

9

References

AspectJ Team. **The AspectJ Programming Guide**. 2007. At: <http://eclipse.org/aspectj/>.

ALDRICH, J. Open Modules: Modular Reasoning about Advice. In: **Proceedings of the European Conference on Object-Oriented Programming**, Glasgow, Scotland, Springer 2005, p.144-168.

ALEXANDER, R.T.; BIEMAN, J.M.; ANDREWS, A.A. **Towards the Systematic Testing of Aspect-Oriented Programs**. Dept. of Computer Science, Colorado State University Fort Collins/Colorado - USA, 2004, 11p. Technical Report.

ALLEN, E. **Bug patterns in Java**. Berkeley: Apress, 2002, 255p., ISBN: 1-59059-061-9.

ANBALAGAN, P.; XIE, T. Efficient Mutant Generation for Mutation Testing of Pointcuts in Aspect-Oriented Programs. In: **Proceedings of the 2nd Workshop on Mutation Analysis (MUTATION 2006)**, Raleigh, NC, 2006, p.51-56.

APEL, S.; LEICH, T.; SAAKE, G. Aspectual Mixin Layers: Aspects and Features in Concert. In: **Proceedings of the Proceeding of the 28th international conference on Software engineering**, Shanghai, China, ACM Press, 2006, p.122-131.

BAEKKEN, J.S. **Fault Model for Pointcuts and Advice in AspectJ Programs**. Washington, 2006, 115p, Master Thesis, School of Electrical Engineering and Computer Science, Washington State University.

BAEKKEN, J.S.; ALEXANDER, R.T. A Candidate Fault Model for AspectJ Pointcuts. In: **Proceedings of the 17th International Symposium on Software Reliability Engineering (ISSRE 2006)**, 2006, p.169-178.

BALDWIN, C.Y.; CLARK, K.B. **Design Rules: The Power of Modularity**. Cambridge, MA: MIT Press, 2000, 483p.

BINDER, R.V. **Testing Object-Oriented Systems, Models, Paterns and Tools**. Addison Wesley, 1999, 1248p.

BOBROW, D.; DIMICHIEL, L.; GABRIEL, R.; KEENE, S.; KICZALES, G.; MOON, D.A. Common Lisp Object System Specification. **ACM SIGPLAN Notices**, 23(SI), 1988, p.1-142.

BODKIN, R. Performance monitoring with AspectJ - A look inside the Glassbox Inspector with AspectJ and JMX, AOP@Work, Sep 2005. Available at: <http://www.ibm.com/developerworks/java/library/j-aopwork10/index.html>. Accessed: 12/2007.

BODKIN, R. AOP@Work: Next steps with aspects, March 2006. Available at: <http://www.ibm.com/developerworks/java/library/j-aopwork16/>. Accessed: 07/2007.

BRUNTINK, M.; DEURSEN, A.V.; TOURWÉ, T. Discovering faults in idiom-based exception handling. In: **Proceedings of the International Conference on Software Engineering**, 2006, p.242-251.

BURKE, B.; BROCK, A. **Aspect-Oriented Programming and JBoss**. O'Reilly Media, Available at: http://www.onjava.com/pub/a/onjava/2003/05/28/aop_jboss.html Accessed: 07/2007.

BUSCHMANN, F.; MEUNIER, R.; ROHNERT, H.; SOMMERLAD, P.; STAL, M. **Pattern-Oriented Software Architecture, Volume 1: A System of Patterns**. Wiley, 1996, 476p.

CABRAL, B.; MARQUES, P. Exception Handling: A Field Study in Java and .NET. In: **Proceedings of the European Conference on Object-Oriented Programming**, Berlin, Germany, Springer Berlin / Heidelberg, 2007, p.151-175.

CECCATO, M.; TONELLA, P.; RICCA, F. Is AOP Code Easier or Harder to Test than OOP Code? . In: **Proceedings of the Testing Aspect-Oriented Programs (WTAOP 2005)**, Illinois, USA, 2005.

