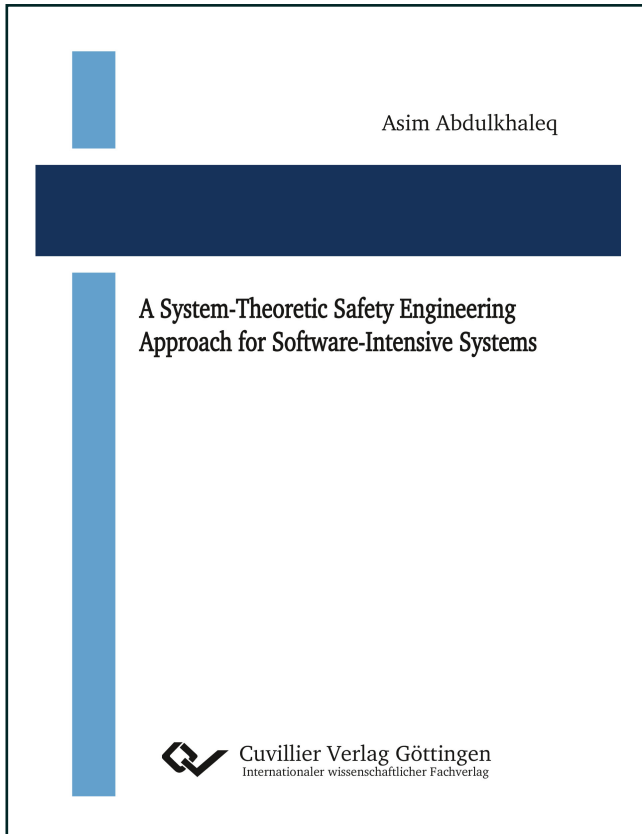




Asim Ali Ahmed Abdulkhaleq (Autor)

A System-Theoretic Safety Engineering Approach for Software-Intensive Systems



<https://cuvillier.de/de/shop/publications/7484>

Copyright:

Cuvillier Verlag, Inhaberin Annette Jentsch-Cuvillier, Nonnenstieg 8, 37075 Göttingen, Germany
Telefon: +49 (0)551 54724-0, E-Mail: info@cuvillier.de, Website: <https://cuvillier.de>



CONTENTS

	Page
Abstract	5
List of Figures	15
List of Tables	17
List of Definitions	19
List of Abbreviations	21
Glossary	23
1. Introduction	25
1.1. Problem Statement	26
1.2. Research Objectives	27
1.3. Contributions	27
1.4. List of Publications	29
1.5. Outline	30
2. Background	31
2.1. Software Safety Challenges	31
2.2. Safety Analysis Techniques	32
2.2.1. Traditional Safety Analysis Techniques	33
2.2.2. System-Theoretic Safety Analysis	35
2.2.3. Software Safety Analysis Challenges	41
2.3. Software Verification	45
2.3.1. Formal Verification	45
2.3.2. Software Testing	49
3. State of the Art	57
3.1. Generating the Unsafe Control Actions in STPA	57
3.2. Combination of Safety Analysis Techniques and Model Checking	58
3.3. Translating Simulink Models into Verification Models	60



3.4. Risk-based Software Testing	62
3.5. Generating Test Cases Using Statechart Diagrams	63
3.6. Generating Test Cases from Simulink Models	64
4. Approach	67
4.1. Deriving Software Safety Requirements	70
4.2. Formalising Software Safety Requirements	73
4.3. Constructing a Safe Software Behavioral Model	74
4.4. Software Safety Verification	77
4.5. Safety-based Test Case Generation	78
4.6. Summary	79
5. Automation of Approach	81
5.1. STPA Components in XML Specification	81
5.2. Automatically Generate Unsafe Scenarios	82
5.2.1. Generate Context Tables	82
5.3. Automatically Formalise the Safety Requirements	86
5.4. Safety-Based Test Case Generation	88
5.4.1. Automatically Transforming a Safe Software Behavioral Model into an SMV Model	88
5.4.2. Automatically generating the Safe Test Model from the Safe Software Behavioral Model	98
5.4.3. Automatically Generating Safety-Based Test Cases	102
5.5. Summary	107
6. Tool Support	109
6.1. XSTAMPP	109
6.1.1. XSTAMPP Architecture	110
6.1.2. Design and implementation	111
6.1.3. XSTAMPP Plug-ins	113
6.2. Summary	124
7. Empirical Validation	125
7.1. Pilot Case Study: Developing A Software Simulator for ACC . . .	125
7.1.1. Case Study Description	125



7.1.2. Results	129
7.1.3. Discussion	140
7.2. Industrial Case Study on BMW's ACC with Stop-and-Go function	142
7.2.1. Case Study Description	142
7.2.2. Case Study Design	143
7.2.3. Results	147
7.2.4. Discussion	163
7.3. Industrial Case Study on Continental's Fully Automated Vehicle .	164
7.3.1. Case Study Description	164
7.3.2. Fully Automated Vehicle	166
7.3.3. Results	169
7.3.4. Discussion	177
7.4. Summary	178
8. Conclusions	179
8.1. Summary	179
8.2. Lessons learned	180
8.3. Limitations	182
8.4. Future Work	184
8.4.1. Using STPA and STPA SwISs in Compliance with ISO 26262	184
8.4.2. Visualization of the STPA Results	185
8.4.3. Using STPA SwISs Results for Auto Safe Code Generation	185
8.4.4. New Improvements to the Tool Support	185
Bibliography	187
Appendix	197
A. ACC Simulator	197
A.1. The C Code of the ACC Simulator	197
A.2. The SMV Model of ACC Simulator	202
A.3. The Promela Model of ACC Source Code	206