



Universiteit
Leiden
The Netherlands

Combining Monitoring with Run-time Assertion Checking

Gouw, C.P.T. de

Citation

Gouw, C. P. T. de. (2013, December 18). *Combining Monitoring with Run-time Assertion Checking*. Retrieved from <https://hdl.handle.net/1887/22891>

Version: Corrected Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/22891>

Note: To cite this publication please use the final published version (if applicable).

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/22891> holds various files of this Leiden University dissertation

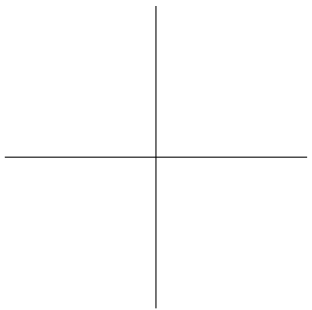
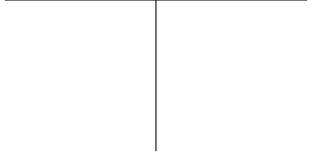
Author: Gouw, Stijn de

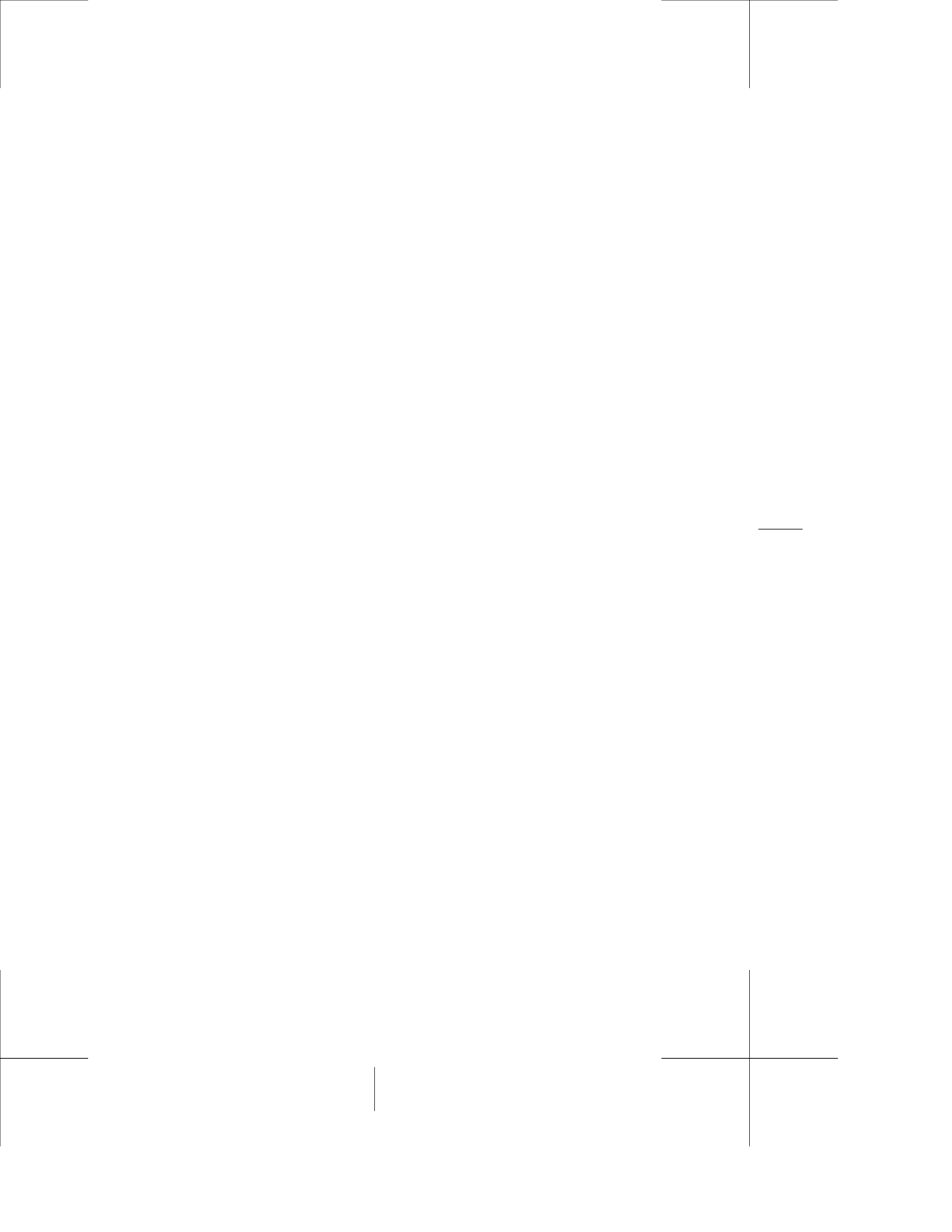
Title: Combining monitoring with run-time assertion checking

Issue Date: 2013-12-18

Combining Monitoring with Run-time Assertion Checking

Stijn de Gouw





Combining Monitoring with Run-time Assertion Checking

Proefschrift

ter verkrijging van

de graad van doctor aan de Universiteit Leiden

op gezag van de Rector Magnificus prof. mr. C.J.J.M. Stolker,

volgens besluit van het College voor Promoties

te verdedigen op woensdag 18 december 2013

klokke 10.00 uur

door

Stijn de Gouw
geboren te 's-Hertogenbosch, in 1985

PhD committee

Promotor: Prof. Dr. F.S. de Boer
Co-promotor: Dr. M. M. Bonsangue

Other members:

Prof. Dr. F. Arbab
Prof. Dr. J.N. Kok
Prof. Dr. O. Owe
Prof. Dr. K.R. Apt

University of Oslo, Norway
University of Amsterdam



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, under the auspices of the research school IPA (Institute for Programming research and Algorithmics). This research was supported by the European FP7-231620 project HATS on Highly Adaptable and Trustworthy Software using Formal Models.

Copyright © 2013 by Stijn de Gouw. All rights reserved.



Contents

1	Introduction	1
1.1	Prevention, Isolation and Fixing Bugs	1
	Type-Checking	2
	Static Verification	2
	Run-Time Checking	4
1.2	Object Orientation	5
1.3	Extension to Concurrency	8
	Outline	9
2	Specifying Object-Oriented Programs: Formalisms and Tools	13
3	Trace Specifications for Control- and Data-Flow	21
3.1	Modeling Framework	23
	Communication View	24
	Context-Free Grammars	28
	Attribute Grammars and Assertions	29
3.2	Discussion	34
4	Implementation	39
4.1	Instantiating the Tool Architecture	43
5	Case Studies	51
5.1	Design by Contract: Stack	52
5.2	Fredhopper Case-Study	57
5.3	Experiment	66
6	Concurrent Object Groups	71

6.1	Language	73
6.2	Semantics	81
6.3	Behavioral Interfaces for Coboxes	89
	Communication Views for COGs	90
	Attribute Grammars	93
6.4	Implementation	99
7	Related- and Future Work	103
	Expressiveness	105
	Learnability	108
	Future Work	110
A	Input and output of SAGA	113
	Samenvatting	121
	Curriculum Vitae	123
	Bibliography	125
	List of Figures	134
	List of Tables	136

