

University of Groningen

Execution architecture views for evolving software-intensive systems

Callo Arias, Trosky Boris

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version

Publisher's PDF, also known as Version of record

Publication date:

2011

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Callo Arias, T. B. (2011). Execution architecture views for evolving software-intensive systems. s.n.

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Contents

Samenvatting	v
Abstract	vii
1 Introduction	1
1.1 Software Architecture	1
1.1.1 Software Architecture Descriptions	2
1.1.2 Software Architecture Reconstruction	4
1.2 Software-Intensive Systems	4
1.2.1 Magnetic Resonance Imaging Scanners	5
1.3 Evolvability of Software-Intensive Systems	7
1.3.1 Architecture-Centric Evolution	8
1.4 Problem Statement	9
1.5 Research Questions	11
1.6 Research Approach	13
1.6.1 Industry as Laboratory	13
1.6.2 Research Methods	13
1.6.3 Types of Research Questions	15
1.6.4 Types of Research Results	15
1.6.5 Validation Techniques	17
1.7 Overview of this Thesis	17
2 A Systematic Review of Dependency Analysis Solutions	21
2.1 Introduction	21
2.2 Context and Research Questions	23
2.2.1 The Software of the Philips MRI Scanner	23
2.2.2 Research Questions	23
2.3 Design of the Review	25
2.3.1 Study search and selection	25
2.3.2 Data extraction	28
2.3.3 Data synthesis	31

2.3.4	Interpretation	31
2.4	Overview of Concepts about Dependencies	32
2.4.1	Definition of dependencies in the literature	32
2.4.2	Types of dependencies	33
2.5	Application Areas of Dependency Analysis	34
2.5.1	Application Level Analysis and Management	36
2.5.2	Architecture Description and Analysis	37
2.5.3	Change Impact Analysis	37
2.5.4	Program/System Understanding	38
2.5.5	Quality Assurance, Testing and Debugging	38
2.5.6	Refactoring and Modularization	39
2.5.7	Traceability and Feature Analysis	39
2.6	Existing Dependency Analysis Solutions	40
2.6.1	Source code-based solutions	40
2.6.2	Descriptions and Model-based solutions	43
2.6.3	Run-time monitored and Configuration-based solutions	45
2.7	Applicability of Dependency Analysis	46
2.7.1	Applicability of definition of dependencies	47
2.7.2	Source code-based solutions	48
2.7.3	Model-based solutions	52
2.7.4	Run-time monitored and Configuration-based solutions	53
2.8	Threats to validity	55
2.8.1	Internal validity	55
2.8.2	External validity	57
2.9	Concluding Remarks	58
3	A Top-Down Approach to Construct Execution Views	59
3.1	Introduction	59
3.2	Overview of the approach	61
3.2.1	Reconstruction design	62
3.2.2	Reconstruction execution	63
3.3	Elements for the reconstruction design phase	63
3.3.1	Execution viewpoints	64
3.3.2	Execution metamodel	66
3.4	The reconstruction execution phase	70
3.4.1	Sources of runtime information	70
3.4.2	Mapping rules repository	72
3.4.3	Task identification	73
3.4.4	Interpretation of runtime information	75
3.4.5	Construction of execution model	77
3.5	Validation of the approach	78
3.5.1	Design reconstruction for an execution profile view	78
3.5.2	Execution reconstruction for an execution profile view	79
3.6	Use cases for an execution profile view	81

3.6.1	Feature analysis.	81
3.6.2	Dependency analysis.	82
3.6.3	Conformance and realization analysis.	85
3.7	Contribution and potential limitations	86
3.7.1	Technical contribution	87
3.7.2	Potential limitations	90
3.8	Related Work	91
3.8.1	Top-down solutions	91
3.8.2	Runtime description and analysis	92
3.9	Conclusions and Future Work	93
4	Constructing a Resource Usage View	95
4.1	Introduction	95
4.2	Customized Top-Down Approach	96
4.2.1	Resource Usage Viewpoint	97
4.2.2	Execution Metamodel	98
4.2.3	Sources of Runtime Information	99
4.3	Input and Interpretation of Runtime Activity	99
4.3.1	Scenario Input Analysis	100
4.3.2	Interpretation of Runtime Activity	100
4.3.3	Input to Construct a Resource Usage Model	101
4.4	Models of a Resource Usage View	102
4.4.1	Task Resource Usage Models	102
4.4.2	Component Resource Usage Models	103
4.4.3	Thread Resource Usage Models	105
4.5	Resource Usage Models in Practice	106
4.5.1	Using a Resource Usage View	107
4.5.2	Supporting Development Activities	107
4.6	Related Work	108
4.7	Conclusions and Future Work	109
5	Defining and Documenting Execution Viewpoints	111
5.1	Introduction	111
5.2	Predefined execution viewpoints	114
5.2.1	Motivation	114
5.2.2	Identified predefined viewpoints	114
5.3	Eliciting the organization’s requirements for execution views	116
5.3.1	Questionnaire design	116
5.3.2	Interviews	117
5.3.3	Identified concepts and concerns	118
5.4	Initial definition and validation of execution viewpoints	122
5.4.1	Initial definition of execution viewpoints	122
5.4.2	Validation of the initial definition of execution viewpoints	125
5.5	Documentation of execution viewpoints	127

5.5.1	Execution profile (formerly called functional mapping)	128
5.5.2	Resource usage viewpoint	131
5.5.3	Execution concurrency viewpoint	134
5.6	Conclusions	138
6	Reverse Architecting Execution Views in Practice	139
6.1	Introduction	139
6.2	Case Study: The Start-Up of the Philips MRI Scanner	141
6.2.1	The Start-Up Process	141
6.2.2	The Start-Up of the Philips MRI Scanner	142
6.2.3	Case Study Concerns and Decisions	142
6.3	The Architecture Reconstruction Approach	143
6.3.1	Inputs to the Process	143
6.3.2	Steps in the Process	145
6.3.3	Output of the Process	146
6.4	The Top-Down Strategy and Its Application	146
6.4.1	Phase 1: Inception	148
6.4.2	Phase 2: Elaboration	149
6.4.3	Phase 3: Analysis	152
6.4.4	Phase 4: Transition	153
6.5	Technical Contribution	155
6.5.1	Top-Down Analysis	155
6.5.2	Communication Between Stakeholders	155
6.5.3	Runtime Concurrency and Performance	156
6.6	Related Work	156
6.6.1	Architecture reconstruction	157
6.6.2	Hypothesis-driven approaches	157
6.6.3	Profiling and Visualization tools	158
6.6.4	Boot Performance Analysis Tools	159
6.7	Discussion and Open Issues	159
6.7.1	Proper Viewpoints and Metamodel	159
6.7.2	Availability of Sources of Runtime Information	160
6.7.3	Mapping Rules and Domain Knowledge	160
6.7.4	Open Issues	161
6.8	Conclusions	161
7	Conclusions	163
7.1	Answers to the Research Questions	163
7.2	Research Contribution and Extrapolation	167
7.3	Open Issues and Future Research	168
	Bibliography	171
	About the Author	183