CHANG, B.-M.; JO, J.-W.; YI, K.; CHOE, K.-M. Interprocedural Exception Analysis for Java. In: **Proceedings of the ACM Symposium on Applied Computing (SAC 2001)**, LasVegas, USA, 2001, p.620-625

CLIFTON, C.; LEAVENS, G.T. Observers and Assistants: A Proposal for Modular Aspect-Oriented Reasoning. In: **Proceedings of the Workshop on Foundations of Aspect Languages (FOAL)**, 2002.

COELHO, R.; ALVES, V.; KULESZA, U.; NETO, A.C.; GARCIA, A.; STAA, A.V.; LUCENA, C.; BORBA, P. On Testing Crosscutting Features using Extension Join Points. In: **Proceedings of the International 3rd Workshop on Software Product Line Testing, 3rd (SPLiT 2006), in conjunction with 10th International Software Product Line Conference (SPLC 2006)**, Baltimore, Maryland, 2006a, p.23-30.

COELHO, R.; DANTAS, A.; KULESZA, U.; STAA, A.; CIRNE, W.; LUCENA, C. The Application Monitor Aspect Pattern. In: **Proceedings of the PLoP 2006, in conjunction with OOPSLA 2006**, 2006b.

COELHO, R.; RASHID, A.; GARCIA, A.; FERRARI, F.; CACHO, N.; KULESZA, U.; STAA, A.V.; LUCENA, C. Assessing the Impact of Aspects on Exception Flows: An Exploratory Study. In: **Proceedings of the 22nd European Conference on Object-Oriented Programming (ECOOP'08)**, Cyprus, LNCS, Springer-Verlag, 2008, pp. 207-234.

COLYER, A. **Eclipse AspectJ: Aspect-Oriented Programming with AspectJ and the Eclipse AspectJ Development Tools**. Addison-Wesley, 2004, 504p.

DEURSEN, A.V.; MARIN, M.; MOONEN, L. **A Systematic Aspect-Oriented Refactoring and Testing Strategy, and its Application to JHotDraw**, Stichting Centrum voor Wiskunde en Informatica, Netherlands, 2005, 25p. Technical Report.

DOOREN, M.V.; STEEGMANS, E. Combining the Robustness of Checked Exceptions with the Flexibility of Unchecked Exceptions Using Anchored Exception Declarations. In: **Proceedings of the 20th annual ACM SIGPLAN conference on Object oriented programming systems languages and applications (OOPSLA'05)**, ACM Press, 2005, p.455–471.

EINARSSON, A.; NIELSEN, J.D. A Survivor's Guide to Java Program Analysis with Soot. Available at: <http://www.brics.dk/SootGuide/sootsurvivorsguide.pdf>. Accessed: 03/2007.

FAHNDRICH, M.; FOSTER, J.; AIKEN, A.; CU, J. **Tracking down exceptions in standard ml programs**. Computer Science Division, University of California, Berkeley, February, 1998, 7p. Technical Report.

FAITELSON, D.; KATZ, S. **The preliminary implementation of CAPE**. AOSD Europe Project, Deliverable available at: <http://www.aosd-europe.net/deliverables/d71.pdf>, 2008, 23p. Technical Report.

FIGUEIREDO, E.; CACHO, N.; SANT'ANNA, C.; MONTEIRO, M.; KULESZA, U.; GARCIA, A.; SOARES, S.; FERRARI, F.; KHAN, S.; FILHO, F.; DANTAS, F. Evolving Software Product Lines with Aspects: An Empirical Study on Design Stability. In: **Proceedings of the International Conference on Software Engineering**, Leipzig, Germany, 2008, p.261-270.

FILHO, F.C.; CACHO, N.; MARANHÃO, R.; FIGUEIREDO, E.; GARCIA, A.; RUBIRA, C.M.F. Exceptions and Aspects: The Devil is in the Details. In: **Proceedings of the 14th ACM. SIGSOFT Symposium on Foundations of Software Engineering**, ACM, 2006, p.112-121.

