Dirk Beyer
Curriculum Vitae

2023-11-28

Coordinates

Affiliation:	Ludwig-Maximilians-Universität München Chair Software and Computational Systems Oettingenstr. 67 D-80538 Munich Germany	Web: Phone: Office:	www.sosy-lab.org/∼dbeyer +49 (89) 2180-9150 062 (main floor)
Citizenship:	German	Year of birth:	1972
Town of birth:	Finsterwalde, Germany	Marital status:	Married, three children

Research Interests

My research focuses on models, algorithms, and tools for the construction and analysis of reliable software systems, in particular (tools in whose development I was involved are given in parentheses):

- Software model checking and static analysis (CPAchecker, Blast, CSIsat)
- Structural analysis and comprehension of large software systems (CrocoPat, CCVisu)
- Interfaces for component-based design (Chic, CoVeriTeam)
- Formal verification of real-time systems (Rabbit)

The conceptional basis of my work is in software engineering, programming languages, data structures and efficient algorithms, and mathematical logic. My goal is not only to develop new concepts, but also to provide efficient tool implementations derived from the research results.

Education

1998 – 2002	Academic degree Dr. rer. nat. “magna cum laude” Brandenburgische Technische Universität Cottbus, Germany
1994 – 1998	Academic degree Diplom-Informatiker “with distinction” Brandenburgische Technische Universität Cottbus, Germany University award 1998 for Master’s thesis

Academic Employment

since 2016	Professor (W3) Ludwig-Maximilians-Universität München, Germany
2009 – 2016	Professor (W3) Universität Passau, Germany
2010 – 2011	Associate Professor Simon Fraser University, B.C., Canada
2006 – 2010	Assistant Professor Simon Fraser University, B.C., Canada
2004 – 2006	Postdoctoral Researcher, Host: Prof. Thomas A. Henzinger EPFL, Lausanne, Switzerland
2003 – 2004	Postdoctoral Researcher, Host: Prof. Thomas A. Henzinger University of California, Berkeley, U.S.A.
1998 – 2003	Research and Teaching Assistant, Advisor: Prof. Claus Lewerentz Brandenburgische Technische Universität Cottbus, Germany

Industrial Employment

1998 – 1998	Software Engineer Siemens AG, Business Services Dresden, Dept. Major Projects

Software (Licensed as free software.)

BenchExec: Reliable benchmarking and resource measurement, 2015.
https://github.com/dbeyer/BenchExec
Contributor.

Blast: Model checking of software, 2002.
http://www.sosy-lab.org/~dbeyer/Blast/
Contributor, conceptual extensions, implementation, and maintenance.

CCVisu: Visual clustering and software-structure assessment, 2005.
http://www.sosy-lab.org/~dbeyer/CCVisu/
Principal designer and implementer.

CheckDep: Tracking software dependencies, 2010.
http://www.sosy-lab.org/~dbeyer/CheckDep/
Principal designer, architect, and maintenance.

Chic: Checking interface compatibility, 2004.
http://www.sosy-lab.org/~dbeyer/Chic/
Contributor, new formalism, and verification algorithm.

CPAchecker: Configurable software verification, 2007.
http://www.sosy-lab.org/~dbeyer/CPAchecker/
Principal designer, architect, implementation, and maintenance.

CrocoPat: Relational programming (for software-structure analysis), 2003.
http://www.sosy-lab.org/~dbeyer/CrocoPat/
Principal designer and implementer.

CSIsat: Interpolation for LA+EUF, 2008.
http://www.sosy-lab.org/~dbeyer/CSIsat/
Contributor and designer.

DepDigger: Detecting complex low-level dependencies, 2010.
http://www.sosy-lab.org/~dbeyer/DepDigger/
Principal designer, architect, and maintenance.

JavaSMT: A unified interface for SMT solvers in Java, 2015.
https://github.com/sosy-lab/java-smt
Contributor.

Rabbit: Verification of real-time systems, 1998.
http://www.sosy-lab.org/~dbeyer/Rabbit/
Principal designer and implementer.

Publications

Electronic versions are available at http://www.sosy-lab.org/∼dbeyer/Publications.

Books

1. Dirk Beyer and Damien Zufferey, editors. *Proceedings of the 21st International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI)*, LNCS 11990. Springer, 2020.
2. Dirk Beyer and Chantal Keller, editors. *Proceedings of the 13th International Conference on Tests and Proofs (TAP)*, LNCS 11823. Springer, 2019.
3. Dirk Beyer, Marieke Huisman, Fabrice Kordon, and Bernhard Steffen, editors. *Proceedings of the 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), Part 3*, LNCS 11429. Springer, 2019.
4. Dirk Beyer and Marieke Huisman, editors. *Proceedings of the 24th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), Part 2*, LNCS 10806. Springer, 2018.
5. Dirk Beyer and Marieke Huisman, editors. *Proceedings of the 24th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), Part 1*, LNCS 10805. Springer, 2018.
6. Dirk Beyer and Michele Boreale, editors. *Proceedings of the 2013 IFIP Joint International Conference on Formal Techniques for Distributed Systems (33rd FORTE / 15th FMOODS)*, LNCS 7892. Springer-Verlag, Heidelberg, 2013.
7. Dirk Beyer, Arie van Deursen, and Michael W. Godfrey, editors. *Proceedings of the 20th IEEE International Conference on Program Comprehension*. IEEE, 2012.

8. Dirk Beyer. *Formale Verifikation von Realzeit-Systemen mittels Cottbus Timed Automata*. Mensch & Buch Verlag, Berlin, 2002. Also: Dissertation, Brandenburgische Technische Universität Cottbus, 2002.

Journal Papers

1. Dirk Beyer, Matthias Dangl, Daniel Dietsch, Matthias Heizmann, Thomas Lemberger, and Michael Tautschnig. Verification witnesses. *ACM Trans. Softw. Eng. Methodol.*, 31(4):57:1–57:69, 2022.
2. Dirk Beyer, Marieke Huisman, Fabrice Kordon, and Bernhard Steffen. TOOLympics II: Competitions on formal methods (intro). *International Journal on Software Tools for Technology Transfer (STTT)*, 23(6):879–881, 2021.
3. Dirk Beyer. First international competition on software testing. *International Journal on Software Tools for Technology Transfer (STTT)*, 23(6):833–846, 2021.
4. Dirk Beyer and Marieke Huisman. TOOLympics I: Competition on software testing (intro). *International Journal on Software Tools for Technology Transfer (STTT)*, 23(6):829–832, 2021.
5. Dirk Beyer and Marie-Christine Jakobs. Cooperative verifier-based testing with CoVeriTest. *International Journal on Software Tools for Technology Transfer (STTT)*, 23(3):313–333, 2021.
6. Dirk Beyer and Marieke Huisman. Tools for the construction and analysis of systems (intro). *International Journal on Software Tools for Technology Transfer (STTT)*, 22(6):685–687, 2020.
7. Dirk Beyer and Marieke Huisman. Selected and extended papers from TACAS 2018: Preface. *Journal of Automated Reasoning*, 64(7):1331–1332, 2020.
8. Dirk Beyer, Stefan Löwe, and Philipp Wendler. Reliable benchmarking: Requirements and solutions. *International Journal on Software Tools for Technology Transfer (STTT)*, 21(1):1–29, 2019.
9. Dirk Beyer, Sumit Gulwani, and David Schmidt. Combining model checking and data-flow analysis. In E. M. Clarke, T. A. Henzinger, H. Veith, and R. Bloem, editors, *Handbook on Model Checking*, pages 493–540. Springer, 2018.
10. Dirk Beyer, Matthias Dangl, and Philipp Wendler. A unifying view on SMT-based software verification. *Journal of Automated Reasoning*, 60(3):299–335, 2018.
11. Dirk Beyer, Rolf Hennicker, Martin Hofmann, Tobias Nipkow, and Martin Wirsing. Softwareverifikation. In A. Bode, M. Broy, H.-J. Bungartz, and F. Matthes, editors, *50 Jahre Universitäts-Informatik in München*, pages 75–86. Springer, 2017.
12. Dirk Beyer and Andreas Stahlbauer. BDD-based software verification: Applications to event-condition-action systems. *International Journal on Software Tools for Technology Transfer (STTT)*, 16(5):507–518, 2014.
13. Falk Howar, Malte Isberner, Maik Merten, Bernhard Steffen, Dirk Beyer, and Corina S. Pasareanu. Rigorous examination of reactive systems: The RERS challenges 2012 and 2013. *International Journal on Software Tools for Technology Transfer (STTT)*, 16(5):457–464, 2014.
14. Dirk Beyer, Thomas A. Henzinger, Ranjit Jhala, and Rupak Majumdar. The software model checker Blast: Applications to software engineering. *International Journal on Software Tools for Technology Transfer (STTT)*, 9(5-6):505–525, 2007. Invited to special issue of selected papers from FASE 2004/05.
15. Dirk Beyer, Andreas Noack, and Claus Lewerentz. Efficient relational calculation for software analysis. *IEEE Transactions on Software Engineering (TSE)*, 31(2):137–149, 2005. Invited to special issue of selected papers from WCRE 2003.
16. Dirk Beyer. Formale verifikation von realzeit-systemen mittels cottbus timed automata (zusammenfassung). *Softwaretechnik-Trends*, 23(2):4, May 2003.

Conference Papers (with published proceedings)

1. Dirk Beyer, Po-Chun Chien, and Nian-Ze Lee. CPA-DF: A tool for configurable interval analysis to boost program verification. In *Proc. ASE*, 2023.
2. Dirk Beyer and Martin Spiessl. Liv: Invariant validation using straight-line programs. In *Proc. ASE*, 2023.
3. Dirk Beyer, Marian Lingsch-Rosenfeld, and Martin Spiessl. Cegar-pt: A tool for abstraction by program transformation. In *Proc. ASE*, 2023.
4. Dirk Beyer, Sudeep Kanav, and Henrik Wachowitz. CoVeriTeam service: Verification as a service. In *Proc. ICSE*, pages 21–25. IEEE, 2023.
5. Dirk Beyer. Software testing: 5th comparative evaluation: Test-Comp 2023. In L. Lambers and S. Uchitel, editors, *Proceedings of the 26th International Conference on Fundamental Approaches*

to Software Engineering (FASE 2023, Paris, France, April 22-27), LNCS 13991, pages 309–323. Springer, 2023.

