Din, Crystal Chang; **Owe, Olaf**

Compositional reasoning about active objects with shared futures. (English) Zbl 1343.68166

Formal Asp. Comput. 27, No. 3, 551-572 (2015).

Summary: Distributed and concurrent object-oriented systems are difficult to analyze due to the complexity of their concurrency, communication, and synchronization mechanisms. The *future mechanism* extends the traditional method call communication model by facilitating sharing of references to futures. By assigning method call result values to futures, third party objects may pick up these values. This may reduce the time spent waiting for replies in a distributed environment. However, futures add a level of complexity to program analysis, as the program semantics becomes more involved. This paper presents a model for asynchronously communicating objects, where return values from method calls are handled by futures. The model facilitates invariant specifications over the locally visible communication history of each object. Compositional reasoning is supported and proved sound, as each object may be specified and verified independently of its environment. A kernel object-oriented language with futures inspired by the ABS modeling language is considered. A compositional proof system for this language is presented, formulated within dynamic logic.

MSC:

68Q85 Models and methods for concurrent and distributed computing (process algebras, bisimulation, transition nets, etc.)
68Q55 Semantics in the theory of computing
68Q60 Specification and verification (program logics, model checking, etc.)

Cited in **2** Documents

Keywords:

distributed systems; object orientation; concurrent objects; asynchronous communication; shared futures; operational semantics; communication history; compositional reasoning; dynamic logic

Software:

Java Jr; Multilisp; Eiffel; Maude; MapReduce

Full Text: DOI

References:

[1] Ahrendt W, Dylla M (2012) A system for compositional verification of asynchronous objects. Sci Comput Program. 77(12):1289-1309. doi:10.1016/j.scico.2010.08.003 · Zbl 1264.68050

[2] Agha, G; Frølund, S; Kim, WY; Panwar, R; Patterson, A; Sturman, D, Abstraction and modularity mechanisms for concurrent computing, Parallel Distrib Technol Syst Appl IEEE, 1, 3-14, (1993)· doi:10.1109/88.218170

[3] Ábrahám, E; Grabe, I; Grüner, A; Steffen, M, Behavioral interface description of an object-oriented language with futures and promises, J Log Algebr Program, 78, 491-518, (2009) · Zbl 1187.68130 · doi:10.1016/j.jlap.2009.01.001

[4] Alpern, B; Schneider, FB, Defining liveness, Inf Process Lett, 21, 181-185, (1985) · Zbl 0575.68030 · doi:10.1016/0020-0190(85)90056-0

[5] Ahern A, Yoshida N (2007) Formalising java rmi with explicit code mobility. Theor Comput Sci 389(3):341-410. Semantic and Logical Foundations of Global Computing · Zbl 1132.68020

[6] Beckert B, Hähnle R, Schmitt PH (eds) (2007) Verification of object-oriented software: the KeY approach. LNCS, vol 4334. Springer, Berlin · Zbl 0758.68043

[7] Baker Jr HG, Hewitt C (1977) The incremental garbage collection of processes. In: Proceedings of the 1977 symposium on artificial intelligence and programming languages, New York, NY, USA. ACM, pp 55-59

[8] Brooke, PJ; Paige, RF, Cameo: an alternative model of concurrency for eiffel, Form Asp Comput, 21, 363-391, (2009) · Zbl 1184.68150 · doi:10.1007/s00165-008-0096-1

[9] Broy M, Stølen K (2001) Specification and development of interactive systems. Monographs in computer science. Springer · Zbl 0981.68115 · doi:10.1007/978-1-4613-0091-5

[10] Clavel M, Durán F, Eker S, Lincoln P, Martí-Oliet N, Meseguer J, Talcott CL (2007) All about Maude—a high-performance logical framework, how to specify, program and verify systems in rewriting logic. LNCS, vol 4350. Springer, Berlin · Zbl 1115.68046

[11] Dahl O-J (1977) Can program proving be made practical? In: Amirchahy M, Néel D (eds) Les Fondements de la Program-

mation. Institut de Recherche d'Informatique et d'Automatique, Toulouse, France, December 1977, pp 57-114

[12] Dahl O-J (1987) Object-oriented specifications. In: Research directions in object-oriented programming. MIT Press, Cambridge, pp 561-576 · Zbl 1187.68130

[13] Dahl O-J (1992) Verifiable programming. International series in computer science. Prentice Hall, New York · Zbl 0790.68005

[14] Boer, FS, A Hoare logic for dynamic networks of asynchronously communicating deterministic processes, Theor Comput Sci, 274, 3-41, (2002) · Zbl 0992.68026 · doi:10.1016/S0304-3975(00)00304-2

[15] de Boer FS, Clarke D, Johnsen EB (2007) A complete guide to the future. In: de Nicola R (ed) Proceedings of the 16th European symposium on programming (ESOP'07), March 2007. LNCS, vol 4421. Springer, Berlin, pp 316-330

