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

Bibliography

[1] Gul Agha. Actors: A model of concurrent computation in distributed systems. Technical report, DTIC Document, 1985.

[2] Gul Agha. *Actors: A Model of Concurrent Computation in Distributed Systems*. The MIT Press, 1986.

[3] Wolfgang Ahrendt, Bernhard Beckert, Richard Bubel, Reiner Hähnle, Peter H. Schmitt, and Mattias Ulbrich, editors. *Deductive Software Verification - The KeY Book – From Theory to Practice*, volume 10001 of *Lecture Notes in Computer Science*. Springer, 2016.

[4] Maksudul Alam, Maleq Khan, and Madhav V Marathe. Distributed-memory parallel algorithms for generating massive scale-free networks using preferential attachment model. In *Proceedings of SC13: International Conference for High Performance Computing, Networking, Storage and Analysis*, page 91. ACM, 2013.

[5] Elvira Albert, Puri Arenas, Jesús Correas, Samir Genaim, Miguel Gómez-Zamalloa, Germán Puebla, and Guillermo Román-Díez. Object-sensitive cost analysis for concurrent objects. *Software Testing, Verification and Reliability*, 25(3):218–271, 2015.

[6] Joe Armstrong. *Making reliable distributed systems in the presence of software errors*. PhD thesis, Mikroelektronik och informationsteknik, 2003.

[7] Joe Armstrong, Robert Virding, Claes Wikström, and Mike Williams. Concurrent programming in erlang. 1993.

[8] Krste Asanovic, Ras Bodik, Bryan Christopher Catanzaro, Joseph James Gebis, Parry Husbands, Kurt Keutzer, David A Patterson, William Lester Plishker, John Shalf, Samuel Webb Williams, et al. The landscape of parallel computing research: A view from berkeley. Technical report, Technical Report UCB/EECS-2006-183, EECS Department, University of California, Berkeley, 2006.

[9] James Atwood, Bruno Ribeiro, and Don Towsley. Efficient network generation under general preferential attachment. *Computational Social Networks*, 2(1):1, 2015.

[10] Keyvan Azadbakht, Nikolaos Bezirgiannis, and Frank S de Boer. Distributed network generation based on preferential attachment in abs. In *International Conference on Current Trends in Theory and Practice of Informatics*, pages 103–115. Springer, 2017.

[11] Keyvan Azadbakht, Nikolaos Bezirgiannis, and Frank S de Boer. On futures for streaming data in ABS. In *International Conference on Formal Techniques for Distributed Objects, Components, and Systems*, pages 67–73. Springer, 2017.

[12] Keyvan Azadbakht, Nikolaos Bezirgiannis, Frank S de Boer, and Sadegh Aliakbary. A high-level and scalable approach for generating scale-free graphs using active objects. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pages 1244–1250. ACM, 2016.

[13] Keyvan Azadbakht, Frank S de Boer, Nikolaos Bezirgiannis, and Erik de Vink. A formal actor-based model for streaming the future. *Science of Computer Programming*, 186:102341, 2019.

[14] Keyvan Azadbakht, Frank S. de Boer, and Erik de Vink. Deadlock detection for actor-based coroutines. In Klaus Havelund, Jan Peleska, Bill Roscoe, and Erik de Vink, editors, *Formal Methods*, pages 39–54, Cham, 2018. Springer International Publishing.

[15] Keyvan Azadbakht, Frank S. de Boer, and Vlad Serbanescu. Multi-threaded actors. In *Proceedings 9th Interaction and Concurrency Experience, ICE 2016*, volume 223 of *EPTCS*, pages 51–66, 2016.

[16] David Bader, Kamesh Madduri, et al. Parallel algorithms for evaluating centrality indices in real-world networks. In *Parallel Processing, 2006. ICPP 2006. International Conference on*, pages 539–550. IEEE, 2006.

[17] Thomas Ball, Rupak Majumdar, Todd Millstein, and Sriram K. Rajamani. Automatic predicate abstraction of C programs. In *Conference on Programming Language Design and Implementation*, pages 203–213, 2001.

