Lecture Notes in Computer Science 10789

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

More information about this series at http://www.springer.com/series/7408

Alessandro Ricci · Philipp Haller (Eds.)

Programming with Actors

State-of-the-Art and Research Perspectives

Springer

Editors
Alessandro Ricci
University of Bologna
Cesena
Italy

Philipp Haller
KTH Royal Institute of Technology
Stockholm
Sweden

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-030-00301-2 ISBN 978-3-030-00302-9 (eBook)
https://doi.org/10.1007/978-3-030-00302-9

Library of Congress Control Number: 2018954065

LNCS Sublibrary: SL2 – Programming and Software Engineering

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

More than 40 years after their invention, *actors* have become a common reference model for designing and developing concurrent and distributed systems.

The actor model was introduced by Carl Hewitt and colleagues in 1973 [7], as a mathematical model of concurrent computation in which *actors* play the role of universal primitives of concurrent computation. In response to a message that it receives, an actor can: make local decisions, create more actors, send more messages, and determine how to respond to the next message received. Actors may modify their own private state, but can only affect each other through messages (avoiding the need for locks or other synchronization mechanisms). Since its conception, the model served both as a framework for a theoretical understanding of computation and as the theoretical basis for several practical implementations of concurrent systems [6].

In the 1980s and 1990s the model helped researchers understand and define the extension of object-oriented programming towards concurrency, e.g., in the form of concurrent object-oriented programming [1, 2, 4, 11].

In the mainstream, meanwhile, the panorama was dominated by sequential programming until the middle of the 2000s, when the "concurrency revolution" began [5, 9]. Since then, concurrent, asynchronous, and distributed programming have gradually become part of everyday design and programming.

If the 1980s and 1990s were dominated by a vision in which mainstream programming and programming paradigms could abstract from concurrency and distribution, in recent years there has been an increasing awareness that this is not feasible (e.g., when building reactive applications[1]), and – moreover – first-class concurrency abstractions, such actors and message passing, provide effective modeling and designing power to deal with the complexity of modern applications and application domains [3, 10].

The AGERE! workshop started in 2011 at the SPLASH conference to investigate the definition of suitable levels of abstraction, programming languages, and platforms to support and promote a decentralized mindset in solving problems, designing systems, as well as programming applications, including the teaching of computer programming [8]. That is, the question is how to think about problems and programs embracing decentralization of control and interaction as the most essential features. To this end, actors and agents were taken as key references, recognized as two main broad families of concepts, abstractions, and programming tools described in the literature, which explicitly promote such decentralized thinking.

The set of papers collected in this issue originated from the AGERE! workshop series – the last edition was held in 2017 – and concern the application of actor-based approaches to mainstream application domains and the discussion of related issues.

[1] For example, see https://www.reactivemanifesto.org.

The issue is divided into two parts. The first part concerns selected application domains:

- *Web Programming* – Parallel and Distributed Web Programming with Actors by Florian Myter, Christophe Scholliers, and Wolfgang De Meuter
- *Data-Intensive Parallel Programming* — OpenCL Actors: Adding Data Parallelism to Actor-Based Programming with CAF by Raphael Hiesgen, Dominik Charousset, and Thomas Schmidt
- *Mobile Computing* — AmbientJS: A Mobile Cross-Platform Actor Library for Multi-networked Mobile Applications by Elisa Gonzalez Boix, Kevin De Porre, Wolfgang De Meuter, and Christophe Scholliers
- *Self-Organizing Systems* — Programming Actor-Based Collective Adaptive Systems by Roberto Casadei and Mirko Viroli

The second part concerns selected issues:

- *Scheduling* — Pluggable Scheduling for the Reactor Programming Model by Aleksandar Prokopec
- *Debugging* — A Study of Concurrency Bugs and Advanced Development Support for Actor-Based Programs by Carmen Torres Lopez, Stefan Marr, Hanspeter Mössenböck, and Elisa Gonzalez Boix
- *Communication and Coordination* — A Model for Separating Communication Concerns of Concurrent Systems by Hongxing Geng and Nadeem Jamali
- *Monitoring* — A Homogeneous Actor-Based Monitor Language for Adaptive Behavior by Tony Clark, Vinay Kulkarni, Souvik Barat, and Balbir Barn

References

1. G. Agha. Concurrent object-oriented programming. Commun. ACM, 33:125–141, September 1990.
2. G. Agha, P. Wegner, and A. Yonezawa, editors. Research directions in concurrent object-oriented programming. MIT Press, Cambridge, MA, USA, 1993.
3. J. Armstrong. Erlang. Commun. ACM, 53(9):68–75, Sept. 2010.
4. J.-P. Briot, R. Guerraoui, and K.-P. Lohr. Concurrency and distribution in object-oriented programming. ACM Comput. Surv., 30(3):291–329, Sept. 1998.
5. K. B. Bruce, A. Danyluk, and T. Murtagh. Introducing concurrency in CS 1. In Proceedings of the 41st ACM Technical Symposium on Computer Science Education, SIGCSE '10, pages 224–228, New York, NY, USA, 2010. ACM.
6. C. Hewitt. Viewing control structures as patterns of passing messages. Artif. Intell., 8(3):323–364, 1977.
7. C. Hewitt, P. Bishop, and R. Steiger. A universal modular actor formalism for artificial intelligence. In Proceedings of the 3rd International Joint Conference on Artificial Intelligence, IJCAI'73, pages 235–245, San Francisco, CA, USA, 1973. Morgan Kaufmann Publishers Inc.

8. A. Ricci, R. H. Bordini, and G. Agha. AGERE! (Actors and aGEnts REloaded): SPLASH 2011 workshop on programming systems, languages and applications based on actors, agents and decentralized control. In Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion, OOPSLA '11, pages 325–326, New York, NY, USA, 2011. ACM.
9. H. Sutter and J. Larus. Software and the concurrency revolution. ACM Queue: Tomorrow's Computing Today, 3(7):54–62, Sept. 2005.
10. V. Vernon. Reactive Messaging Patterns with the Actor Model: Applications and Integration in Scala and Akka. Addison-Wesley Professional, 1st edition, 2015.
11. A. Yonezawa and M. Tokoro, editors. Object-oriented concurrent programming, Cambridge, MA, USA, 1987. MIT Press.

June 2018

Alessandro Ricci
Philipp Haller

Contents

Actors and Programming – Selected Domains

Parallel and Distributed Web Programming with Actors 3
Florian Myter, Christophe Scholliers, and Wolfgang De Meuter

AmbientJS: A Mobile Cross-Platform Actor Library for Multi-Networked Mobile Applications 32
Elisa Gonzalez Boix, Kevin De Porre, Wolfgang De Meuter, and Christophe Scholliers

OpenCL Actors – Adding Data Parallelism to Actor-Based Programming with CAF. 59
Raphael Hiesgen, Dominik Charousset, and Thomas C. Schmidt

Programming Actor-Based Collective Adaptive Systems 94
Roberto Casadei and Mirko Viroli

Actors and Programming – Selected Issues

Pluggable Scheduling for the Reactor Programming Model 125
Aleksandar Prokopec

A Study of Concurrency Bugs and Advanced Development Support for Actor-based Programs. 155
Carmen Torres Lopez, Stefan Marr, Elisa Gonzalez Boix, and Hanspeter Mössenböck

interActors: A Model for Separating Communication Concerns of Concurrent Systems. 186
Hongxing Geng and Nadeem Jamali

A Homogeneous Actor-Based Monitor Language for Adaptive Behaviour . . . 216
Tony Clark, Vinay Kulkarni, Souvik Barat, and Balbir Barn

Author Index 245