6. Dirk Beyer. Competition on software verification and witness validation: SV-COMP 2023. In S. Sankaranarayanan and N. Sharygina, editors, *Proceedings of the 29th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2023, Paris, France, April 22-27)*, LNCS 13994, pages 495–522. Springer, 2023.
7. Dirk Beyer, Po-Chun Chien, and Nian-Ze Lee. Bridging hardware and software analysis with Btor2C: A word-level-circuit-to-C translator. In *Proc. TACAS*, LNCS 13994, pages 1–21. Springer, 2023.
8. Dirk Beyer and Andreas Podelski. Software model checking: 20 years and beyond. In *Principles of Systems Design*, LNCS 13660, pages 554–582. Springer, 2022.
9. Dirk Beyer and Jan Strejček. Case study on verification-witness validators: Where we are and where we go. In *Proceedings of the 29th International Symposium on Static Analysis, (SAS 2022, Auckland, New Zealand, December 5-7, 2022)*, LNCS 13790, pages 160–174. Springer, 2022.
10. Stefan Winter, Christopher Steven Timperley, Ben Hermann, Jürgen Cito, Jonathan Bell, Michael Hilton, and Dirk Beyer. A retrospective study of one decade of artifact evaluations. In Abhik Roychoudhury, Cristian Cadar, and Miryung Kim, editors, *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2022, Singapore, Singapore, November 14-18*, pages 145–156. ACM, 2022.
11. Dirk Beyer, Martin Spiessl, and Sven Umbricht. Cooperation between automatic and interactive software verifiers. In Bernd-Holger Schlingloff and Ming Chai, editors, *Proceedings of the 20th International Conference on Software Engineering and Formal Methods, (SEFM 2022, Berlin, Germany, September 26-30*, LNCS 13550, page 111–128. Springer, 2022.
12. Dirk Beyer, Marian Lingsch Rosenfeld, and Martin Spiessl. A unifying approach for control-flow-based loop abstraction. In Bernd-Holger Schlingloff and Ming Chai, editors, *Proceedings of the 20th International Conference on Software Engineering and Formal Methods, (SEFM 2022, Berlin, Germany, September 26-30*, LNCS 13550, pages 3–19. Springer, 2022.
13. Dirk Beyer, Jan Haltermann, Thomas Lemberger, and Heike Wehrheim. Decomposing software verification into off-the-shelf components: An application to CEGAR. In *Proceedings of the 44th International Conference on Software Engineering (ICSE 2022, Pittsburgh, PA, USA, May 8-20 (Virtual), May 22-27 (In-Person))*, pages 536–548. ACM, 2022.
14. Dirk Beyer and Martin Spiessl. The static analyzer Frama-C in SV-COMP (competition contribution). In Dana Fisman and Grigore Rosu, editors, *Proceedings of the 28th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2022, Munich, Germany, April 2-7*, LNCS 13244, pages 429–434. Springer, 2022.
15. Dirk Beyer. Progress on software verification: SV-COMP 2022. In D. Fisman and G. Rosu, editors, *Proceedings of the 28th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2022, Munich, Germany, April 2-7*, LNCS 13244, pages 375–402. Springer, 2022.
16. Dirk Beyer and Sudeep Kanav. CoVeriTeam: On-demand composition of cooperative verification systems. In D. Fisman and G. Rosu, editors, *Proceedings of the 28th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2022, Munich, Germany, April 2-7*, LNCS 13243, pages 561–579. Springer, 2022.
17. Dirk Beyer. Advances in automatic software testing: Test-Comp 2022. In E. B. Johnsen and M. Wimmer, editors, *Proceedings of the 25th International Conference on Fundamental Approaches to Software Engineering (FASE 2022, Munich, Germany, April 2-7)*, LNCS 13241, pages 321–335. Springer, 2022.
18. Dirk Beyer, Sudeep Kanav, and Cedric Richter. Construction of verifier combinations based on off-the-shelf verifiers. In E. B. Johnsen and M. Wimmer, editors, *Proceedings of the 25th International Conference on Fundamental Approaches to Software Engineering (FASE 2022, Munich, Germany, April 2-7)*, LNCS 13241, pages 49–70. Springer, 2022.
19. Dirk Beyer, Karlheinz Friedberger, and Stephan Holzner. PJBDD: A BDD library for Java and multi-threading. In *Proceedings of the 19th International Symposium on Automated Technology for Verification and Analysis (ATVA21 2021, Gold Coast (Online), Australia, October 18-22)*. Springer, 2021.
20. Daniel Baier, Dirk Beyer, and Karlheinz Friedberger. Javasmt 3: Interacting with smt solvers in java. In A. Silva and K. R. M. Leino, editors, *Proceedings of the 33rd International Conference on Computer-Aided Verification (CAV 2021, Los Angeles, California, USA, July 18-24)*, LNCS 12760, pages 1–13. Springer, 2021.
21. Dirk Beyer. Software verification: 10th comparative evaluation (SV-COMP 2021). In J. F. Groote and K. G. Larsen, editors, *Proceedings of the 27th International Conference on Tools and Algorithms for*

the Construction and Analysis of Systems (TACAS 2021, Luxembourg, Luxembourg, March 27 - April 1), part 2, LNCS 12652, pages 401–422. Springer, 2021.

22. Dirk Beyer. Status report on software testing: Test-Comp 2021. In E. Guerra and M. Stoelinga, editors, *Proceedings of the 24th International Conference on Fundamental Approaches to Software Engineering (FASE 2021, Luxembourg, Luxembourg, March 27 - April 1)*, LNCS 12649, pages 341–357. Springer, 2021.
23. Dirk Beyer and Karlheinz Friedberger. Domain-independent interprocedural program analysis using block-abstraction memoization. In P. Devanbu, M. Cohen, and T. Zimmermann, editors, *Proceedings of the 28th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2020, Virtual Event, USA, November 8-13)*, pages 50–62. ACM, 2020.
24. Dirk Beyer and Karlheinz Friedberger. Violation witnesses and result validation for multi-threaded programs. In T. Margaria and B. Steffen, editors, *Proceedings of the 9th International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2020, Rhodos, Greece, October 26-30), part 1*, LNCS 12476, pages 449–470. Springer, 2020.
25. Dirk Beyer and Sudeep Kanav. An interface theory for program verification. In T. Margaria and B. Steffen, editors, *Proceedings of the 9th International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2020, Rhodos, Greece, October 26-30), part 1*, LNCS 12476, pages 168–186. Springer, 2020.
26. Dirk Beyer and Heike Wehrheim. Verification artifacts in cooperative verification: Survey and unifying component framework. In T. Margaria and B. Steffen, editors, *Proceedings of the 9th International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2020, Rhodos, Greece, October 26-30), part 1*, LNCS 12476, pages 143–167. Springer, 2020.
27. Dirk Beyer, Marie-Christine Jakobs, and Thomas Lemberger. Difference verification with conditions. In F. d. Boer and A. Cerone, editors, *Proceedings of the 18th International Conference on Software Engineering and Formal Methods (SEFM 2020, Virtual, Netherlands, September 14-18)*, LNCS 12310, pages 133–154. Springer, 2020.
28. Dirk Beyer and Marie-Christine Jakobs. FRed: Conditional model checking via reducers and folders. In F. d. Boer and A. Cerone, editors, *Proceedings of the 18th International Conference on Software Engineering and Formal Methods (SEFM 2020, Virtual, Netherlands, September 14-18)*, LNCS 12310, pages 113–132. Springer, 2020.
29. Dirk Beyer and Martin Spiessl. Metaval: Witness validation via verification. In S. K. Lahiri and C. Wang, editors, *Proceedings of the 32nd International Conference on Computer Aided Verification (CAV 2020, Virtual, USA, July 21-24), part 2*, LNCS 12225, pages 165–177. Springer, 2020.
30. Dirk Beyer. Advances in automatic software verification: SV-COMP 2020. In *Proceedings of the 26th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2020, Dublin, Ireland, April 25-30), part 2*, LNCS 12079, pages 347–367. Springer, 2020.
31. Dirk Beyer and Philipp Wendler. Cpu energy meter: A tool for energy-aware algorithms engineering. In *Proceedings of the 26th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2020, Dublin, Ireland, April 25-30), part 2*, LNCS 12079, pages 126–133. Springer, 2020.
32. Dirk Beyer and Matthias Dangl. Software verification with PDR: An implementation of the state of the art. In *Proceedings of the 26th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2020, Dublin, Ireland, April 25-30), part 1*, LNCS 12078, pages 3–21. Springer, 2020.
33. Dirk Beyer. Second competition on software testing: Test-Comp 2020. In *Proceedings of the 23rd International Conference on Fundamental Approaches to Software Engineering (FASE 2020, Dublin, Ireland, April 25-30)*, LNCS 12076, pages 505–519. Springer, 2020.
34. Dirk Beyer and Thomas Lemberger. TestCov: Robust test-suite execution and coverage measurement. In *Proceedings of the 34th IEEE/ACM International Conference on Automated Software Engineering (ASE 2019, San Diego, CA, USA, November 11-15)*, pages 1074–1077. IEEE, 2019.
35. Dirk Beyer and Thomas Lemberger. Conditional testing - off-the-shelf combination of test-case generators. In Yu-Fang Chen, Chih-Hong Cheng, and Javier Esparza, editors, *Proceedings of the 17th International Symposium on Automated Technology for Verification and Analysis (ATVA 2019, Taipei, Taiwan, October 28-31)*, LNCS 11781, pages 189–208. Springer, 2019.
36. Dirk Beyer. A data set of program invariants and error paths. In *Proceedings of the 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR 2019, Montreal, Canada, May 26-27)*, pages 111–115. IEEE, 2019.

37. Dirk Beyer. International competition on software testing (test-comp). In *Proceedings of the 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2019, Prague, Czech Republic, April 6-11), part 3*, LNCS 11429, pages 167–175. Springer, 2019.

38. Dirk Beyer. Automatic verification of C and java programs: SV-COMP 2019. In *Proceedings of the 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2019, Prague, Czech Republic, April 6-11), part 3*, LNCS 11429, pages 133–155. Springer, 2019.

39. E. Bartocci, D. Beyer, P. E. Black, G. Fedyukovich, H. Garavel, A. Hartmanns, M. Huisman, F. Kordon, J. Nagele, M. Sighireanu, B. Steffen, M. Suda, G. Sutcliffe, T. Weber, and A. Yamada. TOOLympics 2019: An overview of competitions in formal methods. In *Proceedings of the 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2019, Prague, Czech Republic, April 6-11), part 3*, LNCS 11429, pages 3–24. Springer, 2019.

40. Dirk Beyer and Marie-Christine Jakobs. Coveritest: Cooperative verifier-based testing. In *Proceedings of the 22nd International Conference on Fundamental Approaches to Software Engineering (FASE 2019, Prague, Czech Republic, April 6-11)*, LNCS 11424, pages 389–408. Springer, 2019.

41. Dirk Beyer and Karlheinz Friedberger. In-place vs. copy-on-write cegar refinement for block summarization with caching. In T. Margaria and B. Steffen, editors, *Proceedings of the 8th International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2018, Part 2, Limassol, Cyprus, November 5-9)*, LNCS 11245, pages 197–215. Springer, 2018.

42. Dirk Beyer and Matthias Dangl. Strategy selection for software verification based on boolean features: A simple but effective approach. In T. Margaria and B. Steffen, editors, *Proceedings of the 8th International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2018, Part 2, Limassol, Cyprus, November 5-9)*, LNCS 11245, pages 144–159. Springer, 2018.

43. Dirk Beyer and Thomas Lemberger. CPA-SymExec: Efficient symbolic execution in CPAchecker. In Marianne Huchard, Christian Kästner, and Gordon Fraser, editors, *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering (ASE 2018, Montpellier, France, September 3-7)*, pages 900–903. ACM, 2018.

44. Dirk Beyer and Karlheinz Friedberger. Domain-independent multi-threaded software model checking. In Marianne Huchard, Christian Kästner, and Gordon Fraser, editors, *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ASE 2018, Montpellier, France, September 3-7, 2018*, pages 634–644. ACM, 2018.

45. Dirk Beyer, Matthias Dangl, Thomas Lemberger, and Michael Tautschnig. Tests from witnesses: Execution-based validation of verification results. In Catherine Dubois and Burkhart Wolff, editors, *Proceedings of the 12th International Conference on Tests and Proofs (TAP 2018, Toulouse, France, June 27-29)*, LNCS 10889, pages 3–23. Springer, 2018.

46. Dirk Beyer, Marie-Christine Jakobs, Thomas Lemberger, and Heike Wehrheim. Reducer-based construction of conditional verifiers. In *Proceedings of the 40th International Conference on Software Engineering (ICSE 2018, Gothenburg, Sweden, May 27 - June 3)*, pages 1182–1193. ACM, 2018.