FILHO, F.C.; GARCIA, A.; RUBIRA, C. Extracting Error Handling to Aspects: A Cookbook. In: **Proceedings of the IEEE International Conference on Software Maintenance (ICSM)**, Paris, France, 2007, p.134-143.

FILMAN, R.; FRIEDMAN, D. Aspect-Oriented Programming is Quantification and Obliviousness. In: **Aspect-Oriented Software Development**: Addison-Wesley, 2005. p.21-35.

FLEURY, M.; REVERBEL, F. The JBoss Extensible Server. In: **Proceedings of the International Middleware Conference 2003**, p.344–373.

FOSTER, J.S.; TERAUCHI, T.; AIKEN, A. Flow-sensitive type qualifiers. **ACM SIGPLAN Notices**, 37(5), 2002, p.1-12.

FU, C.; MILANOVA, A.; RYDER, B.; WONNACOTT, D. Robustness testing of Java Server Applications. **IEEE Transactions on Software Engineering**, 31(4), p.292- 311, 2005.

FU, C.; RYDER, B.; MILANOVA, A.; AND WONNACOTT, D. Testing of Java Web Services for Robustness. In: **Proceedings of the**, 2004, p.23–34.

FU, C.; RYDER, B.G. Testing and Understanding Error Recovery Code in Java Applications. In: **Proceedings of the ECOOP2005 - Workshop on Exception Handling in Object Oriented Systems (EHOOS'2005)**, 2005.

FU, C.; RYDER, B.G. Exception-Chain Analysis: Revealing Exception Handling Architecture in Java Server Applications. In: **Proceedings of the International Conference on Software Engineering (ICSE)**, ACM Press, 2007, p.230-239.

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. **Design Patterns: Elements of Reusable Object-Oriented Software**. Addison-Wesley Longman Publishing Co., 1995, 416p.

GARCIA, A.; SANT'ANNA, C.; FIGUEIREDO, E.; KULESZA, U.; LUCENA, C.; STAA, A.V. Modularizing Design Patterns with Aspects: A Quantitative Study. In: **Proceedings of the Proceedings of the 4th international conference on Aspect-oriented software development**, Chicago, Illinois, ACM Press, 2005, p.3-14.

GARCIA, A.F.; RUBIRA, C.M.F. A Comparative Study of Exception Handling Mechanisms for Building Dependable Object-Oriented Software. **Journal of Systems and Software**, 59 (6), 2001, p.197-222.

GOODENOUGH, J.B. Exception Handling: Issues and a Proposed Notation. **Communications of the ACM**, 18(12), p.683-696, 1975.

GOSLING, J.; JOY, B.; STEELE, G.L. **The Java Language Specification**, 3rd Ed, Boston: Addison-Wesley Publishing Co, 1996, 688p.

GREENWOOD, P.; BARTOLOMEI, T.; FIGUEIREDO, E.; DOSEA, M.; GARCIA, A.; CACHO, N.; SANT'ANNA, C.; SOARES, S.; BORBA, P.; KULESZA, U.; RASHID, A. On the Impact of Aspectual Decompositions on Design Stability: An Empirical Study. In: **Proceedings of the European Conference of Object-Oriented Programming (ECOOP'07)**, Springer-Verlag, 2007, p.176-200.

GRISWOLD, W.G.; SULLIVAN, K.J.; SONG, Y.; SHONLE, M.; TEWARI, N.; CAI, Y.; RAJAN, H. Modular Software Design with Crosscutting Interfaces. **IEEE Software**, 23(1), p.51-60, 2006.

GROVE, D.; CHAMBERS, C. A framework for call graph construction algorithms. . **ACM Transactions on Programming Languages and Systems (TOPLAS)**, 23(6), p.685-746, 2001.

HANENBERG, S.; SCHMIDMEIER, A.; UNLAND, R. AspectJ Idioms for Aspect-Oriented Software Construction. In: **Proceedings of the Proceedings of 8th European Conference on Pattern Languages of Programs (EuroPLoP'2003)**, 2003.

HANNEMANN, J.; KICZALES, G. Design Pattern Implementation in Java and AspectJ. In: **Proceedings of the Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Seattle, Washington, USA, ACM Press, 2002, p.161-173.

