

Poetzsch-Heffter, Arnd; Schäfer, Jan

A representation-independent behavioral semantics for object-oriented components. (English) [Zbl 1202.68100](#)

Bonsangue, Marcello M. (ed.) et al., Formal methods for open object-based distributed systems. 9th IFIP WG 6.1 international conference FMOODS 2007, Paphos, Cyprus, June 6–8, 2007. Proceedings. Berlin: Springer (ISBN 978-3-540-72919-8/pbk). Lecture Notes in Computer Science 4468, 157-173 (2007).

Summary: Behavioral semantics abstracts from implementation details and allows to describe the behavior of software components in a representation-independent way. In this paper, we develop a formal behavioral semantics for class-based object-oriented languages with aliasing, subclassing, and dynamic dispatch. The code of an object-oriented component consists of a class and the classes used by it. A component instance is realized by a dynamically evolving set of objects with a clear boundary to the environment. The behavioral semantics is expressed in terms of the messages crossing the boundary. It is defined as an abstraction of an operational semantics based on an ownership-structured heap. We show how the semantics can be used to define substitutability in a program independent way.

For the entire collection see [\[Zbl 1118.68002\]](#).

MSC:

- [68N19](#) Other programming paradigms (object-oriented, sequential, concurrent, automatic, etc.) Cited in **3** Documents
- [68N15](#) Theory of programming languages
- [68Q55](#) Semantics in the theory of computing

Software:

[Featherweight Java](#); [Spec](#); [JML](#)

Full Text: [DOI](#)

References:

- [1] Abraham, E., Grüner, A., Steffen, M.: Abstract interface behavior of object-oriented languages with monitors. In: Gorrieri, R., Wehrheim, H. (eds.) FMOODS 2006. LNCS, vol. 4037, pp. 218–232. Springer, Heidelberg (2006) · [Zbl 1162.68475](#) · [doi:10.1007/11768869_7](#)
- [2] Back, R.-J., Mikhajlova, A., von Wright, J.: Class refinement as semantics of correct object substitutability. Formal aspects of computing 12(1), 18–40 (2000) · [Zbl 0963.68103](#) · [doi:10.1007/s001650070034](#)
- [3] Banerjee, A., Naumann, D.A.: Ownership confinement ensures representation independence for object-oriented programs. Journal of the ACM 52(6), 894–960 (2005) · [Zbl 1316.68033](#) · [doi:10.1145/1101821.1101824](#)
- [4] Barnett, M., Naumann, D.A.: Friends need a bit more: Maintaining invariants over shared state. In: Kozen, D. (ed.) MPC 2004. LNCS, vol. 3125, pp. 54–84. Springer, Heidelberg (2004) · [Zbl 1106.68338](#) · [doi:10.1007/978-3-540-27764-4_5](#)
- [5] Barnett, M., DeLine, R., Fähndrich, M., Leino, K.R.M., Schulte, W.: Verification of object-oriented programs with invariants. Journal of Object Technology 3(6) (2004) · [Zbl 05431131](#) · [doi:10.5381/jot.2004.3.6.a2](#)
- [6] Barnett, M., Leino, K.R.M., Schulte, W.: The Spec# programming system: An overview. In: Barthe, G., Burdy, L., Huisman, M., Lanet, J.-L., Muntean, T. (eds.) CASSIS 2004. LNCS, vol. 3362, Springer, Heidelberg (2005)
- [7] Boyapati, C., Lee, R., Rinard, M.: Ownership types for safe programming: Preventing data races and deadlocks. In: Proc. OOPSLA 2002, pp. 211–230. ACM Press, New York (2002)
- [8] Broy, M.: A core theory of interfaces and architecture and its impact on object orientation. In: Reussner, R., Stafford, J.A., Szyperski, C.A. (eds.) Architecting Systems with Trustworthy Components. LNCS, vol. 3938, pp. 26–47. Springer, Heidelberg (2006) · [doi:10.1007/11786160_2](#)
- [9] Büchi, M.: Safe Language Mechanisms for Modularization and Concurrency. PhD thesis, Turku Centre for Computer Science (May 2000)
- [10] Chakrabarti, A., de Alfaro, L., Henzinger, T.A., Jurdzinski, M., Mang, F.Y.C.: Interface compatibility checking for software modules. In: Brinksma, E., Larsen, K.G. (eds.) CAV 2002. LNCS, vol. 2404, pp. 428–441. Springer, Heidelberg (2002) · [Zbl 1010.68505](#) · [doi:10.1007/3-540-45657-0_35](#)
- [11] Clarke, D.: Object Ownership and Containment. PhD thesis, University of New South Wales (July 2001) · [Zbl 0982.68763](#)