47. Dirk Beyer and Thomas Lemberger. Software verification: Testing vs. model checking. In O. Strichman and R. Tzoref-Brill, editors, *Proceedings of the 13th Haifa Verification Conference (HVC 2017, Haifa, Israel, November 13-25)*, LNCS 10629, pages 99–114. Springer, 2017.

48. Dirk Beyer. Software verification with validation of results (Report on SV-COMP 2017). In A. Legay and T. Margaria, editors, *Proceedings of the 23rd International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2017, Uppsala, Sweden, April 22-29)*, LNCS 10206, pages 331–349. Springer-Verlag, Heidelberg, 2017.

49. Dirk Beyer, Matthias Dangl, Daniel Dietsch, and Matthias Heizmann. Correctness witnesses: Exchanging verification results between verifiers. In T. Zimmermann, J. Cleland-Huang, and Z. Su, editors, *Proceedings of the 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2016, Seattle, WA, USA, November 13-18)*, pages 326–337. ACM, 2016.

50. Sven Apel, Dirk Beyer, Vitaly Mordan, Vadim Mutilin, and Andreas Stahlbauer. On-the-fly decomposition of specifications in software model checking. In T. Zimmermann, J. Cleland-Huang, and Z. Su, editors, *Proceedings of the 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2016, Seattle, WA, USA, November 13-18)*, pages 349–361. ACM, 2016.

51. Dirk Beyer. Partial verification and intermediate results as a solution to combine automatic and interactive verification techniques. In T. Margaria and B. Steffen, editors, *7th International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2016, Part 1, Imperial, Corfu, Greece, October 10-14)*, LNCS 9952, pages 874–880. Springer, 2016.

52. Dirk Beyer and Thomas Lemberger. Symbolic execution with CEGAR. In T. Margaria and B. Steffen, editors, *7th International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2016, Part 1, Imperial, Corfu, Greece, October 10-14)*, LNCS 9952, pages 195–211. Springer, 2016.

53. Dirk Beyer and Matthias Dangl. Verification-aided debugging: An interactive web-service for exploring error witnesses. In S. Chaudhuri and A. Farzan, editors, *28th International Conference on Computer Aided Verification (CAV 2016, Part 2, Toronto, ON, Canada, July 17-23)*, LNCS 9780, pages 502–509. Springer, 2016.

54. Dirk Beyer. Reliable and reproducible competition results with BenchExec and witnesses (Report on SV-COMP 2016). In M. Chechik and J.-F. Raskin, editors, *Proceedings of the 22nd International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2016, Eindhoven, The Netherlands, April 2-8)*, LNCS 9636, pages 887–904. Springer-Verlag, Heidelberg, 2016.

55. Dirk Beyer, Matthias Dangl, Daniel Dietsch, Matthias Heizmann, and Andreas Stahlbauer. Witness validation and stepwise testification across software verifiers. In E. Di Nitto, M. Harman, and P. Heymans, editors, *Proceedings of the 2015 10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on Foundations of Software Engineering (ESEC/FSE 2015, Bergamo, Italy, August 31 - September 4)*, pages 721–733. ACM, New York, 2015.

56. Dirk Beyer, Stefan Löwe, and Philipp Wendler. Refinement selection. In B. Fischer and J. Geldenhuys, editors, *Proceedings of the 22nd International Symposium on Model Checking of Software (SPIN 2015, Stellenbosch, South Africa, August 24-26)*, LNCS 9232, pages 20–38. Springer-Verlag, Heidelberg, 2015.

57. Dirk Beyer, Stefan Löwe, and Philipp Wendler. Benchmarking and resource measurement. In B. Fischer and J. Geldenhuys, editors, *Proceedings of the 22nd International Symposium on Model Checking of Software (SPIN 2015, Stellenbosch, South Africa, August 24-26)*, LNCS 9232, pages 160–178. Springer-Verlag, Heidelberg, 2015.

58. Dirk Beyer, Matthias Dangl, and Philipp Wendler. Boosting k-induction with continuously-refined invariants. In D. Kröning and C. S. Pasareanu, editors, *Proceedings of the 27th International Conference on Computer Aided Verification (CAV 2015, San Francisco, CA, USA, July 18-24)*, LNCS 9206, pages 622–640. Springer-Verlag, Heidelberg, 2015.

59. Dirk Beyer, Stefan Löwe, and Philipp Wendler. Sliced path prefixes: An effective method to enable refinement selection. In S. Graf and M. Viswanathan, editors, *Proceedings of the 35th IFIP WG 6.1 International Conference on Formal Techniques for Distributed Objects, Components, and Systems (FORTE 2015, Grenoble, France, June 2-4)*, LNCS 9039, pages 228–243. Springer-Verlag, Heidelberg, 2015.

60. Alexander von Rhein, Alexander Grebhahn, Sven Apel, Norbert Siegmund, Dirk Beyer, and Thorsten Berger. Presence-condition simplification in highly configurable systems. In A. Bertolino, G. Canfora, and S. Elbaum, editors, *Proceedings of the 37th International Conference on Software Engineering (ICSE 2015, Florence, Italy, May 16-24)*, pages 178–188. IEEE, 2015.

61. Johannes Bürdek, Malte Lochau, Stefan Bauregger, Andreas Holzer, Alexander von Rhein, Sven Apel, and Dirk Beyer. Facilitating reuse in multi-goal test-suite generation for software product lines. In A. Egyed and I. Schaefer, editors, *Proceedings of the 18th International Conference on Fundamental Approaches to Software Engineering (FASE 2015, London, UK, April 13-15)*, LNCS 9033, pages 84–99. Springer-Verlag, Heidelberg, 2015.

62. Dirk Beyer. Software verification and verifiable witnesses (report on SV-COMP 2015). In C. Baier and C. Tinelli, editors, *Proceedings of the 21st International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2015, London, UK, April 13-17)*, LNCS 9035, pages 401–416. Springer-Verlag, Heidelberg, 2015.

63. Dirk Beyer, Georg Dresler, and Philipp Wendler. Software verification in the Google App-Engine cloud. In A. Biere and R. Bloem, editors, *Proceedings of the 26th International Conference on Computer-Aided Verification (CAV 2014, Vienna, Austria, July 18-22)*, LNCS 8559, pages 327–333. Springer-Verlag, Heidelberg, 2014.

64. Dirk Beyer and Peter Häring. A formal evaluation of DepDegree based on Weyuker's properties. In C. Roy, A. Begel, and L. Moonen, editors, *Proceedings of the 22nd International Conference on Program Comprehension (ICPC 2014, Hyderabad, India, June 2-3)*, pages 258–261. ACM, 2014.

65. Dirk Beyer. Status report on software verification (competition summary SV-COMP 2014). In E. Abraham and K. Havelund, editors, *Proceedings of the 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2014, Grenoble, France, April 5-13)*, LNCS 8413, pages 373–388. Springer-Verlag, Heidelberg, 2014.

66. Sven Apel, Dirk Beyer, Karlheinz Friedberger, Franco Raimondi, and Alexander von Rhein. Domain types: Abstract-domain selection based on variable usage. In V. Bertacco and A. Legay, editors, *Proceedings of the 9th Haifa Verification Conference (HVC 2013, Haifa, Israel, November 5-7)*, LNCS 8244, pages 262–278. Springer-Verlag, Heidelberg, 2013.

67. Dirk Beyer, Stefan Löwe, Evgeny Novikov, Andreas Stahlbauer, and Philipp Wendler. Precision reuse for efficient regression verification. In B. Meyer, L. Baresi, and M. Mezini, editors, *Proceedings of the 9th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on Foundations of Software Engineering (ESEC/FSE 2013, St. Petersburg, Russia, August 18-26)*, pages 389–399. ACM, 2013.

68. Dirk Beyer and Philipp Wendler. Reuse of verification results: Conditional model checking, precision reuse, and verification witnesses. In E. Bartocci and C. R. Ramakrishnan, editors, *Proceedings of the 2013 International Symposium on Model Checking of Software (SPIN 2013, Stony Brook, NY, USA, July 8-9)*, LNCS 7976, pages 1–17. Springer-Verlag, Heidelberg, 2013.

69. Sven Apel, Alexander von Rhein, Philipp Wendler, Armin Größlinger, and Dirk Beyer. Strategies for product-line verification: Case studies and experiments. In D. Notkin, B. H. C. Cheng, and K. Pohl, editors, *Proceedings of the 35th International Conference on Software Engineering (ICSE 2013, San Francisco, CA, USA, May 18-26)*, pages 482–491. IEEE, 2013.

70. Dirk Beyer. Second competition on software verification (Summary of SV-COMP 2013). In N. Piterman and S. Smolka, editors, *Proceedings of the 19th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2013, Rome, Italy, March 16-24)*, LNCS 7795, pages 594–609. Springer-Verlag, Heidelberg, 2013.

71. Dirk Beyer, Andreas Holzer, Michael Tautschnig, and Helmut Veith. Information reuse for multi-goal reachability analyses. In M. Felleisen and P. Gardner, editors, *Proceedings of the 22nd European Symposium on Programming (ESOP 2013, Rome, Italy, March 19-22)*, LNCS 7792, pages 472–491. Springer-Verlag, Heidelberg, 2013.

72. Dirk Beyer and Stefan Löwe. Explicit-state software model checking based on CEGAR and interpolation. In V. Cortellessa and D. Varro, editors, *Proceedings of the 16th International Conference on Fundamental Approaches to Software Engineering (FASE 2013, Rome, Italy, March 20-22)*, LNCS 7793, pages 146–162. Springer-Verlag, Heidelberg, 2013.

73. Dirk Beyer, Thomas A. Henzinger, M. Erkan Keremoglu, and Philipp Wendler. Conditional model checking: A technique to pass information between verifiers. In Tevfik Bultan and Martin Robillard, editors, *Proceedings of the 20th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE 2012, Cary, NC, November 10-17)*. ACM, 2012.

74. Dirk Beyer and Philipp Wendler. Algorithms for software model checking: Predicate abstraction vs. IMPACT. In Gianpiero Cabodi and Satnam Singh, editors, *Proceedings of the 12th International Conference on Formal Methods in Computer-Aided Design (FMCAD 2012, Cambrige, UK, October 22-25)*, pages 106–113. FMCAD, 2012.

75. Dirk Beyer. Competition on software verification (SV-COMP). In C. Flanagan and B. König, editors, *Proceedings of the 18th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2012, Tallinn, Estonia, March 27-30)*, LNCS 7214, pages 504–524. Springer-Verlag, Heidelberg, 2012.

76. Sven Apel, Hendrik Speidel, Philipp Wendler, Alexander von Rhein, and Dirk Beyer. Detection of feature interactions using feature-aware verification. In *Proceedings of the 26th International Conference on Automated Software Engineering (ASE 2011, Lawrence, KS, November 6-10)*, pages 372–375. IEEE, 2011.

77. Dirk Beyer and M. Erkan Keremoglu. CPAchecker: A tool for configurable software verification. In G. Gopalakrishnan and S. Qadeer, editors, *Proceedings of the 23rd International Conference on Computer Aided Verification (CAV 2011, Snowbird, UT, July 14-20)*, LNCS 6806, pages 184–190. Springer-Verlag, Heidelberg, 2011.

78. Sven Apel and Dirk Beyer. Feature cohesion in software product lines: An exploratory study. In *Proceedings of the 33rd International Conference on Software Engineering (ICSE 2011, Honolulu, HI, May 21-28)*, pages 421–430. ACM Press, New York (NY), 2011.

79. Dirk Beyer, M. Erkan Keremoglu, and Philipp Wendler. Predicate abstraction with adjustable-block encoding. In *Proceedings of the 10th International Conference on Formal Methods in Computer-Aided Design (FMCAD 2010, Lugano, October 20-23)*, pages 189–197. FMCAD, 2010.

