

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/45620> holds various files of this Leiden University dissertation

Author: Nobakht, Behrooz

Title: Actors at work

Issue Date: 2016-12-15

Leiden University



Faculty of Science
Leiden Institute of Advanced Computer Science

Actors at Work

Actors at Work

Behrooz Nobakht

ACTORS AT WORK

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Leiden
op gezag van de Rector Magnificus prof. dr. C. J. J. M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op donderdag 15 december 2016
klokke 11.15 uur

door

Behrooz Nobakht
geboren te Tehran, Iran,
in 1981

Promotion Committee

Promotor: Prof. Dr. F.S. de Boer

Co-promotor: Dr. C. P. T. de Gouw

Other members:

Prof. Dr. F. Arbab

Dr. M.M. Bonsangue

Prof. Dr. E. B. Johnsen University of Oslo, Norway

Prof. Dr. M. Sirjani Reykjavik University, Iceland



The work reported in this thesis has been carried out at the Center for Mathematics and Computer Science (CWI) in Amsterdam and Leiden Institute of Advanced Computer Science at Leiden University. This research was supported by the European FP7-231620 project ENVISAGE on Engineering Virtualized Resources.

Copyright © 2016 by Behrooz Nobakht. All rights reserved.

October, 2016

Behrooz Nobakht

Actors at Work

Actors at Work, October, 2016

ISBN: 978-94-028-0436-2

Promotor: Prof. Dr. Frank S. de Boer

Cover Design: Ehsan Khakbaz <ehsan@khakbaz.com>

Built on 2016-11-02 17:00:24 +0100 from 397717ec11adfadec33e150b0264b0df83bdf37d at
<https://github.com/nobeh/thesis> using:

This is pdfTeX, Version 3.14159265-2.6-1.40.16 (TeX Live 2015/Debian) kpathsea version
6.2.1

Leiden University

Leiden Institute of Advanced Computer Science

Faculty of Science

Niels Bohrweg 1

2333 CA and Leiden

Contents

I	Introduction	1
1	Introduction	3
1.1	Objectives and Architecture	6
1.2	Literature Overview	8
1.2.1	Programming Languages	8
1.2.2	Frameworks and Libraries	9
1.3	Outline and Contributions	10
II	Programming Model	13
2	Application-Level Scheduling	15
2.1	Introduction	15
2.2	Application-Level Scheduling	17
2.3	Tool Architecture	19
2.3.1	A New Method Invocation	20
2.3.2	Scheduling the Next Method Invocation	21
2.3.3	Executing a Method Invocation	22
2.3.4	Extension Points	23
2.4	Case Study	23
2.5	Related Work	26
2.6	Conclusion	27
3	The Future of a Missed Deadline	29
3.1	Introduction	29
3.2	Programming with deadlines	30
3.2.1	Case Study: Fredhopper Distributed Data Processing	32
3.3	Operational Semantics	34
3.3.1	Local transition system	34
3.3.2	Global transition system	37
3.4	Implementation	38
3.5	Related Work	42
3.6	Conclusion and future work	44

III	Implementation	45
4	Programming with actors in Java 8	47
4.1	Introduction	47
4.2	Related Work	49
4.3	State of the Art: An example	50
4.4	Actor Programming in Java	52
4.5	Java 8 Features	53
4.6	Modeling actors in Java 8	55
4.7	Implementation Architecture	58
4.8	Experiments	61
4.9	Conclusion	63
IV	Application	65
5	Monitoring Method Call Sequences using Annotations	67
5.1	Introduction	67
5.2	Method Call Sequence Specification	70
5.3	Annotations for method call sequences	73
5.3.1	Sequenced Object Annotations	73
5.3.2	Sequenced Method Annotations	74
5.4	JMSeq by example	75
5.4.1	Sequenced Execution Specification	75
5.4.2	Exception Verification	76
5.5	The JMSeq framework	77
5.5.1	JMSeq Architecture	79
5.5.2	JUnit Support	83
5.6	The Fredhopper Access Server: A Case Study	85
5.6.1	Discussion	88
5.7	Performance Results	89
5.8	Related Work	91
5.9	Conclusion and future work	93
6	Formal verification of service level agreements through distributed monitoring	95
6.1	Introduction	95
6.2	Related Work	97
6.3	SDL Fredhopper Cloud Services	98
6.4	Distributed Monitoring Model	100
6.5	Service Characteristics Verification	104
6.6	Evaluation of the monitoring model	109
6.7	Future work	110

Bibliography	117
List of Figures	127
List of Tables	129

