Universiteit
Leiden
The Netherlands

Asynchronous Programming in the Abstract Behavioural Specification Language

Azadbakht, K.

Citation
Azadbakht, K. (2019, December 11). *Asynchronous Programming in the Abstract Behavioural Specification Language*. Retrieved from https://hdl.handle.net/1887/81818

Version:	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/1887/81818

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

Cover Page

Universiteit Leiden

The handle http://hdl.handle.net/1887/81818 holds various files of this Leiden University dissertation.

Author: Azadbakht, K.
Title: Asynchronous Programming in the Abstract Behavioural Specification Language
Issue Date: 2019-12-11

Asynchronous Programming in the Abstract Behavioural Specification Language

Keyvan Azadbakht

Asynchronous Programming in the Abstract Behavioural Specification Language

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 11 december 2019
klokke 11:15 uur

door

Keyvan Azadbakht

geboren te Kohdasht, Iran
in 1987

PhD Committee

Promotor:		
	Prof. dr. F.S. de Boer	
Co-promotor:		
	Dr. E.P. de Vink	Eindhoven University of Technology
other members:		
	Prof. dr. ir. F. Arbab	
	Dr. M.M. Bonsangue	
	Dr. L. Henrio	CNRS, Paris
	Prof. dr. A. Plaat	
	Prof. dr. M. Sirjani	Malardalen University, Sweden

The work 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 (LIACS) at Leiden University. This research was supported by the European project FP7-612985 UpScale (From Inherent Concurrency to Massive Parallelism through Type-based Optimizations).

Cover design: Mostafa Dehghani

Contents

Prologue **1**

I Background: Abstract Behavioural Specification **3**

1 The ABS Language **5**
1.1 Introduction 5
1.2 ABS and Other Languages 7
1.3 Model of Concurrency 7
1.4 Language Definition 8
1.5 Distributed ABS 10
1.6 Example 11

II Case Study: Preferential Attachment in ABS **13**

2 Preferential Attachment on Multicore Systems **15**
2.1 Introduction 15
2.2 The Modeling Framework 16
2.3 Parallel Model of the PA 17
2.4 Related Work 22
2.5 Conclusion and Future Work 23

3 Preferential Attachment on Distributed Systems **25**
3.1 Introduction 25
3.2 Distributed PA 26
3.3 Implementation 30
3.4 Conclusion and Future Work 32

III Enhancing Parallelism **35**

4 Futures for Streaming Data **37**

4.1 Introduction . . . 37
4.2 Future-Based Data Streams . . . 39
4.3 Subject Reduction for the Extended ABS . . . 58
4.4 Data Streams in Distributed Systems . . . 60
4.5 Implementation . . . 61
4.6 Case Study . . . 67
4.7 Related Work . . . 68
4.8 Future work . . . 70

5 Multi-Threaded Actors 71
5.1 Introduction . . . 71
5.2 Motivating Example . . . 73
5.3 Syntax of MAC . . . 74
5.4 Operational Semantics . . . 77
5.5 Experimental Methodology and Implementation . . . 82
5.6 Conclusion and Future Work . . . 87

IV Deadlock Analysis 91

6 Deadlock Detection for Actor-Based Coroutines 93
6.1 Introduction . . . 93
6.2 The Programming Language . . . 95
6.3 The Concrete System . . . 97
6.4 The Abstract System . . . 100
6.5 Correctness of Predicate Abstraction . . . 103
6.6 Example . . . 104
6.7 Decidability of Deadlock Detection . . . 105
6.8 Conclusion . . . 109

Epilogue 111

Acknowledgements 113

Summary 115

Samenvatting 117

Prologue

This manuscript studies the Abstract Behavioural Specification (ABS), a formal language for designing executable models of parallel and distributed object-oriented systems [48]. ABS is defined in terms of a formal operational semantics which enables a variety of static and dynamic analysis techniques for ABS models, e.g., deadlock detection [14, 39], verification [30] and resource analysis [5].

The overall goal of this thesis is to extend the asynchronous programming model and the corresponding analysis techniques in ABS. Based on the different results, the thesis is structured as follows: Part I gives a preliminary overview of the ABS. In part II, we apply an extension of ABS with a notion of shared memory which preserves encapsulation to a case study, where we provide a parallel and distributed model of *preferential attachment* which is used to simulate large-scale social networks with certain mathematical properties. Encapsulation is preserved by a single-write policy. In Part III, we formally extend ABS to enhance both asynchronous programming by data streaming between processes, and parallelism by multi-threading within an actor. Finally in part IV, a new technique based on predicate abstraction is introduced to analyze the ABS models for the absence of deadlock within an actor.

Validation. This work has been carried out in the context of the UpScale Project, an EU-funded project where the vision was:

> to provide programming language support to efficiently develop applications that seamlessly scale to the available parallelism of manycore chips without abandoning the object-oriented paradigm and the associated software engineering methodologies.

In particular, the above extension of ABS concerning streaming of data has been validated by the case study on *preferential attachment* for the efficient multicore and distributed simulation of large-scale social networks. The results of this thesis have been separately validated by the peer-reviewed scientific publications listed in Table 1.

Table 1: List of publications used in the thesis.

Publication	Chapter
Azadbakht, Keyvan, et al. "A high-level and scalable approach for generating scale-free graphs using active objects." Proceedings of the 31st Annual ACM Symposium on Applied Computing. ACM, 2016.	2
Azadbakht, Keyvan, Nikolaos Bezirgiannis, and Frank S. de Boer. "Distributed network generation based on preferential attachment in ABS." International Conference on Current Trends in Theory and Practice of Informatics. Springer, Cham, 2017.	3
Azadbakht, Keyvan, Nikolaos Bezirgiannis, and Frank S. de Boer. "On Futures for Streaming Data in ABS." International Conference on Formal Techniques for Distributed Objects, Components, and Systems. Springer, Cham, 2017.	4
Azadbakht, Keyvan, Frank S. de Boer, Nikolaos Bezirgiannis, and Erik de Vink. "A formal actor-based model for streaming the future." Science of Computer Programming 186 (2019): 102341.	4
Azadbakht, K., Frank S. de Boer, Vlad Serbanescu. "Multi-threaded actors." In: Proceedings 9th Interaction and Concurrency Experience, ICE 2016, Heraklion, Greece, volume 223 of EPTCS, pp. 51–66 2016.	5
Azadbakht, Keyvan, Frank S. de Boer, and Erik de Vink. "Deadlock Detection for Actor-Based Coroutines." International Symposium on Formal Methods. Springer, Cham, 2018.	6