HILSDALE, E.; HUGUNIN, J. Advice weaving in AspectJ. In: **Proceedings of the In 3rd International. Conference on Aspect-oriented Software Development (AOSD 2004)**, Lancaster, UK, 2004, p.26-35.

ISHIO, T.; KUSUMOTO, S.; KATSUROINOUE. Debugging Support for Aspect-Oriented Program Based on Program Slicing and Call Graph. In: **Proceedings of the Proceedings of the 20th IEEEInternational Conference on SoftwareMaintenance (ICSM)**, Washington, DC,USA, IEEE Computer Society., 2004, p.178-187.

JO, J.-W.; CHANG, B.-M.; YI, K.; CHOE, K.-M. An Uncaught Exception Analysis for Java. **Journal of Systems and Software**, 72 (1), 2004, p.59-69.

JOHNSON, R. **Spring Reference, Chapters 6 and 7: Spring AOP: Aspect Oriented Programming with Spring.** Available at: <http://static.springframework.org/spring/docs/2.5.x/reference/aop.html>. Accessed: 10/2007.

JOHNSON, R.; HOELLER, J.; ARENDSSEN, A.; RISBERG, T.; SAMPALEANU, C. **Professional Java Development with the Spring Framework.** Wrox, 2005, 672p.

KATZ, S. Aspect Categories and Classes of Temporal Properties. **Transactions on Aspect-Oriented Software Development**, 3380(12), 2006, p.106-134.

KICZALES, G. Aspect-oriented programming. **ACM Computing**, 28(154), 1996.

KICZALES, G.; HILSDALE, E.; HUGUNIN, J.; KERSTEN, M.; PALM, J.; GRISWOLD, W. Getting Started with AspectJ. **Communications of ACM**, 44(10), p.59-65, 2001a.

KICZALES, G.; HILSDALE, E.; HUGUNIN, J.; KERSTEN, M.; PALM, J.; GRISWOLD, W. An Overview of AspectJ. In: **Proceedings of the 15th European Conference on Object-Oriented Programming (ECOOP'01)**, London, UK, Springer-Verlag, 2001b, p.327-353.

KICZALES, G.; LAMPING, J.; MENDHEKAR, A.; MAEDA, C.; LOPES, C.; LOINGTIER, J.-M.; IRWIN, J. Aspect-Oriented Programming. In: **Proceedings of the European Conference of Object-Oriented Programming (ECOOP'97)**, Springer-Verlag, 1997, p.220-242.

KICZALES, G.; MEZINI, M. Aspect-oriented Programming and Modular Reasoning. In: **Proceedings of the Proceedings of the 27th International Conference on Software Engineering**, St. Louis, MO, USA, 2005, p.49-58.

KIENZLE, J.; GUERRAOU, R. AOP:Does it make sense? the case of concurrency and failures. In: **Proceedings of the 16th European Conference Object Oriented Programming**, France, 2002, p.37-54.

KRISHNAMURTHI, S.; FISLER, K.; GREENBERG, M. Verifying Aspect Advice Modularly. In: **Proceedings of the ACM SIGSOFT International Symposium on the Foundations of Software Engineering**, California, Irvine, USA, 2004, p.137-146.

KULESZA, U.; SANT'ANNA, C.; GARCIA, A.; COELHO, R.; STAA, A.V.; LUCENA, C.J.P.D. Quantifying the Effects of Aspect-Oriented Programming: A Maintenance Study. In: **Proceedings of the Proceedings of 22nd IEEE International Conference on Software Maintenance (ICSM 2006)**, Philadelphia, Pennsylvania, USA, IEEE Computer Society 2006, p.223-233.

LEMO, O.; FERRARI, F.; LOPES, C.; MASIERO, P. Testing aspect-oriented programming Pointcut Descriptors. In: **Proceedings of the Workshop on Testing Aspect-Oriented Programs**, Portland, USA, 2006, p.33-38.