80. Dirk Beyer and Ashgan Fararooy. CheckDep: A tool for tracking software dependencies. In *Proceedings of the 18th IEEE International Conference on Program Comprehension (ICPC 2010, Braga, June 30 - July 2)*, pages 42–43. IEEE Computer Society Press, Los Alamitos (CA), 2010.

81. Dirk Beyer and Ashgan Fararooy. DEPDIGGER: A tool for detecting complex low-level dependencies. In *Proceedings of the 18th IEEE International Conference on Program Comprehension (ICPC 2010, Braga, June 30 - July 2)*, pages 40–41. IEEE Computer Society Press, Los Alamitos (CA), 2010.

82. Dirk Beyer and Ashgan Fararooy. A simple and effective measure for complex low-level dependencies. In *Proceedings of the 18th IEEE International Conference on Program Comprehension (ICPC 2010, Braga, June 30 - July 2)*, pages 80–83. IEEE Computer Society Press, Los Alamitos (CA), 2010.

83. Dirk Beyer, Thomas A. Henzinger, Grégory Théoduloz, and Damien Zufferey. Shape refinement through explicit heap analysis. In D.S. Rosenblum and G. Taentzer, editors, *Proceedings of the 13th International Conference on Fundamental Approaches to Software Engineering (FASE 2010, Paphos, Cyprus, March 22-26)*, LNCS 6013, pages 263–277. Springer-Verlag, Heidelberg, 2010.

84. Dirk Beyer, Alessandro Cimatti, Alberto Griggio, M. Erkan Keremoglu, and Roberto Sebastiani. Software model checking via large-block encoding. In *Proceedings of the 9th International Conference on Formal Methods in Computer-Aided Design (FMCAD 2009, Austin, TX, November 15-18)*, pages 25–32. IEEE Computer Society Press, Los Alamitos (CA), 2009.

85. Dirk Beyer, Thomas A. Henzinger, and Grégory Théoduloz. Program analysis with dynamic precision adjustment. In *Proceedings of the 23rd IEEE/ACM International Conference on Automated Software Engineering (ASE 2008, L'Aquila, September 15-19)*, pages 29–38. IEEE Computer Society Press, Los Alamitos (CA), 2008.

86. Dirk Beyer, Damien Zufferey, and Rupak Majumdar. CSISAT: Interpolation for LA+EUF. In A. Gupta and S. Malik, editors, *Proceedings of the 20th International Conference on Computer Aided Verification (CAV 2008, Princeton, NY, July 7-14)*, LNCS 5123, pages 304–308. Springer-Verlag, Heidelberg, 2008.

87. Dirk Beyer. CCVISU: Automatic visual software decomposition. In *Proceedings of the 30th ACM/IEEE International Conference on Software Engineering (ICSE 2008, Leipzig, May 10-18)*, pages 967–968. ACM Press, New York (NY), 2008.

88. Dirk Beyer, Arindam Chakrabarti, Thomas A. Henzinger, and Sanjit A. Seshia. An application of web-service interfaces. In *Proceedings of the 2007 IEEE International Conference on Web Services (ICWS 2007, Salt Lake City, UT, July 9-13)*, pages 831–838. IEEE Computer Society Press, Los Alamitos (CA), 2007.

89. Dirk Beyer, Thomas A. Henzinger, and Vasu Singh. Algorithms for interface synthesis. In W. Damm and H. Hermanns, editors, *Proceedings of the 19th International Conference on Computer Aided Verification (CAV 2007, Berlin, July 3-7)*, LNCS 4590, pages 4–19. Springer-Verlag, Heidelberg, 2007.

90. Dirk Beyer, Thomas A. Henzinger, and Grégory Théoduloz. Configurable software verification: Concretizing the convergence of model checking and program analysis. In W. Damm and H. Hermanns, editors, *Proceedings of the 19th International Conference on Computer Aided Verification (CAV 2007, Berlin, July 3-7)*, LNCS 4590, pages 504–518. Springer-Verlag, Heidelberg, 2007.

91. Dirk Beyer, Thomas A. Henzinger, Rupak Majumdar, and Andrey Rybalchenko. Path invariants. In *Proceedings of the 2007 ACM Conference on Programming Language Design and Implementation (PLDI 2007, San Diego, CA, June 10-13)*, pages 300–309. ACM Press, New York (NY), 2007.

92. Dirk Beyer, Thomas A. Henzinger, Rupak Majumdar, and Andrey Rybalchenko. Invariant synthesis for combined theories. In B. Cook and A. Podelski, editors, *Proceedings of the Eighth International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI 2007, Nice, January 14-16)*, LNCS 4349, pages 378–394. Springer-Verlag, Heidelberg, 2007.

93. Dirk Beyer and Ahmed E. Hassan. Animated visualization of software history using evolution storyboards. In *Proceedings of the 13th IEEE Working Conference on Reverse Engineering (WCRE 2006, Benevento, October 23-27)*, pages 199–208. IEEE Computer Society Press, Los Alamitos (CA), 2006.

94. Dirk Beyer, Thomas A. Henzinger, and Grégory Théoduloz. Lazy shape analysis. In T. Ball and R.B. Jones, editors, *Proceedings of the 18th International Conference on Computer Aided Verification (CAV 2006, Seattle, WA, August 17-20)*, LNCS 4144, pages 532–546. Springer-Verlag, Heidelberg, 2006.

95. Dirk Beyer and Ahmed E. Hassan. Evolution storyboards: Visualization of software structure dynamics. In *Proceedings of the 14th IEEE International Conference on Program Comprehension (ICPC 2006, Athens, June 14-16)*, pages 248–251. IEEE Computer Society Press, Los Alamitos (CA), 2006.

96. Dirk Beyer. Relational programming with CrocoPat. In *Proceedings of the 28th ACM/IEEE International Conference on Software Engineering (ICSE 2006, Shanghai, May 20-28)*, pages 807–810. ACM Press, New York (NY), 2006.

97. Basil Becker, Dirk Beyer, Holger Giese, Florian Klein, and Daniela Schilling. Symbolic invariant verification for systems with dynamic structural adaptation. In *Proceedings of the 28th ACM/IEEE International Conference on Software Engineering (ICSE 2006, Shanghai, May 20-28)*, pages 72–81. ACM Press, New York (NY), 2006.

98. Dirk Beyer. Co-change visualization. In *Proceedings of the 21st IEEE International Conference on Software Maintenance (ICSM 2005, Budapest, September 25-30), Industrial and Tool volume*, pages 89–92, Budapest, 2005.

99. Dirk Beyer and Andreas Noack. Clustering software artifacts based on frequent common changes. In *Proceedings of the 13th IEEE International Workshop on Program Comprehension (IWPC 2005, St. Louis, MO, May 15-16)*, pages 259–268. IEEE Computer Society Press, Los Alamitos (CA), 2005.

100. Dirk Beyer, Arindam Chakrabarti, and Thomas A. Henzinger. Web service interfaces. In *Proceedings of the 14th ACM International World Wide Web Conference (WWW 2005, Chiba, May 10-14)*, pages 148–159. ACM Press, New York (NY), 2005.

101. Dirk Beyer, Thomas A. Henzinger, Ranjit Jhala, and Rupak Majumdar. Checking memory safety with Blast. In M. Cerioli, editor, *Proceedings of the Eighth International Conference on Fundamental Approaches to Software Engineering (FASE 2005, Edinburgh, April 2-10)*, LNCS 3442, pages 2–18. Springer-Verlag, Heidelberg, 2005.

102. Dirk Beyer, Adam J. Chlipala, Thomas A. Henzinger, Ranjit Jhala, and Rupak Majumdar. The Blast query language for software verification. In R. Giacobazzi, editor, *Proceedings of the 11th International Static Analysis Symposium (SAS 2004, Verona, August 26-28)*, LNCS 3148, pages 2–18. Springer-Verlag, Heidelberg, 2004.

103. Dirk Beyer, Thomas A. Henzinger, Ranjit Jhala, and Rupak Majumdar. An Eclipse plug-in for model checking. In *Proceedings of the 12th IEEE International Workshop on Program Comprehension (IWPC 2004, Bari, June 24-26)*, pages 251–255. IEEE Computer Society Press, Los Alamitos (CA), 2004.

104. Dirk Beyer, Adam J. Chlipala, Thomas A. Henzinger, Ranjit Jhala, and Rupak Majumdar. Generating tests from counterexamples. In *Proceedings of the 26th IEEE International Conference on Software Engineering (ICSE 2004, Edinburgh, May 26-28)*, pages 326–335. IEEE Computer Society Press, Los Alamitos (CA), 2004.

105. Dirk Beyer, Andreas Noack, and Claus Lewerentz. Simple and efficient relational querying of software structures. In *Proceedings of the Tenth IEEE Working Conference on Reverse Engineering (WCRE 2003, Victoria, BC, November 13-16)*, pages 216–225. IEEE Computer Society Press, Los Alamitos (CA), 2003.

106. Dirk Beyer and Andreas Noack. Can decision diagrams overcome state space explosion in real-time verification? In H. König, M. Heiner, and A. Wolisz, editors, *Proceedings of the 23rd IFIP International Conference on Formal Techniques for Networked and Distributed Systems (FORTE 2003, Berlin, September 29 - October 2)*, LNCS 2767, pages 193–208. Springer-Verlag, Heidelberg, 2003.

107. Dirk Beyer, Claus Lewerentz, and Andreas Noack. Rabbit: A tool for BDD-based verification of real-time systems. In W. A. Hunt and F. Somenzi, editors, *Proceedings of the 15th International Conference on Computer Aided Verification (CAV 2003, Boulder, CO, July 8-12)*, LNCS 2725, pages 122–125. Springer-Verlag, Heidelberg, 2003.

108. Dirk Beyer and Claus Lewerentz. CrocoPat: Efficient pattern analysis in object-oriented programs. In *Proceedings of the 11th IEEE International Workshop on Program Comprehension (IWPC 2003, Portland, OR, May 10-11)*, pages 294–295. IEEE Computer Society Press, Los Alamitos (CA), 2003.

109. Dirk Beyer. Efficient reachability analysis and refinement checking of timed automata using BDDs. In T. Margaria and T. F. Melham, editors, *Proceedings of the 11th IFIP Advanced Research Working Conference on Correct Hardware Design and Verification Methods (CHARME 2001, Livingston, September 4-7)*, LNCS 2144, pages 86–91. Springer-Verlag, Heidelberg, 2001.

110. Dirk Beyer. Improvements in BDD-based reachability analysis of timed automata. In J. N. Oliveira and P. Zave, editors, *Proceedings of the Tenth International Symposium of Formal Methods Europe (FME 2001, Berlin, March 12-16): Formal Methods for Increasing Software Productivity*, LNCS 2021, pages 318–343. Springer-Verlag, Heidelberg, 2001.

111. Dirk Beyer, Claus Lewerentz, and Frank Simon. Impact of inheritance on metrics for size, coupling, and cohesion in object oriented systems. In R. Dumke and A. Abran, editors, *Proceedings of the Tenth International Workshop on Software Measurement (IWSM 2000, Berlin, October 4-6): New Approaches in Software Measurement*, LNCS 2006, pages 1–17. Springer-Verlag, Heidelberg, 2001.

112. Dirk Beyer and Heinrich Rust. A tool for modular modelling and verification of hybrid systems. In A. Crespo and J. Vila, editors, *Proceedings of the 25th IFAC/IFIP Workshop on Real-Time Programming (WRTP 2000, Palma, May 17-19)*, pages 169–174. Elsevier Science, Oxford, 2000.