- [12] de Alfaro, L., Henzinger, T.A.: Interface automata. In: ESEC/FSE-9, pp. 109–120. ACM Press, New York (2001) · doi:10.1145/503209.503226
- [13] Flatt, M., Krishnamurthi, S., Felleisen, M.: A programmer’s reduction semantics for classes and mixins. *Formal Syntax and Semantics of Java* 1523, 241–269 (1999) · doi:10.1007/3-540-48737-9_7
- [14] Igarashi, A., Pierce, B.C., Wadler, P.: Featherweight Java: A minimal core calculus for Java and GJ. *TOPLAS* 23(3), 396–450 (2001) · Zbl 05459286 · doi:10.1145/503502.503505
- [15] Leavens, G.T., Naumann, D.A.: Behavioral subtyping, specification inheritance, and modular reasoning. Technical Report TR06-20a, Computer Science, Iowa State University (2006)
- [16] Leavens, G.T., Baker, A.L., Ruby, C.: Preliminary design of JML a behavioral interface specification language for Java. *SIGSOFT Softw. Eng. Notes* 31(3), 1–38 (2006) · doi:10.1145/1127878.1127884
- [17] Leino, K.R.M., Müller, P.: Object invariants in dynamic contexts. In: Odersky, M. (ed.) ECOOP 2004. LNCS, vol. 3086, pp. 491–516. Springer, Heidelberg (2004) · doi:10.1007/978-3-540-24851-4_22
- [18] Liskov, B., Wing, J.: A behavioral notion of subtyping. *ACM Transactions on Programming Languages and Systems* 16(6), 1811–1841 (1994) · doi:10.1145/197320.197383
- [19] Müller, P.: Modular Specification and Verification of Object-Oriented Programs. LNCS, vol. 2262. Springer, Heidelberg (2002) · Zbl 0998.68034 · doi:10.1007/3-540-45651-1
- [20] Nierstrasz, O.: Regular types for active objects. In: Proc. OOPSLA ’93, October 1993, pp. 1–15. ACM Press, New York (1993) · doi:10.1145/165854.167976
- [21] Poetsch-Heffter, A., Geilmann, K., Schäfer, J.: Inferring ownership types for encapsulated object-oriented program components. In: Program Analysis and Compilation, Theory and Practice: Essays Dedicated to Reinhard Wilhelm. LNCS, vol. 4444, Springer, Heidelberg (2007) · Zbl 1149.68342
- [22] Reynolds, J.C.: Separation logic: A logic for shared mutable data structures. In: Proceedings of LICS’02, pp. 55–74 (2002) · doi:10.1109/LICS.2002.1029817
- [23] Rinetzký, N., Poetsch-Heffter, A., Ramalingam, G., Sagiv, M., Yahav, E.: Modular shape analysis for dynamically encapsulated programs. In: ESOP’07. European Symposium on Programming, March 2007, Springer, Heidelberg (2007)
- [24] Schäfer, J., Poetsch-Heffter, A.: A parameterized type system for simple loose ownership domains. *Journal of Object Technology* (to appear, 2007) · Zbl 05737914 · doi:10.5381/jot.2007.6.5.a3

This reference list is based on information provided by the publisher or from digital mathematics libraries. Its items are heuristically matched to zbMATH identifiers and may contain data conversion errors. It attempts to reflect the references listed in the original paper as accurately as possible without claiming the completeness or perfect precision of the matching.