LEMO, O.A.L.; VINCENZI, A.M.R.; MALDONADO, J.C.; MASIERO, P.C. Control and data flow structural testing criteria for aspect-oriented programs. **Journal of Systems and Software** 80(6), 2007, p.862-882.

LHOTAK, O. **Spark: A flexible points-to analysis framework for Java.** 2002. Master Thesis. McGill University, Montreal, 2002.

LHOTAK, O.; HENDREN, L. Context-sensitive points-to analysis: is it worth it? . In: **Proceedings of the 15th International Conference on Compiler Construction**, Vienna, Springer - LNCS, 2006, p.47–64.

LIANG, D.; PENNINGS, M.; HARROLD, M.J. Extending and evaluating fow-insensitive and context-insensitive points-to analyses for Java. **ACM SIGPLAN Notes**, p.73-79, 2001.

LIANG, D.; PENNINGS, M.; HARROLD, M.J. Evaluating the impact of context-sensitivity on Andersen's algorithm for Java programs. In: **Proceedings of the 6th ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering (PASTE 2005)**, 2005, p.6-12.

LIPPERT, M.; LOPES, C.V. A Study on Exception Detection and Handling Using Aspect-Oriented Programming. In: **Proceedings of the International Conference on Software Engineering (ICSE'00)**, ACM Press, 2000, p.418-427.

LOPES, C.; NGO, T. Unit-Testing Aspectual Behavior. In: **Proceedings of the Workshop on Testing Aspect-Oriented Programs 2005 (WTAOP 2005)**, 2005.

LOPEZ-HERREJON, R.; BATORY, D.; LENGAUER, C. A disciplined approach to aspect composition. In: **Proceedings of the ACM/SIGPLAN Workshop Partial Evaluation and Semantics-Based Program Manipulation** Charleston, South Carolina, 2006, p.68-77.

LOURIDAS, P. Static Code Analysis. **IEEE Software** 23(4), p.58-61, 2006.

MACKINNON, T.; FREEMAN, S.; CRAIG, P. Endo-testing: Unit testing with mock objects. In: **Extreme Programming Examined**, Giancarlo Succi and Michele Marchesi: Addison-Wesley, 2001.

MARIN, M.; MOONEN, L.; DEURSEN, A.V. Integrated Crosscutting Concern Migration Strategy and its Application to JHotDraw. In: **Proceedings of the Source Code Analysis and Manipulation (SCAM 2007)**, Paris, France, 2007, p.101-110.

MASSICOTTE, E.P.; BADRI, L.; BADRI, M. Towards a Tool Supporting Integration Testing of Aspect-Oriented Programs. **Journal of Object Technology**, 6(1), 2007, p.67-89.

MASSONI, T.; ALVES, V.; SOARES, S.; BORBA, P. PDC: Persistent Data Collections pattern. In: **Proceedings of the In First Latin American Conference on Pattern Languages of Programming — SugarLoafPLoP**, University of Sao Paulo Magazine - ICMC, 2001, p.311–326.

MATSUOKA, S.; YONEZAWA, A. **Analysis of inheritance anomaly in object-oriented concurrent programming languages**. Cambridge, MA: The MIT Press, 1993.

MCCUNE, T. **Java.net: Exception Handling Antipatterns**. Available at: <http://today.java.net/pub/a/today/2006/04/06/exception-handling-antipatterns.html>. Accessed: 19/12/2007.

MCEACHEN, N.; ALEXANDER, R.T. Distributing classes with woven concerns: an exploration of potential fault scenarios. In: **Proceedings of the International Conference on Aspect Oriented Software Development (AOSD) 2005**, p.192-200.

MEZINI, M.; OSTERMANN, K. Conquering Aspects with Caesar. In: **Proceedings of the Proceedings of the 2nd International Conference on Aspect-oriented Software Development**, Boston, Massachusetts, ACM Press, 2003, p.90-99.

MILANOVA, A.; ROUNTEV, A.; RYDER, B. Practical Object-sensitive Points-to Analysis for Java. Available at: www.iei.pi.cnr.it/ISSTA/slides/07222002/sess2pr1.ppt Accessed: 09/2007.