Workshop Papers

1. Dirk Beyer, Jan Haltermann, Thomas Lemberger, and Heike Wehrheim. Component-based CEGAR - building software verifiers from off-the-shelf components. In G. Engels, R. Hebig, and M. Tichy, editors, *Software Engineering 2023, Fachtagung des GI-Fachbereichs Softwaretechnik, 20.-24. Februar 2023, Paderborn*, LNI P-332, pages 37–38. GI, 2023.
2. Markus Schordan, Dirk Beyer, and Irena Bojanova. Software verification tools (track introduction). In T. Margaria and B. Steffen, editors, *Proceedings of the 9th International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2020, Rhodes, Greece, October 20–30), part 4*, LNCS 12479, pages 177–181. Springer-Verlag, Heidelberg, 2010.
3. Dirk Beyer and Marie-Christine Jakobs. Cooperative test-case generation with verifiers. In M. Felderer, W. Hasselbring, R. Rabiser, and R. Jung, editors, *Proceedings of the Conference on Software Engineering (SE 2020, Innsbruck, Austria, February 24-28)*, LNI P-300, pages 107–108. GI, 2020.
4. Dirk Beyer, Marie-Christine Jakobs, Thomas Lemberger, and Heike Wehrheim. Combining verifiers in conditional model checking via reducers. In S. Becker, I. Bogicevic, G. Herzwurm, and S. Wagner, editors, *Proceedings of the Conference on Software Engineering and Software Management (SE/SWM 2019, Stuttgart, Germany, February 18-22)*, LNI P-292, pages 151–152. GI, 2019.
5. Markus Schordan, Dirk Beyer, and Stephen F. Siegel. Evaluating tools for software verification (track introduction). In T. Margaria and B. Steffen, editors, *Proceedings of the 8th International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2018, Limassol, Cyprus, November 5–9), Part 2*, LNCS 11245, pages 139–143. Springer-Verlag, Heidelberg, 2018.
6. Dirk Beyer, Matthias Dangl, Daniel Dietsch, and Matthias Heizmann. Exchanging verification witnesses between verifiers. In J. Jürjens and K. Schneider, editors, *Tagungsband Software Engineering 2017, Fachtagung des GI-Fachbereichs Softwaretechnik (21.-24. Februar 2017, Hannover, Deutschland)*, LNI P-267, pages 93–94. Gesellschaft für Informatik (GI), 2017.
7. Dirk Beyer and Karlheinz Friedberger. A light-weight approach for verifying multi-threaded programs with cpachecker. In J. Bouda, L. Holík, J. Kofroň, J. Strejček, and A. Rambousek, editors, *Proceedings of the 11th Doctoral Workshop on Mathematical and Engineering Methods in Computer Science (MEMICS 2016, Telč, Czechia, October 21-23)*, EPTCS 233, pages 61–71. ArXiV, 2016.
8. Markus Schordan, Dirk Beyer, and Jonas Lundberg. Evaluation and reproducibility of program analysis and verification (track introduction). In T. Margaria and B. Steffen, editors, *Proceedings of the 7th International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2016, Corfu, Greece, October 10–14)*, LNCS 9952, pages 191–194. Springer-Verlag, Heidelberg, 2016.
9. Dirk Beyer and Matthias Dangl. SMT-based software model checking: An experimental comparison of four algorithms. In *Proc. VSTTE*, LNCS 9971, pages 181–198. Springer, 2016.
10. Egor George Karpenkov, Karlheinz Friedberger, and Dirk Beyer. JavaSMT: A unified interface for SMT solvers in Java. In *Proc. VSTTE*, LNCS 9971, pages 139–148. Springer, 2016.
11. Dirk Beyer, Matthias Dangl, Daniel Dietsch, Matthias Heizmann, and Andreas Stahlbauer. Verification witnesses. In J. Knoop and U. Zdun, editors, *Tagungsband Software Engineering 2016, Fachtagung des GI-Fachbereichs Softwaretechnik (23.-26. Februar 2016, Wien, Österreich)*, LNI 252, pages 105–106. Gesellschaft für Informatik (GI), 2016.
12. Malte Lochau, Johannes Bürdek, Stefan Bauregger, Andreas Holzer, Alexander von Rhein, Sven Apel, and Dirk Beyer. On facilitating reuse in multi-goal test-suite generation for software product lines. In J. Knoop and U. Zdun, editors, *Tagungsband Software Engineering 2016, Fachtagung des GI-Fachbereichs Softwaretechnik (23.-26. Februar 2016, Wien, Österreich)*, LNI 252, pages 81–82. Gesellschaft für Informatik (GI), 2016.
13. Dirk Beyer and Stefan Löwe. Interpolation for value analysis. In U. Aßmann, B. Demuth, T. Spitta, G. Püschel, and R. Kaiser, editors, *Tagungsband Software Engineering 2015, Fachtagung des GI-Fachbereichs Softwaretechnik (17. März - 20. März 2015, Dresden, Deutschland)*, LNI 239, pages 73–74. Gesellschaft für Informatik (GI), 2015.
14. Markus Schordan, Welf Löwe, and Dirk Beyer. Evaluation and reproducibility of program analysis (track introduction). In T. Margaria and B. Steffen, editors, *Proceedings of the 6th International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2014, Corfu, Greece, October 8-11)*, LNCS 8803, pages 479–481. Springer-Verlag, Heidelberg, 2014.

15. Dirk Beyer, Marieke Huisman, Vladimir Klebanov, and Rosemary Monahan. Evaluating software verification systems: Benchmarks and competitions (dagstuhl reports 14171). *Dagstuhl Reports*, 4(4):1–19, 2014.

16. Dirk Beyer, Andreas Holzer, Michael Tautschnig, and Helmut Veith. Reusing information in multi-goal reachability analyses. In W. Hasselbring and N. C. Ehmke, editors, *Tagungsband Software Engineering 2014, Fachtagung des GI-Fachbereichs Softwaretechnik (25. Februar - 28. Februar 2014, Kiel, Deutschland)*, LNI 227, pages 97–98. Gesellschaft für Informatik (GI), 2014.

17. Dirk Beyer, Stefan Löwe, Evgeny Novikov, Andreas Stahlbauer, and Philipp Wendler. Precision reuse in cpachecker. In W. Hasselbring and N. C. Ehmke, editors, *Tagungsband Software Engineering 2014, Fachtagung des GI-Fachbereichs Softwaretechnik (25. Februar - 28. Februar 2014, Kiel, Deutschland)*, LNI 227, pages 41–42. Gesellschaft für Informatik (GI), 2014.

18. Dirk Beyer and Andreas Stahlbauer. BDD-based software model checking with CPAchecker. In A. Kucera et al., editor, *Proceedings of the Annual Doctoral Workshop on Mathematical and Engineering Methods in Computer Science (MEMICS 2012, Znojmo, Czech Republic, October 26-28)*, LNCS 7721, pages 1–11. Springer-Verlag, Heidelberg, 2013.

19. Dirk Beyer and Alexander K. Petrenko. Linux driver verification. In T. Margaria and B. Steffen, editors, *Proceedings of the 5th International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2012, Part II, Heraklion, Crete, October 15-18)*, LNCS 7610, pages 1–6. Springer-Verlag, Heidelberg, 2012.

20. Falk Howar, Malte Isberner, Maik Merten, Bernhard Steffen, and Dirk Beyer. The RERS grey-box challenge 2012: Analysis of event-condition-action systems. In T. Margaria and B. Steffen, editors, *Proceedings of the 5th International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2012, Part I, Heraklion, Crete, October 15-18)*, LNCS 7609, pages 608–614. Springer-Verlag, Heidelberg, 2012.

21. Alain Wegmann, Lam-Son Le, Lotfi Hussami, and Dirk Beyer. A tool for verified design using Alloy for specification and CrocoPat for verification. In D. Jackson and P. Zave, editors, *Proceedings of the First Alloy Workshop (ALLOY 2006, Portland, OR, November 6)*, 2006.

22. Dirk Beyer. Co-change visualization applied to PostgreSQL and ArgoUML. In *Proceedings of the Third International Workshop on Mining Software Repositories (MSR 2006, Shanghai, May 22-23)*, pages 165–166. ACM Press, 2006.

23. Dirk Beyer, Arindam Chakrabarti, and Thomas A. Henzinger. An interface formalism for web services. In *Proceedings of the First International Workshop on Foundations of Interface Technologies (FIT 2005, San Francisco, CA, August 21)*, 2005.

24. Dirk Beyer. Rabbit: Verification of real-time systems. In P. Pettersson and S. Yovine, editors, *Proceedings of the Workshop on Real-Time Tools (RT-TOOLS 2001, Aalborg, August 20)*, pages 13–21, Uppsala, 2001.

25. Dirk Beyer and Andreas Noack. Efficient verification of timed automata using BDDs. In S. Gnesi and U. Ultes-Nitsche, editors, *Proceedings of the Sixth International ERCIM Workshop on Formal Methods for Industrial Critical Systems (FMICS 2001, Paris, July 16-17)*, pages 95–113. INRIA, Paris, 2001.

26. Dirk Beyer and Andy Heinig. Different strategies for BDD-based reachability analysis of timed automata. In C. Rattray, M. Sveda, and J. Rozenblit, editors, *Proceedings of the Second IEEE/IFIP Joint Workshop on Formal Specifications of Computer-Based Systems (FSCBS 2001, Washington, D.C., April 20)*, pages 89–98, Stirling, 2001.

27. Dirk Beyer and Heinrich Rust. Cottbus Timed Automata: Formal definition and semantics. In C. Rattray, M. Sveda, and J. Rozenblit, editors, *Proceedings of the Second IEEE/IFIP Joint Workshop on Formal Specifications of Computer-Based Systems (FSCBS 2001, Washington, D.C., April 20)*, pages 75–87, Stirling, 2001.

28. Dirk Beyer and Andreas Noack. BDD-basierte Verifikation von Realzeit-Systemen. In J. Grabowski and S. Heymer, editors, *Tagungsband Formale Beschreibungstechniken für verteilte Systeme (FBT 2000, Lübeck, June 22-23)*, pages 79–89. Shaker Verlag, Aachen, 2000.

29. Dirk Beyer and Heinrich Rust. Modular modelling and verification with Cottbus Timed Automata. In C. Rattray and M. Sveda, editors, *Proceedings of the IEEE/IFIP Joint Workshop on Formal Specifications of Computer-Based Systems (FSCBS 2000, Edinburgh, April 6-7)*, pages 17–24, Stirling, 2000.

30. Dirk Beyer, Claus Lewerentz, and Heinrich Rust. Modelling and analysing a railroad crossing in a modular way. In S. Gnesi, I. Schieferdecker, and A. Rennoch, editors, *Proceedings of the Fifth International ERCIM Workshop on Formal Methods for Industrial Critical Systems (FMICS 2000, Berlin, April 3-4)*, pages 287–303, Berlin, 2000.

31. Dirk Beyer and Heinrich Rust. Concepts of Cottbus Timed Automata. In K. Spies and B. Schätz, editors, *Tagungsband Formale Beschreibungstechniken für verteilte Systeme (FBT 1999, München, June 17-18)*, pages 27–34. Herbert Utz Verlag, München, 1999.
32. Dirk Beyer and Heinrich Rust. Modeling a production cell as a distributed real-time system with Cottbus Timed Automata. In H. König and P. Langendörfer, editors, *Tagungsband Formale Beschreibungstechniken für verteilte Systeme (FBT 1998, Cottbus, June 4-5)*, pages 148–159. Shaker Verlag, Aachen, 1998.