[18] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[19] Vladimir Batagelj and Ulrik Brandes. Efficient generation of large random networks. *Physical Review E*, 71(3):036113, 2005.

[20] Nikolaos Bezirgiannis and Frank de Boer. ABS: a high-level modeling language for cloud-aware programming. In *International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, pages 433–444. Springer, 2016.

[21] Stephan Brandauer, Elias Castegren, Dave Clarke, Kiko Fernandez-Reyes, Einar Broch Johnsen, Ka I Pun, S Lizeth Tapia Tarifa, Tobias Wrigstad, and Albert Mingkun Yang. Parallel objects for multicores: A glimpse at the parallel language encore. In *Formal Methods for Multicore Programming*, pages 1–56. Springer, 2015.

[22] Denis Caromel. Keynote 1-strong programming model for strong weak mobility: The proactive parallel suite. In *Mobile Data Management, 2008. MDM'08. 9th International Conference on*, pages xvi–xvi. IEEE, 2008.

[23] Deepayan Chakrabarti, Yiping Zhan, and Christos Faloutsos. R-mat: A recursive model for graph mining. In *SDM*, volume 4, pages 442–446. SIAM, 2004.

[24] Melvin E. Conway. Design of a separable transition-diagram compiler. *Commununications of the ACM*, 6(7):396–408, 1963.

[25] Leonardo Dagum and Rameshm Enon. Openmp: an industry standard api for shared-memory programming. *Computational Science & Engineering, IEEE*, 5(1):46–55, 1998.

[26] Frank S. de Boer, Mario Bravetti, Immo Grabe, Matias David Lee, Martin Steffen, and Gianluigi Zavattaro. A Petri net based analysis of deadlocks for active objects and futures. In *FACS*, volume 7684, pages 110–127. Springer, 2012.

[27] Frank S De Boer, Dave Clarke, and Einar Broch Johnsen. A complete guide to the future. In *Programming Languages and Systems*, pages 316–330. Springer, 2007.

[28] Frank S. de Boer and Stijn de Gouw. Being and change: Reasoning about invariance. In *Correct System Design - Symposium in Honor of Ernst-Rüdiger Olderog on the Occasion of His 60th Birthday, Oldenburg, Germany, September 8-9, 2015. Proceedings*, pages 191–204, 2015.

[29] Frank S. de Boer, Mohammad Mahdi Jaghoori, Cosimo Laneve, and Gianluigi Zavattaro. Decidability problems for actor systems. In *CONCUR 2012 - Concurrency Theory - 23rd International Conference, CONCUR 2012, Newcastle upon Tyne, UK, September 4-7, 2012. Proceedings*, pages 562–577, 2012.

[30] Crystal Chang Din, Richard Bubel, and Reiner Hähnle. Key-abs: A deductive verification tool for the concurrent modelling language abs. In *International Conference on Automated Deduction*, pages 517–526. Springer International Publishing, 2015.

[31] Johan Dovland, Einar Broch Johnsen, and Olaf Owe. Verification of concurrent objects with asynchronous method calls. In *2005 IEEE International Conference on Software - Science, Technology and Engineering (SwSTE 2005), 22-23 February 2005, Herzelia, Israel*, pages 141–150, 2005.

[32] Jeff Epstein, Andrew P Black, and Simon Peyton-Jones. Towards haskell in the cloud. In *ACM SIGPLAN Notices*, volume 46, pages 118–129. ACM, 2011.

[33] Paul Erdös and Alfréd Rényi. On the central limit theorem for samples from a finite population. *Publ. Math. Inst. Hungar. Acad. Sci*, 4:49–61, 1959.

[34] Patrick Th Eugster, Pascal A Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. *ACM computing surveys (CSUR)*, 35(2):114–131, 2003.

[35] Kiko Fernandez-Reyes, Dave Clarke, and Daniel S. McCain. Part: An asynchronous parallel abstraction for speculative pipeline computations. In *COORDINATION*. To appear, 2016.

[36] Alain Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1):63–92, 2001.

