

Lei, Bin; Li, Xuandong; Liu, Zhiming; Morisset, Charles; Stolz, Volker
Robustness testing for software components. (English) [Zbl 1209.68140](#)
Sci. Comput. Program. 75, No. 10, 879-897 (2010).

Summary: Component-based development allows one to build software from existing components and promises to improve software reuse and reduce costs. For critical applications, the user of a component must ensure that it fits the requirements of the application. To achieve this, testing is a well-suited means when the source code of the components is not available. Robustness testing is a testing methodology to detect the vulnerabilities of a component under unexpected inputs or in a stressful environment. As components may fail differently in different states, we use a state machine based approach to robustness testing. First, a set of paths is generated to cover transitions of the state machine, and it is used by the test cases to bring the component into a specific control state. Second, method calls with invalid inputs are fed to the component in different states to test the robustness. By traversing the paths, the test cases cover more states and transitions compared to stateless API testing. We apply our approach to several components, including open source software, and compare our results with existing approaches.

MSC:

[68N99](#) Theory of software

Keywords:

[robustness testing](#); [component](#); [contract](#); [state machine](#)

Software:

[JCrasher](#); [rCOS](#)

Full Text: [DOI](#)

References:

- [1] Ali, S.; Briand, L. C.; Rehman, M. J.; Asghar, H.; Iqbal, M. Z. Z.; Nadeem, A.: A state-based approach to integration testing based on UML models, *Inf. softw. Technol.* 49, No. 11-12, 1087-1106 (2007)
- [2] Beck, K.; Gamma, E.: Test-infected: programmers love writing tests, *More Java gems (SIGS reference library)*, 357-376 (2000)
- [3] Binder, R. V.: *Testing object-oriented systems: models, patterns, and tools*, (1999)
- [4] Bouquet, F.; Grandpierre, C.; Legeard, B.; Peureux, F.; Vacelet, N.; Utting, M.: A subset of precise UML for model-based testing, , 95-104 (2007)
- [5] CCRA. Common Criteria for Information Technology Security Evaluation, <http://www.commoncriteriaportal.org/>.
- [6] Chen, X.; He, J.; Liu, Z.; Zhan, N.: A model of component-based programming, *Lecture notes in computer science* 4767, 191-206 (2007) · [Zbl 1141.68363](#) · [doi:10.1007/978-3-540-75698-9_13](#)
- [7] Z. Chen, A.H. Hannousse, D.V. Hung, I. Knoll, X. Li, Y. Liu, Z. Liu, Q. Nan, J.C. Okika, A. P. Ravn, V. Stolz, L. Yang, N. Zhan, *Modelling with relational calculus of object and component systems-rCOS*, in: Rausch et al. [29], chapter 3.
- [8] Csallner, C.; Smaragdakis, Y.: Jcrasher: an automatic robustness tester for Java, *Softw. pract. Exper.* 34, No. 11, 1025-1050 (2004)
- [9] R. Davies, *Practical refinement-type checking*. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2005.
- [10] Dowson, M.: The ariane 5 software failure, *SIGSOFT softw. Eng. notes* 84, No. 22 (1997)
- [11] D. Drusinsky, *Modeling and verification using UML statecharts: a working guide to reactive system design, Runtime Monitoring and Execution-based Model Checking*. Newnes, 2006.
- [12] Eclipse. Model Development Tools (MDT)/UML2 Tools, 2008. <http://www.eclipse.org/modeling/mdt/?project=uml2>.
- [13] Gallagher, L.; Offutt, J.: Test sequence generation for integration testing of component software, *Comput. J.* (2007)
- [14] Harrold, M.; Liang, D.; Sinha, S.: *An approach to analyzing and testing component-based systems*, (1999)
- [15] He, J.; Liu, Z.; Li, X.: Rcos: a refinement calculus of object systems, *Theoret. comput. Sci.* 365, No. 1-2, 109-142 (2006) · [Zbl 1118.68049](#)
- [16] Hoare, C. A. R.; He, J.: Unifying theories of programming, (1998) · [Zbl 1005.68036](#)

- [17] IEEE. Standard glossary of software engineering terminology, IEEE Std 610.12-1990, 10 Dec. 1990.
- [18] Kindrick, J. D.; Sauter, J. A.; Matthews, R. S.: Improving conformance and interoperability testing, *Standardview* 4, No. 1, 61-68 (1996)
- [19] Kropp, N. P.; Koopman, P. J.; Siewiorek, D. P.: Automated robustness testing of off-the-shelf software components, , 230 (1998)
- [20] Ledru, Y.; Du Bousquet, L.; Maury, O.; Bontron, P.: Filtering TOBIAS combinatorial test suites, *Lecture notes in computer science* 2984, 281-294 (2004)
- [21] Lei, B.; Liu, Z.; Morisset, C.; Li, X.: State based robustness testing for components, *Electr. notes in theoret. Comput. sci.* 260, 173-188 (2010)
- [22] Z. Liu, E. Kang, N. Zhan, Composition and refinement of components, in: *Post event Proceedings of UTP08*, in: *Lecture Notes in Computer Science*, Springer, 2009 (in press).
- [23] Liu, Z.; Morisset, C.; Stolz, V.: Rcos: theory and tools for component-based model driven development, *Lecture notes in computer science* 5961 (2009)
- [24] E. Martin, Automated testing and response analysis of web services, *Web Services, 2007, ICWS 2007, IEEE International Conference on*, pp. 647-654, July 2007.
- [25] Meyer, B.: Applying 'design by contract', *Computer* 25, No. 10, 40-51 (1992)
- [26] Miller, B. P.; Fredriksen, L.; So, B.: An empirical study of the reliability of UNIX utilities, *Commun. ACM* 33, No. 12, 32-44 (1990)
- [27] Necula, G. C.: Proof-carrying code, , 106-119 (1997)
- [28] Pacheco, C.; Lahiri, S. K.; Ernst, M. D.; Ball, T.: Feedback-directed random test generation, , 75-84 (2007)
- [29] , *Lecture notes in computer science* 5153 (2008)
- [30] SUN, The Reflection API, 2008. <http://java.sun.com/docs/books/tutorial/reflect/TOC.html>.
- [31] Szyperski, C.: *Component software: beyond object-oriented programming*, (1997)
- [32] Tretmans, J.: Test generation with inputs, outputs and repetitive quiescence, *Softw. concepts tools* 17, No. 3, 103-120 (1996) · [Zbl 0858.68061](#)
- [33] J. Warmer, A. Kleppe, Octopus open source project, 2006. <http://www.klasse.nl/>.
- [34] Whaley, J.; Martin, M. C.; Lam, M. S.: Automatic extraction of object-oriented component interfaces, *SIGSOFT softw. Eng. notes* 27, No. 4, 218-228 (2002)
- [35] Y. Wu, D. Pan, M.-H. Chen, Techniques for testing component-based software, in: *Engineering of Complex Computer Systems*, *IEEE International Conference on*, pp. 222-232, 2001.
- [36] W. Zheng, G. Bundell, Model-based software component testing: a UML-based approach. *Computer and Information Science*, 2007, *ICIS 2007, 6th IEEE/ACIS International Conference on*, pp. 891-899, 11-13 July 2007.

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.