Technical Reports

1. Dirk Beyer, Nian-Ze Lee, and Philipp Wendler. Interpolation and sat-based model checking revisited: Adoption to software verification. Technical Report 2208.05046, arXiv/CoRR, August 2022.
2. Dirk Beyer, Lars Grunske, Thomas Lemberger, and Minxing Tang. Towards a benchmark set for program repair based on partial fixes. Technical Report 2107.08038, arXiv/CoRR, July 2021.
3. Dirk Beyer and Matthias Dangl. Software verification with pdr: Implementation and empirical evaluation of the state of the art. Technical Report 1908.06271, arXiv/CoRR, August 2019.
4. Dirk Beyer and Heike Wehrheim. Verification artifacts in cooperative verification: Survey and unifying component framework. Technical Report 1905.08505, arXiv/CoRR, May 2019.
5. Dirk Beyer, Matthias Dangl, and Philipp Wendler. Combining k-induction with continuously-refined invariants. Technical Report MIP-1503, Department of Computer Science and Mathematics (FIM), University of Passau (PA), January 2015.
6. Dirk Beyer, Stefan Löwe, and Philipp Wendler. Domain-type-guided refinement selection based on sliced path prefixes. Technical Report MIP-1501, Department of Computer Science and Mathematics (FIM), University of Passau (PA), January 2015.
7. Dirk Beyer, Stefan Löwe, Evgeny Novikov, Andreas Stahlbauer, and Philipp Wendler. Reusing precisions for efficient regression verification. Technical Report MIP-1302, Department of Computer Science and Mathematics (FIM), University of Passau (PA), May 2013.
8. Sven Apel, Dirk Beyer, Karlheinz Friedberger, Franco Raimondi, and Alexander von Rhein. Domain types: Selecting abstractions based on variable usage. Technical Report MIP-1303, Department of Computer Science and Mathematics (FIM), University of Passau (PA), May 2013.
9. Dirk Beyer and Stefan Löwe. Explicit-value analysis based on CEGAR and interpolation. Technical Report MIP-1205, Department of Computer Science and Mathematics (FIM), University of Passau (PA), December 2012.
10. Sven Apel, Hendrik Speidel, Philipp Wendler, Alexander von Rhein, and Dirk Beyer. Feature-aware verification. Technical Report MIP-1105, Department of Computer Science and Mathematics (FIM), University of Passau (PA), September 2011.
11. Dirk Beyer, Thomas A. Henzinger, M. Erkan Keremoglu, and Philipp Wendler. Conditional model checking. Technical Report MIP-1107, Department of Computer Science and Mathematics (FIM), University of Passau (PA), September 2011.
12. Dirk Beyer, Alessandro Cimatti, Alberto Griggio, M. Erkan Keremoglu, and Roberto Sebastiani. Software model checking via large-block encoding. Technical Report SFU-CS-2009-09, School of Computing Science (CMPT), Simon Fraser University (SFU), April 2009.
13. Dirk Beyer and M. Erkan Keremoglu. CPAchecker: A tool for configurable software verification. Technical Report SFU-CS-2009-02, School of Computing Science (CMPT), Simon Fraser University (SFU), January 2009.
14. Dirk Beyer, Arindam Chakrabarti, and Thomas A. Henzinger. An interface formalism for web services. Technical Report MTC-REPORT-2007-002, School of Computer and Communication Sciences (IC), Ecole Polytechnique Fédérale de Lausanne (EPFL), December 2007.
15. Dirk Beyer, Thomas A. Henzinger, Rupak Majumdar, and Andrey Rybalchenko. Path invariants. Technical Report MTC-REPORT-2006-003, School of Computer and Communication Sciences (IC), Ecole Polytechnique Fédérale de Lausanne (EPFL), December 2006.
16. Dirk Beyer, Thomas A. Henzinger, and Vasu Singh. Three algorithms for interface synthesis: A comparative study. Technical Report MTC-REPORT-2006-001, School of Computer and Communication Sciences (IC), Ecole Polytechnique Fédérale de Lausanne (EPFL), May 2006.
17. Dirk Beyer, Thomas A. Henzinger, and Grégory Théoduloz. Lazy shape analysis. Technical Report MTC-REPORT-2005-006, School of Computer and Communication Sciences (IC), Ecole Polytechnique Fédérale de Lausanne (EPFL), December 2005.

18. Dirk Beyer and Andreas Noack. Mining co-change clusters from version repositories. Technical Report IC/2005/003, School of Computer and Communication Sciences (IC), Ecole Polytechnique Fédérale de Lausanne (EPFL), January 2005.
19. Dirk Beyer and Andreas Noack. CrocoPat 2.1 Introduction and reference manual. Technical Report UCB//CSD-04-1338, Computer Science Division (EECS), University of California, Berkeley, July 2004. Also: The Computing Research Repository (CoRR), cs.PL/0409009, September 2004.
20. Dirk Beyer and Claus Lewerentz. CrocoPat: A tool for efficient pattern recognition in large object-oriented programs. Technical Report I-04/2003, Institute of Computer Science, Brandenburgische Technische Universität Cottbus, January 2003.
21. Dirk Beyer and Andreas Noack. A comparative study of decision diagrams for real-time verification. Technical Report I-03/2003, Institute of Computer Science, Brandenburgische Technische Universität Cottbus, January 2003.
22. Dirk Beyer. Rabbit: Verification of real-time systems. Technical Report I-05/2001, Institute of Computer Science, Brandenburgische Technische Universität Cottbus, March 2001.
23. Dirk Beyer. Reachability analysis and refinement checking for BDD-based model checking of timed automata. Technical Report I-04/2001, Institute of Computer Science, Brandenburgische Technische Universität Cottbus, February 2001.
24. Dirk Beyer and Andreas Noack. Efficient verification of real-time systems using BDDs. Technical Report I-13/2000, Institute of Computer Science, Brandenburgische Technische Universität Cottbus, December 2000.
25. Dirk Beyer, Claus Lewerentz, and Frank Simon. Flattening inheritance structures – or – Getting the right picture of large OO-systems. Technical Report I-12/2000, Institute of Computer Science, Brandenburgische Technische Universität Cottbus, November 2000.
26. Frank Simon and Dirk Beyer. Considering inheritance, overriding, overloading and polymorphism for measuring C++ sources. Technical Report I-04/2000, Institute of Computer Science, Brandenburgische Technische Universität Cottbus, May 2000.
27. Dirk Beyer and Heinrich Rust. A formalism for modular modelling of hybrid systems. Technical Report I-10/1999, Institute of Computer Science, Brandenburgische Technische Universität Cottbus, October 1999.
28. Dirk Beyer and Heinrich Rust. A modular hybrid modelling notation. Technical Report I-03/1999, Institute of Computer Science, Brandenburgische Technische Universität Cottbus, February 1999.

Guest Lectures and Invited Talks

1. *Configurable Software Model Checking: CPAchecker.*
   SEFM 2022 Summer School, Berlin, Germany, 2022-09-21.
2. *Cooperative Software Verification: Combination Approaches that Share Information.*
   Manchester University, Manchester, UK, 2022-08-16.
3. *Reliable Benchmarking: Requirements and Solutions.*
   Manchester University, Manchester, UK, 2022-08-16.
4. *Interpolation and SAT-Based Model Checking Revisited: Adoption to Software Verification.*
   Isaac Newton Institute for Mathematics, Program Verified Software, Cambridge, UK, 2022-07-22.
5. *Cooperative Software Verification Combination Approaches that Share Information.*
   Isaac Newton Institute for Mathematics, Program Verified Software, Cambridge, UK, 2022-07-22.
6. *Configurable Software Model Checking: CPAchecker.*
   Amazon Web Services, Dresden, Germany, 2022-06-08.
7. *Modular Model Checking (Frameworks).*
   Open Source Model Checker, Work Meeting, Virtual, 2022-03-29.
8. *CoVeriTeam: Cooperative Verification via Off-the-Shelf-Components.*
   ConVeY Research Graduate School, Raitenhaslach, Germany, 2021-10-22.
9. *PDR for Software.*
   TACAS Conference, Virtual, 2021-03-31.
10. *Competition on Software Verification.*
    ConVeY Graduate School, Garching, Munich, Germany, 2019-10-24.
11. *Reliable Benchmarking.*
    ConVeY Graduate School, Garching, Munich, Germany, 2019-10-24.

12. *Software Verification — An Overview of the State of the Art.*
    Invited Talk at the European Joint Conferences on Theory and Practice of Software 2019, Prague, Czechia, 2019-04-09.
13. *Correctness Witnesses.*
    Lorentz Workshop, Leiden, Netherlands, 2019-02-22.
14. *Reliable Benchmarking.*
    Lorentz Workshop, Leiden, Netherlands, 2019-02-21.
15. *Competition on Software Verification.*
    Lorentz Workshop, Leiden, Netherlands, 2019-02-18.
16. *Cooperative Verification.*
    Alumni Workshop, Cottbus, Germany, 2018-09-28.
17. *Reducer-Based Construction of Conditional Model Checkers.*
    CPAchecker Workshop 2018, 2018-09-25.
18. *Cooperative Verification: The Art of Combining Verification Tools.*
    fortiss (Forschungsinstitut des Freistaats Bayern für softwareintensive Systeme und Services), Munich, Germany, 2018-07-27.
19. *Cooperative Verification: The Art of Combining Verification Tools.*
    Universität Konstanz, Konstanz, Germany, 2018-07-19.
20. *Cooperative Verification: The Art of Combining Verification Tools.*
    Keynote at the 12th International Conference on Tests and Proofs, Toulouse, France, 2018-06-27.
21. *Cooperative Verification: The Art of Combining Verification Tools.*
    Lectures at Eighth Summer School on Formal Techniques, Menlo College, Atherton (CA), USA, 2018-05-23 – 2018-05-24.
22. *Forschungswettbewerb als Motor des Technologie-Transfers.*
    Invited Talk at Dagstuhl Seminar, Dagstuhl, Germany, 2018-04-11.
23. *Verification with Reusing Exchangeable Results.*
    Lectures at RiSE/SHiNE Winter School, TU Wien, Austria, 2018-02-08 – 2018-02-09.
24. *Configurable Software Verification.*
    Lectures at RiSE/SHiNE Winter School, TU Wien, Austria, 2018-02-08 – 2018-02-09.
25. *Best Practices On Artifact Integration.*
    ACM Task Force Workshop on Reproducibility and Artifacts, ACM Headquarter, New York, NY, USA, 2017-12-07.
26. *Getting Software Verifiers Ready for Industrial Use.*
    Lectures at SOAMED Project Meeting, Zeuthen, Germany, 2017-10-05.
27. *Software Verification and Verifiable Witnesses.*
    Johannes Kepler Universität Linz, Linz, Austria, 2015-03-12.
28. *CPAchecker: A Flexible Framework for Software Verification.*
    Siemens Nürnberg, Nürnberg, Germany, 2015-02-16.
29. *Conditional Model Checking.*
    ISoLA 2014, Corfu, Greece, 2014-10-11.
30. *Stateful Verificaion.*
    University of Freiburg, Freiburg, Germany, 2014-03-27.
31. *CPAchecker: A Flexible Framework for Software Verification.*
    Dagstuhl Seminar 14352, Schloss Dagstuhl, Germany, 2014-08-25.
32. *Competition on Software Verification.*
    Dagstuhl Seminar 14171, Schloss Dagstuhl, Germany, 2014-04-24.
33. *Automatic Software Verification.*
    Dagstuhl Seminar 14171, Schloss Dagstuhl, Germany, 2014-04-23.
34. *Stateful Verification.*
    IST Austria, Klosterneuburg, Austria, 2013-11-06.
35. *Reuse of Verification Results: Conditional Model Checking, Precision Reuse, and Verification Witnesses.*
    SPIN 2013, Stony Brook, NY, 2013-07-09.
36. *Competition on Software Verification – An Overview.*
    Dagstuhl Seminar, Schloss Dagstuhl, 2012-11-15.