[37] David Gelernter. Generative communication in linda. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 7(1):80–112, 1985.

[38] Elena Giachino, Ludovic Henrio, Cosimo Laneve, and Vincenzo Mastandrea. Actors may synchronize, safely! In *18th International Symposium on Principles and Practice of Declarative Programming*, pages 118–131, 2016.

[39] Elena Giachino, Cosimo Laneve, and Michael Lienhardt. A framework for deadlock detection in core ABS. *Software & Systems Modeling*, 15(4):1013–1048, 2016.

[40] Susanne Graf and Hassen Saïdi. Construction of abstract state graphs with PVS. In *International Conference on Computer Aided Verification*, pages 72–83. Springer, 1997.

[41] Douglas Gregor and Andrew Lumsdaine. The parallel bgl: A generic library for distributed graph computations. *Parallel Object-Oriented Scientific Computing (POOSC)*, page 2, 2005.

[42] Philipp Haller and Martin Odersky. Scala actors: Unifying thread-based and event-based programming. *Theoretical Computer Science*, 410(2):202–220, 2009.

[43] Max Haustein and Klaus-Peter Löhr. Jac: declarative java concurrency. *Concurrency and Computation: Practice and Experience*, 18(5):519–546, 2006.

[44] Ludovic Henrio, Fabrice Huet, and Zsolt István. Multi-threaded active objects. In *COORDINATION*, pages 90–104. Springer, 2013.

[45] Ludovic Henrio, Cosimo Laneve, and Vincenzo Mastandrea. Analysis of synchronisations in stateful active objects. In *International Conference on Integrated Formal Methods*, pages 195–210, 2017.

[46] Ludovic Henrio and Justine Rochas. From modelling to systematic deployment of distributed active objects. In *International Conference on Coordination Languages and Models*, pages 208–226. Springer, 2016.

[47] Carl Hewitt. Description and theoretical analysis (using schemata) of planner: A language for proving theorems and manipulating models in a robot. Technical report, Massachusetts Inst of Tech Cambridge Artificial Intelligence Lab, 1972.

[48] Einar Broch Johnsen, Reiner Hähnle, Jan Schäfer, Rudolf Schlatte, and Martin Steffen. ABS: A core language for abstract behavioral specification. In *International Symposium on Formal Methods for Components and Objects*, pages 142–164. Springer, 2010.

[49] Einar Broch Johnsen and Olaf Owe. An asynchronous communication model for distributed concurrent objects. *Software & Systems Modeling*, 6(1):39–58, 2007.

[50] Einar Broch Johnsen, Olaf Owe, Dave Clarke, and Joakim Bjørk. A formal model of service-oriented dynamic object groups. *Science of Computer Programming*, 115:3–22, 2016.

[51] Einar Broch Johnsen, Olaf Owe, and Ingrid Chieh Yu. Creol: A type-safe object-oriented model for distributed concurrent systems. *Theoretical Computer Science*, 365(1-2):23–66, 2006.

[52] Eduard Kamburjan, Crystal Chang Din, and Tzu-Chun Chen. Session-based compositional analysis for actor-based languages using futures. In *International Conference on Formal Engineering Methods*, pages 296–312, 2016.

[53] Eric Kerfoot, Steve McKeever, and Faraz Torshizi. Deadlock freedom through object ownership. In *5th International Workshop on Aliasing, Confinement and Ownership in Object-Oriented Programming*, 2009.

[54] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, D Sivakumar, Andrew Tomkins, and Eli Upfal. Stochastic models for the web graph. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 57–65. IEEE, 2000.

[55] R Greg Lavender and Douglas C Schmidt. Active object–an object behavioral pattern for concurrent programming. 1995.

[56] Jurij Leskovec. *Dynamics of large networks*. ProQuest, 2008.

[57] Kai Li and Paul Hudak. Memory coherence in shared virtual memory systems. *ACM Transactions on Computer Systems (TOCS)*, 7(4):321–359, 1989.