MILLER, R.; TRIPATHI, A. Issues with Exception Handling in Object-Oriented Systems. . In: **Proceedings of the European Conference on Object Oriented Programming (ECOOP'97)**, Springer, 1997, p.85–103.

MINE, A. Field-sensitive value analysis of embedded C programs with union types and pointer arithmetics. **ACM SIGPLAN Notices**, 41(7), 2006, p.54-63.

MYERS, G.J. **The Art of Software Testing**. New York: John Wiley & Sons, 2004, 192p.

OGASAWARA, T.; KOMATSU, H.; NAKATANI, T. A study of exception handling and its dynamic optimization in Java. In: **Proceedings of the ACM Conference on Object-Oriented Programming Systems, Languages, and Applications**, 2001, p.83-95.

RASHID, A.; CHITCHYAN, R. Persistence as an Aspect. In: **Proceedings of the 2nd International Conference on Aspect-oriented Software Development**, Boston, Massachusetts, ACM Press, 2003, p.120-129.

RINARD, M. A Classification System and Analysis for Aspect-Oriented Programs. In: **Proceedings of the Proceedings of the Twelfth International Symposium on the Foundations of Software Engineering**, Newport Beach, California, 2004 p.147-158.

ROBILLARD, M.; MURPHY, G. Analyzing Exception Flow in Java Programs. In: **Proceedings of the 7th European Software Engineering Conference and 7th ACM SIGSOFT Symposium on the Foundations of Software Engineering**, Springer-Verlag, 1999, p.322-337.

ROBILLARD, M.; MURPHY, G. Designing Robust Java Programs with Exceptions. In: **Proceedings of the Eighth International Symposium on the Foundations in Software Engineering**, San Diego, California, USA, 2000, p.2-10.

ROBILLARD, M.; MURPHY, G. Static Analysis to Support the Evolution of Exception Structure in Object-Oriented Systems. **ACM Transactions on Software Engineering Methodologies**, 12(2), 2003, p.191-221.

ROUNTEV, A.; KAGAN, S.; GIBAS, M. Static and dynamic analysis of call chains in Java. In: **Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'04)**, ACM, 2004, p.1-11.

RYDER, B. Constructing the Call Graph of a Program. **IEEE Transactions on Software Engineering**, 5(3), p.216-226, 1979.

SACRAMENTO, P.; CABRAL, B.; MARQUES, P. Unchecked Exceptions: Can the Programmer be Trusted to Document Exceptions? In: **Proceedings of the**

International Conference on Innovative Views of .NET Technologies (IVNET'06), Florianópolis, Brazil, 2006.

SALCIANU, A. **Pointer analysis and its applications to Java programs**. 2001, 135p, Master Thesis. MIT Laboratory for Computer Science.

SANEN, F. Classifying And Documenting Aspect Interactions. In: **Proceedings of the 5th AOSD Workshop on Aspects, Components, and Patterns for Infrastructure Software (ACP4IS) at AOSD 2006**, Bonn, Germany, 2006.

SCHAEFER, C.F.; BUNDY, G.N. Static analysis of exception handling in Ada. **Software: Practice and Experience**, 23(10), p.1157–1174, 1993.

SINHA, S.; HARROLD, M.J. Analysis of programs with exception handling constructs. In: **Proceedings of the International Conference on Software Maintenance**, Los Alamitos, California, USA, IEEE Computer Society Press, 1998, p.348-357.

SINHA, S.; HARROLD, M.J. Criteria for testing exception-handling constructs in Java programs. In: **Proceedings of the International Conference on Software Maintenance**, IEEE Computer Society Press, Los Alamitos, CA, 1999, p.265-274.

SINHA, S.; HARROLD, M.J. Analysis and testing of programs with exception-handling constructs. **IEEE Transactions on Software Engineering**, 26(9), p.849-871, 2000.