37. *Conditional Model Checking: A Technique to Pass Information between Verifiers.*
Dagstuhl Seminar, Schloss Dagstuhl, 2012-11-12.

38. *CPAchecker: The Configurable Software-Verification Platform.*
MEMICS Workshop, Znojmo, 2012-10-26.

39. *Conditional Model Checking.*
University of Paderborn, Paderborn, 2011-10-25.

40. *Zuverlässige Softwaresysteme.*
Alumni Day at BTU Cottbus, Cottbus, 2011-06-17.

41. *Towards a Unified Framework for Software Verification.*
Lecture at Graduate College, TU Munich, Garching, 2011-02-04.

42. *Towards a Unified Framework for Software Verification.*
Alpine Verification Meeting, Lugano, 2010-10-19.

43. *Adjustable-Block Encoding.*
Oxford University, 2010-10-14.

44. *Protocol Interfaces.*
Workshop on Foundations of Interface Technologies, Paris, 2010-08-30.

45. *Adjustable-Block Encoding — Towards a Unified Framework for Software Verification.*
Computer Science Symposium at IST Austria, Klosterneuburg, 2010-05-07.

46. *Program Analysis with Dynamic Change of Precision.*
University of California, Berkeley (CA), 2009-04-16.

47. *Panelist on Talent, University-Industry Cooperation, and Curriculum Development.*
Pacific Northwest Wireless Summit 2009 (PNWS'09), Vancouver, 2009-01-19.

48. *Datenfluss-Analyse mit dynamischer Anpassung der Genauigkeit.*
Brandenburg University of Technology, Cottbus, 2008-11-07.

49. *Predicate Abstraction with Summarization.*
TRESOR Seminar at EPFL, Lausanne, 2008-11-05.

50. *Building Software-Engineering Tools in Academia.*
Second International Workshop on Advanced Software Development Tools and Techniques (WASDeTT'08), Workshop at ICSM'09, Beijing, 2008-10-03.

51. *Struktur-Analyse und Verifikation Großer Software-Systeme*
University of Passau, Passau, 2008-07-21.

52. *Teaching Software Engineering on Mobile Devices.*
NOKIA University Relations Forum, NOKIA, Burnaby, 2008-06-12.

53. *The Software Model Checker* BLAST.
Guest Lecture in Viktor Kuncak's Verification Course, EPFL-IC-LARA, Lausanne, 2008-05-08.

54. *Structure Analysis of Large Software Systems.*
University of Victoria, Victoria, 2007-09-21.

55. *Path Invariants.*
University of British Columbia, Vancouver, 2007-06-20.

56. *Web Service Interfaces.*
Workshop on Constraints for Composing Web Services, LORIA, Nancy, 2006-06-27.

57. *Combining Model Checking and Shape Analysis.*
Dagstuhl Seminar 06081 "Software Verification", 2006-02-20.

58. *Structure Analysis of Large Software Systems.*
Oxford University, 2006-02-14.

59. *Formal and Semi-Formal Methods in Software Engineering.*
IT University, Göteborg, 2005-12-15.

60. *Combining Data Flow Analysis with Lazy Abstraction Refinement in* BLAST.
Politecnico di Milano, 2005-12-02.

61. *Formale Verifikation von Realzeit-Systemen mittels Cottbus Timed Automata.*
Fraunhofer-Gesellschaft FIRST, Berlin, 2003-04-11.

62. *Efficient BDD Representation for Reachability Analysis of Timed Automata.*
Carnegie Mellon University, Pittsburgh (PA), 2003-01-14.

63. *A Modular Approach for Formal Verification of Real-Time Systems.*
Software Engineering Institute, Pittsburgh (PA), 2003-01-10.

64. *Improvements in BDD-based Reachability Analysis of Timed Automata.*
Naval Research Laboratory, Washington (D.C.), 2001-04-23.
65. *Entwurfsmuster: Eine Einführung.*
PC-Soft GmbH, Senftenberg, 1997-12-16.

Conference and Other Presentations

1. *Cooperative Software Verification.*
Alpine Verification Meeting 2022, Chiemsee, Germany, 2022-09-12.
2. *An Interface Theory for Program Verification.*
ISOLA Conference, Rhodos, Greece, 2021-10-29.
3. *Verification Artifacts in Cooperative Verification: Survey and Unifying Component Framework.*
ISOLA Conference, Rhodos, Greece, 2021-10-29.
4. *Violation Witnesses and Result Validation for Multi-threaded Programs: Implementation and Evaluation with CPAchecker.*
ISOLA Conference, Rhodos, Greece, 2021-10-25.
5. *Software Verification: Historical Landmarks and Current Developments.*
6th International Workshop on CPAchecker (CPA 2021), Virtual, 2021-10-01.
6. *CPU-Energy-Meter.*
TACAS Conference, Virtual, 2021-03-31.
7. *CPU-Energy-Meter.*
VMCAI Winter School, New Orleans, USA, 2020-01-16 – 2020-01-18.
8. *Cooperative Verification.*
CPAchecker Workshop 2019, Chiemsee, Germany, 2019-10-01.
9. *Verification Witnesses — A Data Show Case.*
MSR 2019, Montreal, Canada, 2019-05-25.
10. *Test-Comp at TOOLympics.*
TOOLympics 2019 at ETAPS, Prague, Czechia, 2019-04-07.
11. *SV-COMP at TOOLympics.*
TOOLympics 2019 at ETAPS, Prague, Czechia, 2019-04-07.
12. *1st Competition on Software Testing.*
Test-Comp 2019, Prague, Czechia, 2019-04-06.
13. *Competition on Software Verification 2019.*
SV-COMP 2019, Prague, Czechia, 2019-04-06.
14. *Applying CPAchecker to Large Explicit State Spaces.*
RERS at ISOLA Conference, 2018-11-08.
15. *In-Place vs. Copy-on-Write CEGAR Refinement for Block Summarization with Caching.*
ISOLA Conference, 2018-11-06.
16. *Strategy Selection.*
ISOLA Conference, 2018-11-05.
17. *Software Verification with Validation of Results.* TACAS Conference, SV-COMP Session, Uppsala, Sweden, 2016-04-27.
18. *Partial Verification and Intermediate Results as a Solution to Combine Automatic and Interactive Verification Techniques.* ISOLA Conference, 2016-10.
19. *Symbolic Execution with CEGAR.* ISOLA Conference, 2016-10-10.
20. *Software Verification and Verifiable Witnesses.*
21st International Conference on Tools and Algorithms for the Construction and of Analysis Systems TACAS 2015, London, UK, 2015-04-16.
21. *Status Report on Software Verification.*
20th International Conference on Tools and Algorithms for the Construction and of Analysis Systems TACAS 2014, Grenoble, France, 2014-04-10.
22. *Second Competition on Software Verification.*
19th International Conference on Tools and Algorithms for the Construction and of Analysis Systems TACAS 2013, Rome, 2013-03-21.

23. *Competition on Software Verification.*
18th International Conference on Tools and Algorithms for the Construction and of Analysis Systems
TACAS 2012, Tallinn, 2012-03-29.

24. *CPAchecker: A Tool for Configurable Software Verification.*
23rd International Conference on Computer Aided Verification
CAV 2011, Snowbird (UT), 2011-07-19.

25. *CPAchecker: A Tool for Configurable Software Verification.*
15th Biennial Workshop on Programmiersprachen und Grundlagen der Programmierung
KPS 2009, Maria Taferl, 2009-10-14.

26. *Evolution Storyboards: Visualization of Software Structure Dynamics.*
14th International Conference on Program Comprehension
ICPC 2006, Athens, 2006-06-16.

27. *Co-change Visualization Applied to PostgreSQL and ArgoUML.*
3rd International Workshop on Mining Software Repositories
MSR 2006, Shanghai, 2006-05-23.

28. *Co-Change Visualization.*
21st IEEE International Conference on Software Maintenance
ICSM 2005, Budapest, 2005-09-26.

29. *Clustering Software Artifacts Based on Frequent Common Changes.*
13th IEEE International Workshop on Program Comprehension
IWPC 2005, St. Louis, 2005-05-16.

30. *CrocoPat: An Efficient Calculator for Relational Programs.*
TRESOR seminar, EPFL, Lausanne, 2005-04-14.

31. *An Introduction to Binary Decision Diagrams.*
CAV lecture, EPFL, Lausanne, 2004-11-18.

32. *An Eclipse Plug-in for Model Checking.*
12th IEEE International Workshop on Program Comprehension
IWPC 2004, Bari, 2004-06-26.

33. *Generating Tests from Counterexamples.*
26th International Conference on Software Engineering
ICSE 2004, Edinburgh, 2004-05-27.

34. *Generating Tests from Counterexamples.*
EPFL, Lausanne, 2004-05-18.

35. *Simple and Efficient Relational Querying.*
OSQ seminar, University of California, Berkeley (CA), 2004-02-02.

36. *How to Make Model Checking of Timed Automata Efficient.*
University of California, Berkeley (CA), 2003-10-29.

37. *Rabbit: A Tool for BDD-based Verification of Real-Time Systems.*
15th International Conference on Computer Aided Verification
CAV 2003, Boulder (CO), 2003-07-09.

38. *CrocoPat: Efficient Pattern Analysis in Object-Oriented Programs.*
11th IEEE International Workshop on Program Comprehension
IWPC 2003, Portland (OR), 2003-05-11.

39. *Formale Verifikation von Realzeit-Systemen mittels Cottbus Timed Automata.*
Dissertation, Cottbus, 2002-11-26.

40. *Efficient Reachability Analysis and Refinement Checking of Timed Automata using BDDs.*
11th IFIP Working Conference on Correct Hardware Design and Verification Methods
CHARME 2001, Livingston, 2001-09-04.

41. *Verification of Real-Time Systems.*
Workshop on Real-Time Tools
RT-TOOLS 2001, Aalborg, 2001-08-20.

42. *Efficient Verification of Timed Automata using BDDs.*
6th International ERCIM Workshop on Formal Methods for Industrial Critical Systems
FMICS 2001, Paris, 2001-07-16.

43. *Different Strategies for BDD-based Reachability Analysis of Timed Automata.*
2nd IEEE/IFIP Joint Workshop on Formal Specifications of Computer-Based Systems
FSCBS 2001, Washington (D.C.), 2001-04-20.

44. *Cottbus Timed Automata: Formal Definition and Semantics.*
    2nd IEEE/IFIP Joint Workshop on Formal Specifications of Computer-Based Systems
    FSCBS 2001, Washington (D.C.), 2001-04-20.
45. *Improvements in BDD-based Reachability Analysis of Timed Automata.*
    10th International Symposium of Formal Methods Europe
    FME 2001, Berlin, 2001-03-15.
46. *A Tool for Modular Modelling and Verification of Hybrid Systems.*
    25th IFAC/IFIP Workshop on Real-Time Programming
    WRTP 2000, Palma, 2000-05-19.
47. *Modular Modelling and Verification with Cottbus Timed Automata.*
    IEEE/IFIP Joint Workshop on Formal Specifications of Computer-Based Systems
    FSCBS 2000, Edinburgh, 2000-04-06.
48. *Modelling and Analysing a Railroad Crossing in a Modular Way.*
    5th International ERCIM Workshop on Formal Methods for Industrial Critical Systems
    FMICS 2000, Berlin, 2000-04-04.
49. *Concepts of Cottbus Timed Automata.*
    Workshop Formale Beschreibungstechniken für verteilte Systeme
    FBT 1999, München, 1999-06-17.
50. *Ein Analysewerkzeug für zeitbehaftete Automaten.*
    Diplomarbeit, Cottbus, 1998-06-30.
51. *Modeling a Production Cell as a Distributed Real-Time System with Cottbus Timed Automata.*
    Workshop Formale Beschreibungstechniken für verteilte Systeme
    FBT 1998, Cottbus, 1998-06-04.

