{tag}

{/tag}
International Journal of Computer Applications

Volume 33 - Number 8

Year of Publication: 2011

Authors:

Somayeh Madadpour

Seyed-Hassan Mirian-Hosseinabadi

Vahdat Abdelzad

10.5120/4038-5783

{bibtex}pxc3875783.bib{/bibtex}

Abstract

Aspect-Oriented Programming is a software engineering paradigm that offers new constructs, such as join points, pointcuts, advices, and aspects in order to improve separation of crosscutting concerns. The new constructs bring new types of programming faults with respect to crosscutting concerns, such as incorrect pointcuts, advice, or aspect precedence. In fact,

existing object-oriented testing techniques are not adequate for testing aspect-oriented programs. As a result, new testing techniques must be developed. In this paper, an approach based upon UML activity diagram for testing aspect-oriented programs is presented. The proposed approach focuses on integration of one or several crosscutting concerns to a primary concern and tests whether or not an aspect-oriented program conforms to its expected crosscutting behaviors. The proposed approach generates test sequences based on interaction between aspects and primary models, and verifies the execution of the selected sequences. It also, follows an iterative process which causes to discover faults easily and quickly. The approach is based on several test criteria that we defined. To illustrate the approach, we use a case study which its results show that the approach is capable of revealing several aspect-specific faults.

Reference

- Filman, R. E., Elrad, T., Clarke, S., and Aksit, M., “Aspect-Oriented Software Development”, Addison-Wesley Professional, Boston, 2004.
- Hursch, W. L., and Lopes, C. V., “Separation of Concerns”, Technical Report No. NUCCS-95-03, College of Computer Science, Northeastern University, Boston, 1995.
- Colyer, A., Clement, A., Harly, G., and Webster, M., “Eclipse AspectJ: Aspect-Oriented Programming with AspectJ and the Eclipse AspectJ Development Tools”, Addison- Wesley Professional, 2004.
- El-Far, I. K., and Whittaker, J.A., ‘Model-based software testing”, In Encyclopedia on Software Engineering (edited by Marciniak), Wiley, 2001.
- Pretschner, A., Prenninger, W., Wagner, S., Kühnel, C., Baumgartner, M., Sostawa, B., Zölch, R., and Stauner, T., “One evaluation of model-based testing and its automation”, In Proc. of the 27th International Conf.on Software Engineering (ICSE'05), 2005.
- Pretschner, A., Slotosch, O., Aiglstorfer, E., and Kriebel, S., “Model-based testing for real - The inhouse card case study”, J. Software Tools for Technology Transfer 5(2-3):140-157, 2004.
- 7 Dalal, S. R., Jain, A., Karunanithi, N., Leaton, J. M., Lott, C. M., Patton, G. C., and Horowitz, B. M., ‘Model-based testing in practice”, In Proc. of the 21st International Conf. on Software Engineering (ICSE'99), 1999.
- Blackburn, M., Busser, R., Nauman, A., Knickerbocker, R., and Kasuda, R., “Mars Polar Lander fault identification using model-based testing”, In Proc. of the Eighth International Conference on Engineering of Complex Computer Systems, 2002.
- OMG, UML Superstructure v2.1, http://www.omg.org/ documents/formal/uml.htm.
- Alexander, R. T., Bieman, J. M., and Andrews, A.A., “Towards the systematic testing of aspect-oriented programs”, Technical Report, Colorado State University, http://www.cs. colostate.edu/~rta/publications/CS-04-105.pdf, 2004.
- Zhao, J. “Data-flow-based unit testing of aspect-oriented programs”, Proc. of COMPSAC'03, pp.188-197, Dallas, Texas, USA, 2003.
- Zhao, J. and Rinard, M., “System dependence graph construction for aspect-oriented programs”, MIT-LCSTR-891, Laboratory for Computer Science, MIT, 2003.
- Zhou, Y., Richardson, D., and Ziv, H., “Towards a practical approach to test aspect-oriented software”, In Proc. of the 2004 Workshop on Testing Component-based Systems (TECOS 2004), 2004.

- Xie, T. and Zhao, J., “A framework and tool supports for generating test inputs of AspectJ programs”, In Proc. of the 5th International Conference on Aspect-Oriented Software Development (AOSD’ 06), pp. 190-201, 2006.
- Xie, T., Zhao, J., Marinov, D., and Notkin, D., “Automated test generation for AspectJ programs”, AOSD 2005 Workshop on Testing Aspect-Oriented Programs, Chicago, 2005.
- Xu, D., Xu, W., and Nygard, K., “A state-based approach to testing aspect-oriented programs”, In Proceedings of the 17th International Conference on Software Engineering and Knowledge Engineering, pp. 366-371, 2005.
- Xu, D., and Xu, W., “State-based incremental testing of aspect-oriented programs”, In Proceedings of the 5th International Conference on Aspect-Oriented Software Development, pp. 180-189, 2006.
- Xu, W., and Xu, D., “State-based testing of integration aspects”, In Proceedings of the 2nd Workshop on Testing Aspect-Oriented Programs, pp. 7-14, 2006.
- Binder, R. V., “Testing Object-Oriented Systems: Models, Patterns, and Tools”, Addison-Wesley Professional, Boston, 2000.
- Xu, W., Xu, D., and Wong, W. E., “Testing Aspect-Oriented Programs with UML Design Models”, International Journal of Software Engineering and Knowledge Engineering, Vol. 18, No. 3, pp. 413-437, May 2008.
- Xu, W. and Xu, D., “A model-based approach to test generation for aspect-oriented programs”, AOSD 2005 Workshop on Testing Aspect-Oriented Programs, Chicago, 2005.
- Liu, C. H., and Chang, C. W., “A State-Based Testing Approach for Aspect-oriented Programming”, In Journal of Information Science and Engineering , pp. 11-31, 2008.
- Badri, B., Badri, L., Fortin, M. B., “Automated State-Based Unit Testing for Aspect-Oriented Programs: A Supporting Framework”, In Journal of Object Technology, vol. 8, no. 3, pp. 121-126, 2009.
- Cui, Z., Wang, L., and Li, X., “Modeling and integrating aspects with uml activity diagrams”, Proceedings of the 2009 ACM symposium on Applied Computing, 2009.
- Mortensen, M., and Alexander, R., “Adequate Testing of Aspect-Oriented Programs”, Technical report CS 04-110, Colorado State University, Fort Collins, Colorado, USA, December 2004.
- Offut, J., Xiong, Y., and Liu, S., “Criteria for Generating Specification-based Tests”, In Engineering of Complex Computer Systems, ICECCS '99, 1999.
- Offut, J., and Voas, J., “Subsumption of Condition Coverage Techniques by Mutation Testing”, ISSE-TR-96-01, January 1996.
- Beizer, B., “Software Testing Techniques”, International Thomson Computer Press, 1990.

- AspectJ Web Site, http://eclipse.org/aspectj/.

Index Terms

Computer Science Software Engineering

Key words

Aspect-Oriented Programming
Model-Based Testing
Aspect-Oriented Modeling
UML Activity Diagrams
Test Sequences