[58] Yi-Chen Lo, Hung-Che Lai, Cheng-Te Li, and Shou-De Lin. Mining and generating large-scaled social networks via mapreduce. *Social Network Analysis and Mining*, 3(4):1449–1469, 2013.

[59] Yi-Chen Lo, Cheng-Te Li, and Shou-De Lin. Parallelizing preferential attachment models for generating large-scale social networks that cannot fit into memory. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 229–238. IEEE, 2012.

[60] Sadegh Nobari, Xuesong Lu, Panagiotis Karras, and Stéphane Bressan. Fast random graph generation. In *Proceedings of the 14th International Conference on Extending Database Technology*, pages 331–342. ACM, 2011.

[61] Gordon D Plotkin. A structural approach to operational semantics. 1981.

[62] John C Reynolds. The discoveries of continuations. *Lisp and symbolic computation*, 6(3-4):233–247, 1993.

[63] Jan Schäfer and Arnd Poetzsch-Heffter. Jcobox: Generalizing active objects to concurrent components. In *ECOOP 2010–Object-Oriented Programming*. Springer, 2010.

[64] Jan Schäfer and Arnd Poetzsch-Heffter. Jcobox: Generalizing active objects to concurrent components. In *ECOOP 2010–Object-Oriented Programming*, pages 275–299. Springer, 2010.

[65] Christophe Scholliers, Éric Tanter, and Wolfgang De Meuter. Parallel actor monitors: Disentangling task-level parallelism from data partitioning in the actor model. *Science of Computer Programming*, 80:52–64, 2014.

[66] Yury Selivanov. Asynchronous generators. `https://www.python.org/dev/peps/pep-0525/`, 2016.

[67] Koushik Sen and Mahesh Viswanathan. Model checking multithreaded programs with asynchronous atomic methods. In *International Conference on Computer Aided Verification*, pages 300–314. Springer, 2006.

[68] Vlad Şerbănescu, Keyvan Azadbakht, and Frank de Boer. A java-based distributed approach for generating large-scale social network graphs. In *Resource Management for Big Data Platforms*, pages 401–417. Springer, 2016.

[69] Marjan Sirjani. Rebeca: Theory, applications, and tools. In *Formal Methods for Components and Objects*, pages 102–126. Springer, 2007.

[70] Streams. Version 2.4.17. `http://doc.akka.io/docs/akka/2.4/scala/stream/index.html`, 2017.

[71] Herb Sutter. The free lunch is over: A fundamental turn toward concurrency in software. *Dr. Dobb's journal*, 30(3):202–210, 2005.

[72] Ming-Chit Tam, Jonathan M Smith, and David J Farber. A taxonomy-based comparison of several distributed shared memory systems. *ACM SIGOPS Operating Systems Review*, 24(3):40–67, 1990.

[73] Roberto Tonelli, Giulio Concas, and Mario Locci. Three efficient algorithms for implementing the preferential attachment mechanism in yule-simon stochastic process. *WSEAS Transactions on Information Science and Applications*, 7(2):176–185, 2010.

[74] Thierry Van Cutsem, Stijn Mostinckx, Elisa Gonzalez Boix, Jessie Dedecker, and Wolfgang De Meuter. Ambienttalk: object-oriented event-driven programming in mobile ad hoc networks. In *Chilean Society of Computer Science, 2007. SCCC'07. XXVI International Conference of the*, pages 3–12. IEEE, 2007.

[75] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

[76] Peter Welch and Neil Brown. Communicating sequential processes for java^{tm} (jcsp). `https://www.cs.kent.ac.uk/projects/ofa/jcsp/`, 2014.

[77] Antonty S Williams, Alexander A Mitchell, Robert G Atkinson, C Douglas Hodges, Johann Posch, and Craig H Wittenberg. Method and system for multithreaded processing, January 30 2001. US Patent 6,182,108.

[78] Akinori Yonezawa and Mario Tokoro. Object-oriented concurrent programming. 1986.

[79] Andy Yoo and Keith Henderson. Parallel generation of massive scale-free graphs. *arXiv preprint arXiv:1003.3684*, 2010.