[16] Din, CC; Dovland, J; Johnsen, EB; Owe, O, Observable behavior of distributed systems: component reasoning for concurrent objects, J Log Algebr Program, 81, 227-256, (2012) · Zbl 1247.68184 · doi:10.1016/j.jlap.2012.01.003

[17] Din CC, Dovland J, Owe O (2012) An approach to compositional reasoning about concurrent objects and futures. Research Report 415, Department of Informatics, University of Oslo, February 2012. http://urn.nb.no/URN:NBN:no-30589 · Zbl 1315.68192

[18] Din CC, Dovland J, Owe O (2012) Compositional reasoning about shared futures. In: Eleftherakis G, Hinchey M, Holcombe M (eds) Proceedings of the international conference on software engineering and formal methods (SEFM'12). LNCS, vol 7504. Springer, Berlin, pp 94-108 · Zbl 1315.68192

[19] Dean, J; Ghemawat, S, Mapreduce: simplified data processing on large clusters, Commun. ACM, 51, 107-113, (2008)· doi:10.1145/1327452.1327492

[20] Dovland J, Johnsen EB, Owe O (2005) Verification of concurrent objects with asynchronous method calls. In: Proceedings of the IEEE international conference on software science, technology and engineering (SwSTE'05), February 2005. IEEE Computer Society Press, pp 141-150

[21] Dahl O-J, Owe O (1998) Formal methods and the RM-ODP. Research Report 261, Department of Informatics, University of Oslo, Norway, May 1998

[22] de Roever W-P, de Boer F, Hannemann U, Hooman J, Lakhnech Y, Poel M, Zwiers J (2001) Concurrency verification: introduction to compositional and noncompositional methods. Cambridge University Press, New York · Zbl 1009.68020

[23] Falkner KEK, Coddington PD, Oudshoorn MJ (1999) Implementing asynchronous remote method invocation in java · Zbl 0961.68511

[24] Full ABS Modeling Framework (2011). Deliverable 1.2 of project FP7-231620 (HATS). http://www.hats-project.eu

[25] Halstead, RH, Multilisp: a language for concurrent symbolic computation, ACM Trans Program Lang Syst, 7, 501-538, (1985) · Zbl 0581.68037 · doi:10.1145/4472.4478

[26] Hoare CAR (1985) Communicating sequential processes. International series in computer science. Prentice Hall, Englewood Cliffs · Zbl 0637.68007

[27] International Telecommunication Union (1995) Open distributed processing-reference model parts 1-4. Technical report, ISO/IEC, Geneva, July 1995

[28] Johnsen EB, Owe O (2004) Object-oriented specification and open distributed systems. In: Owe O, Krogdahl S, Lyche T (eds) From object-orientation to formal methods: essays in memory of Ole-Johan Dahl. LNCS, vol 2635. Springer, Berlin, pp 137-164 · Zbl 0543.68010

[29] Johnsen, EB; Owe, O, An asynchronous communication model for distributed concurrent objects, Softw Syst Model, 6, 35-58, (2007)· doi:10.1007/s10270-006-0011-2

[30] Jeffrey ASA, Rathke J (2005) Java Jr.: fully abstract trace semantics for a core Java language. In: Proceedings of the European symposium on programming. LNCS, vol 3444. Springer, Berlin, pp 423-438 · Zbl 1108.68349

[31] Liskov BH, Shrira L (1988) Promises: linguistic support for efficient asynchronous procedure calls in distributed systems. In: Wise DS (ed) Proceedings of the SIGPLAN conference on programming language design and implementation (PLDI'88). ACM Press, pp 260-267

[32] Morandi B, Bauer SS, Meyer B (2008) SCOOP—a contract-based concurrent object-oriented programming model. In: Müller P (ed) Advanced lectures on software engineering, LASER Summer School 2007/2008. Lecture notes in computer science, vol 6029. Springer, Berlin, pp 41-90

[33] Meseguer, J, Conditional rewriting logic as a unified model of concurrency, Theor Comput Sci, 96, 73-155, (1992) · Zbl 0758.68043 · doi:10.1016/0304-3975(92)90182-F

[34] Meyer, B, Systematic concurrent object-oriented programming, Commun. ACM, 36, 56-80, (1993)· doi:10.1145/162685.162705

[35] Meyer B (1997) Object-oriented software construction. 2nd edn Prentice-Hall, Inc. · Zbl 0987.68516

[36] Soundararajan, N, Axiomatic semantics of communicating sequential processes, ACM Trans Program Lang Syst, 6, 647-662, (1984) · Zbl 0542.68013 · doi:10.1145/1780.1805

[37] Soundararajan, N, A proof technique for parallel programs, Theor Comput Sci, 31, 13-29, (1984) · Zbl 0543.68010 · doi:10.1016/0304-3975(84)90122-1

[38] Yonezawa A, Briot J-P, Shibayama E (1986) Object-oriented concurrent programming in ABCL/1. In: Conference on object-oriented programming systems, languages and applications (OOPSLA'86). Sigplan Notices, vol 21, no 11, pp 258-268, November 1986