Print)

978-93-92892-19-6 (Online)

Typeset and Cover designed by  
Chennai Publishing Services Pvt. Ltd., Chennai, India

Printed by Ridha Prints Pvt. Ltd., Madurai, India

## ACKNOWLEDGEMENT

At the outset, it is imperative to mention that this work would not have been completed without the helping hands of the noble souls. To them, I owe my gratitude, the depth of which, mere words cannot express. Though it is a small recompense for the invaluable assistance they have been given, I nevertheless avail this opportunity to thank them.

I wish to express my sincere thanks to all my faculty colleagues, my students and friends who had supported me whenever needed. My sincere thanks to my parents and family members for their continuous encouragement and support to complete my research work.

Finally, I thank the GOD almighty and all unseen hands behind my success and pray pardon for the omission of their names.

**R. AROUL CANESSANE**



## ABSTRACT

Many advanced systems are created through many inventions and new technologies which make a major impact on the mankind. The development and structure of complex systems are focused on the field of systems engineering, which plays a major role on these inventions and technologies in the field of Information Technology.

Understanding the fulfillment of the requirements is the fundamental idea of software development process for surviving among the end users else we have to face some critical consequences, such as failure. Reworking on the same product is much expensive i.e., after completion of the product. We cannot give completeness for those products rather than it can provide only sufficiency, where the life time of the software product will automatically reduce.

The increase in terms of size, logical distribution and complexity of interaction within the software systems out phases the growing importance of early performance assessment. Predictions and Evolution of quality of software at the early stage of developing the software will improvise the quality of the software and increase the life time of the product.

Analyzing the quality attributes at the stage of design phase of the software product will reduce the risks both functional and non functional aspects that will arise after the completion of the product. In order to overcome the problems that is attached to the software development, can be rectified using the performance engineering, a field that proposes methods and tools, which is an emerging trend.

Many applications detect flaws at the design phase for software security, verification and validations, which can be rectified at the early stages to get a effective end product. Many approaches have effectively solved the difficult task of transforming software models into performance models. The research issue is proposed a performance model which has a feedback mechanism for refactoring the software design for yielding to an effective software product.



# TABLE OF CONTENTS

<b>ACKNOWLEDGEMENT</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>v</b>
<b>1. INTRODUCTION</b>	<b>1</b>
1.1 SOFTWARE PERFORMANCE ENGINEERING	1
1.2 ARCHITECTURE OF THE SOFTWARE	2
1.3 SOFTWARE DEVELOPMENT PROCESS	2
1.4 UML	4
1.4.1 Capsules	4
1.4.2 Ports	5
1.4.3 Connectors	5
1.5 FEEDBACK	5
1.6 PROBLEM DESCRIPTION	6
1.7 RESEARCH OBJECTIVE	6
1.8 RESEARCH APPROACH	7
<b>2. PROBLEM FORMULATION</b>	<b>8</b>
2.1 OVERVIEW	8
2.1.1 Automated Software Performance Model	9
2.2 PROCESS OF AUTOMATED SOFTWARE PERFORMANCE	11
2.3 PERFORMANCE PREDICTION USING MODELS	12
2.3.1 Performance Model from Scenarios	13
2.3.2 Performance based on Objects and Components	14
2.4 ARCHITECTURAL DESCRIPTIONS FOR TRANSFORMATION TOOL TO QN	14
2.5 SUMMARY	15
<b>3. REFRACTORING OF SOFTWARE MODEL</b>	<b>16</b>
3.1 INTRODUCTION	16
3.2 PROPOSED SYSTEM	16
3.3 PROPOSED ARCHITECTURE DESIGN	16
3.3.1 Phases of Research	18
3.4 ARCHITECTURE FRAMEWORK	18

<b>4.</b>	<b>PERFORMANCE OPTIMIZATION</b>	<b>20</b>
4.1	RE-ENGINEERING – CHANGES IDENTIFICATION	20
4.1.1	Parameter Selection in the Analysis	20
4.1.2	Rules for Performance Identification	21
4.1.3	Modification Rules	21
4.2	QUEUING NETWORK	23
4.2.1	Queuing Network Transformation UML	23
4.2.2	Algorithm: Queuing Network	24
4.2.3	QN Transformation a Hierarchical Approach	24
4.2.3.1	Connectivity Rules to Basic Elements of Queuing Network	25
4.3	INDICES OF PERFORMANCE	25
4.4	PARTICLE SWARM OPTIMIZATION (PSO)	28
4.4.1	PSO Algorithm	29
4.5	SUMMARY	29
<b>5.</b>	<b>RESULTS AND DISCUSSIONS</b>	<b>30</b>
5.1	INTRODUCTION	30
5.2	THROUGHPUT UTILIZATION ANALYSIS	30
5.3	PERFORMANCE ANALYSIS	30
5.3.1	Analysis of Real-time	31
5.3.2	Analysis of Stochastic Performance	32
5.3.3	Analysis of Sensitivity	32
5.4	ANALYSIS WITH RELIABILITY MODELING	32
5.5	MATHEMATICAL COMPLEXITY AND NONLINEARITY OF SYSTEM	33
5.6	ANALYSIS OF TRANSFORMATION	34
5.7	REDESIGNING PROCESS RESULTS	36
5.8	VARIOUS PERFORMANCE PREDICTION METHODS	38
5.9	SUMMARY	41
<b>6.</b>	<b>CONCLUSION AND FUTURE ENHANCEMENTS</b>	<b>42</b>
6.1	OVERVIEW	42
6.2	CONCLUSION	42
6.3	FURTHER ENHANCEMENTS	43
	<b>REFERENCES</b>	<b>44</b>