SINHA, S.; ORSO, A.; HARROLD, M.J. Automated Support for Development, Maintenance, and Testing in the Presence of Implicit Control Flow. In: **Proceedings of the 26th IEEE and ACM SIGSOFT International Conference on Software Engineering (ICSE 2004)**, Edinburgh, Scotland, 2004, p.336-345.

SOARES, S. **An Aspect-Oriented Implementation Method, PhD Thesis**. 2004. PhD Thesis. Federal University of Pernambuco, Computer Science Department, 2004.

SOARES, S.; BORBA, P.; LAUREANO, E. Distribution and Persistence as Aspects. **Software Practice and Experience (SP&E)**, 36(7), p.711-759, 2006.

Soot: Java Optimization Framework. At: <http://www.sable.mcgill.ca/>

SULLIVAN, K.; GRISWOLD, W.G.; SONG, Y.; CAI, Y.; SHONLE, M.; TEWARI, N.; RAJAN, H. Information Hiding Interfaces for Aspect-Oriented Design. In: **Proceedings of the Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering**, Lisbon, Portugal, ACM Press, 2005, p.166-175.

THOMAS, D. The Deplorable State of Class Libraries. **Journal of Object Technology**, 1(1), 2002, p.21-27.

VALLÉE-RAI, R. **Soot: A Java bytecode optimization framework**. 2000. Master Thesis. McGill University, Montréal, Québec, Canada, 2000.

VALLÉE-RAI, R.; CO, P.; GAGNON, E.; HENDREN, L.; LAM, P. Soot - a java bytecode optimization framework. In: **Proceedings of the CASCON '99**, 1999.

VINCENZI, A.; WONG, W.E.; DELAMARO, M.; MALDONADO, J.C. JaBUTi: A Coverage Analysis Tool for Java Programs. In: **Proceedings of the 17th**

Brazilian Symposium on Software Engineering (SBES), Manaus, AM, Brazil, 2003.

WESTON, N.; TAIANI, F.; RASHID, A. Interaction Analysis for Fault Tolerance in Aspect-Oriented Programming. In: **Proceedings of the Workshop on Methods, Models and Tools for Fault Tolerance**, Oxford, UK, Technical report CS-TR-1032, Newcastle University, UK, 2007, p.95-102.

WOHLIN, C.; RUNESON, P.; HOST, M.; OHLSSON, M.C.; REGNELL, B.; WESSLEN, A. **Experimentation in Software Engineering - An Introduction**. 1st Ed. Kluwer: Springer, 2000, 228p. ISBN: 978-0792386827.

XIE, T.; ZHAO, J. A framework and tool supports for generating test inputs of AspectJ programs. In: **Proceedings of the In Proc. 5th International Conference on Aspect-Oriented Software Development**, 2006, p.190–201.

XIE, T.; ZHAO, J.; MARINOV, D.; NOTKIN, D. Detecting Redundant Unit Tests for AspectJ Programs. In: **Proceedings of the 17th International Symposium on Software Reliability Engineering (ISSRE 2006)**, Raleigh, North Carolina, USA, Publisher IEEE Computer Society 2006, p.179-190.

XU, W.; XU, D. State-Based Testing of Integration Aspects. In: **Proceedings of the Second Workshop on Testing of Aspect-Oriented Programs (WTAOP'06). In conjunction with ISSTA'06**, USA, 2006, p.7-14.

YI, K.; RYU, S. A cost-effective estimation of uncaught exceptions in Standard ML programs. 277(1-2), 2002, p.185-217.

ZHANG, S.; ZHAO, J. On Identifying Bug Patterns in Aspect-Oriented Programs. In: **Proceedings of the Computer Software and Applications Conference (COMPSAC 2007)**, 2007, p.431–438.

ZHANG, W.; RYDER, B.G. Automatic construction of accurate application call graph with library call abstraction for Java. **Journal of Software Maintenance**, 19(4), 2007, p.231-252.

ZHAO, J. Data-Flow-Based Unit Testing of Aspect-Oriented Programs. In: **Proceedings of the 27th Annual IEEE International Computer Software and Applications Conference (COMPSAC'2003)**, Dallas, Texas, USA, 2003, p.188 - 197.