University Activities

Instruction at Ludwig-Maximilians-Universität München

Undergraduate courses:

Software Engineering
Winter 2016/17, Winter 2017/18, Winter 2018/19, Winter 2020/21, Winter 2021/22

Formal Specification and Verification
Summer 2017, Summer 2018

Formale Sprachen und Komplexität
Summer 2020

Theoretische Informatik für Medieninformatiker
Summer 2020

Graduate courses:

Semantics for Programming Languages
Winter 2019/20, Winter 2020/21, Winter 2021/22

Testing
Summer 2018, Summer 2019, Summer 2020, Summer 2021, Summer 2022

Software Analysis and Verification
Winter 2020/21, Winter 2021/22

Software Verification
Winter 2018/19, Winter 2019/20

Formal Specification and Verification II
Winter 2017/18

Science and Practice in Software Engineering
Summer 2021, Summer 2022

Graduate seminars:

Seminars on Software Engineering, Software Analysis, Verification, and Testing

Instruction at University of Passau

Undergraduate courses:

Foundations of Computer Science, 5100
Winter 2010/11

Algorithms and Data Structures, 5200
Summer 2010

Software Engineering, 5300
Winter 2009/10, Winter 2012/13, Winter 2013/14, Winter 2014/15

Theoretische Informatik I, 5306
Winter 2015/16

Theoretische Informatik II, 5308
Winter 2015/16

Spezifikation und Verifikation von Eingebetteten Systemen, 5463
Winter 2014/15, Winter 2015/16

Software Engineering Praktikum, 5500
Summer 2014

Graduate courses:

Software Analysis, 5840
Winter 2009/10, Winter 2011/12

Software Verification, 5843
Winter 2012/13, Winter 2013/14, Winter 2014/15, Winter 2015/16

Object-Oriented Programming, 5620
Summer 2010

Principles of Compiler Design, 5790
Summer 2012, Summer 2013, Summer 2014

Academic and Scientific Methods, 5844
Summer 2012, Summer 2014

Graduate seminars:

Seminar Seminar Entwurf und Analyse von Softwaresystemen, 5846/5847
Winter 2009/10, Winter 2010/11, Winter 2011/12, Summer 2012, Winter 2013/14, Summer 2014, Winter 2014/15, Winter 2015/16

Seminar Software Testing and Analysis, 5848
Summer 2010, Winter 2012/13

Events for High-School Students:

Sommercamp Informatik, 5900
Summer 2010, Summer 2011, Summer 2012, Summer 2013, Summer 2014, Summer 2015

Instruction at SFU

Undergraduate courses:

Principles of Compiler Design, CMPT 379
Spring 2008, Spring 2009

Software Engineering II, CMPT 475
Spring 2007, Spring 2008, Fall 2008

Graduate courses:

Software Engineering, CMPT 745
Fall 2007, Fall 2008

Special Topics in CS - Program Analysis, CMPT 880
Spring 2007

Graduate seminars:

Software Verification, CMPT 894
Spring 2008, Spring 2009

Teaching Assistant at EPFL and BTU

Undergraduate courses:

Theoretical Computer Science, Prof. Henzinger
Winter 2005/06

Introduction to Computer Science, Prof. Bachmann
Winter 2002/03

Introduction to Data Structures and Efficient Algorithms, Prof. Heiner
Winter 2000/01, Winter 1998/99

Introduction to Software Engineering, Prof. Lewerentz
Summer 2000, Summer 1999

Introduction to Software Engineering (for teachers from high schools), Prof. Lewerentz
Summer 2000

Introduction to Software Engineering (for engineers from industry), Prof. Lewerentz
Summer 2002, Summer 1999

Introduction to Software Engineering (for students of engineering), Prof. Lewerentz
Summer 2001

Computer Science (for students of environmental and resource management), Dr. Rust
Winter 2000/01, Winter 1999/2000

Software Project, Prof. Lewerentz
Winter 2002/03, Summer 2002, Winter 2001/02, Summer 2001, Winter 2000/01, Summer 1999

Graduate courses:

Computer-Aided Verification, Prof. Henzinger
Winter 2004/05

Software Engineering I, Prof. Lewerentz
Winter 2001/02

Software Engineering II, Prof. Lewerentz
Sommer 2003, Summer 2002, Summer 2001

Software Project Management, Prof. Lewerentz
Winter 2001/02

Graduate seminars:

Thrust in Reliable Software (TRESOR)
Winter 2004/05, Summer 2005, Winter 2005/06

Complexity and Software
Winter 2002/03

Advising

Current postdocs:

Nian-Ze Lee;

Philipp Wendler;

Stefan Winter;

Current students:

Marvin Brieger, PhD program;

Thomas Bunk, PhD program;

Po-Chun Chien, PhD program;

Sudeep Kanav, PhD program;

Matthias Kettl, PhD program;

Martin Spießl, PhD program;

Henrik Wachowitz, PhD program;

PhD thesis supervisor:

Matthias Dangl. Witness-based validation of verification results with applications to software-model checking. LMU Munich, 2022.

Thomas Lemberger. Towards cooperative software verification with test generation and formal verification. LMU Munich, 2022.

Karlheinz Friedberger. Efficient software model checking with block-abstraction memoization. LMU Munich, 2022.

Sabine Bauer. Decidability of linear tree constraints for resource analysis of object-oriented programs. LMU Munich, 2019.

Philipp Wendler. Towards practical predicate analysis. University of Passau, 2017.

Stefan Löwe. Effective approaches to abstraction refinement for automatic software verification. University of Passau, 2017.

Mehmet Erkan Keremoglu. Towards scalable software analyisis using combinations and conditions with CPAchecker. Simon Fraser University, 2011.

Grégory Théoduloz. Software verification by combining program analyses of adjustable precision. EPFL, MTC Lab, supervised by Prof. Thomas Henzinger, 2010.

Master's thesis supervisor:

Daniel Baier. Implementation of value analysis over symbolic memory graphs in cpachecker. LMU Munich, 2022.

Martin Pletl. A new spin on verification with symbolic execution: Symbolic execution as formula-based predicate analysis in cpachecker. LMU Munich, 2022.

Maximilian Hailer. New approaches and visualization for verification coverage. LMU Munich, 2022.

Matthias Kettl. Adjustable block analysis: Actor-based creation of block summaries for scaling formal verification. LMU Munich, 2022.

Philipp Waldinger. Concurrent software verification through block-based task partitioning and continuous summary refinement. LMU Munich, 2022.

Moritz Beck. Solver-based analysis of memory safety using separation logic. LMU Munich, 2020.

Alexander Koos. Implementation and evaluation of a framework for canonization and caching of SMT formulae. LMU Munich, 2019.

Stephan Holzner. Design und Implementierung einer parallelen BDD-Bibliothek. LMU Munich, 2019.

Michael Maier. SMT-based verification of ECMAScript programs in CPAchecker. LMU Munich, 2019.

Mirjam Trapp. Heuristics for effective predicate refinement in CPAchecker. LMU Munich, 2019.

Thomas Bunk. LTL software model checking in CPAchecker. LMU Munich, 2019.

Johannes Knaut. Symbolic heap abstraction with automatic refinement. LMU Munich, 2018.

Martin Spiessl. Configurable software verification based on slicing abstractions. LMU Munich, 2018.

Thomas Lemberger. Abstraction refinement for model checking: Program slicing + CEGAR. LMU Munich, 2018.

Thomas Stieglmaier. Augmenting predicate analysis with auxiliary invariants. University of Passau, 2016.

Sebastian Ott. Implementing a termination analysis using configurable software analysis. University of Passau, 2016.

Karlheinz Friedberger. Block-abstraction memoization as an approach to verify recursive procedures. University of Passau, 2015.

Matthias Dangl. Light-weight invariant generation for software verification with CPAchecker. University of Passau, 2013.

Christopher Jahn. Implementation of a CFA and ARG visualization and navigation tool in Java. University of Passau, 2012.

Andreas Stahlbauer. Block-encoding strategies for predicate analysis: An experimental study. University of Passau, 2012.

Peter Häring. A comparative study of software measures as problem-predictors. University of Passau, 2012.

Andra-Maria Babau. Modeling and verification of airport security processes using BPMN and protocol interfaces: A case study. University of Passau, 2011.

Dmitry Balzer. Werkzeugunterstützung für Verstehen und Monitoring von Software-Abhängigkeiten. University of Passau, 2010.

Alexander von Rhein. Verification tasks for software model checking. University of Passau, 2010.

Ashgan Fararooy. Performing static structure analysis using software dependencies. Simon Fraser University, 2010.

Philipp Wendler. Software verification based on adjustable large-block encoding. University of Passau, 2010.

Damien Zufferey. EPFL, MTC Lab, with Prof. Thomas Henzinger, 2009.

Grégory Théoduloz. Integrating shape analysis into the model checker BLAST. EPFL, MTC Lab, with Prof. Thomas Henzinger, 2006.

Andreas Noack. BDD-basierte Verifikation von Echtzeitsystemen. BTU Cottbus, with Prof. Claus Lewerentz, 2000.

Bachelor's thesis supervisor:

Moritz Bierwirth. Designing and assessing a benchmark set for fault localization using fault injection. LMU Munich, 2023.

Klara Cimbalnik. Program transformation in CPACHECKER: Design and implementation of a source-respecting translation from control-flow automata to c code. LMU Munich, 2022.

Tobias Kleinert. Developing a verifier based on parallel portfolio with COVERITEAM. LMU Munich, 2022.

Robin Gloster. Cgroups v2 support for BENCHEXEC. LMU Munich, 2022.

Ludwig Glückstadt. Genetic programming in software verification. LMU Munich, 2021.

Simon Antonischki. A cpa for string analysis for java programs in CPACHECKER. LMU Munich, 2021.

Penelope Powers. Mutation based automatic program repair in CPACHECKER. LMU Munich, 2021.

Korab Zogu. Sv-comp benchmarks for weak memory models. LMU Munich, 2021.

Yun Zhang. Verification witnesses: from llvm to c. LMU Munich, 2021.

Simon Raths. Implementation and evaluation of tbdds in pjbdd. LMU Munich, 2021.

Sebastian Tschoepel. Implementation and evaluation of a simple taint analysis for CPACHECKER. LMU Munich, 2021.

Dennis Simon. Shareable benchmarking reports with enhanced filters and dynamic statistics for BENCHEXEC. LMU Munich, 2021.

Martin Zehendner. Software verification with numerical domains in CPACHECKER. LMU Munich, 2020.

Sven Umbricht. Converting between acsl annotations and witness invariants. LMU Munich, 2020.

Benedikt Damböck. Implementierung und evaluation von einfacher schleifenabstraktion für das CPACHECKER-framework. LMU Munich, 2020.

Sven Massard. Improve analysis of java programs in CPACHECKER. LMU Munich, 2020.

Frederic Schönberger. Converting test goals to condition automata. LMU Munich, 2020.

Jakob Selberg. Automatic generation of test harnesses for pointer-based c programs: Implementation of a pointer-tracking analysis and harness-generation engine in the formal verification framework CPACHECKER. LMU Munich, 2020.

Yannick Adams. Domain types for predicate analysis in CPACHECKER. LMU Munich, 2020.

Vladyslav Kolesnykov. SMT-based model checking of concurrent programs. LMU Munich, 2020.

Radu-Cristian Rusanu. Interval-based optimization for smt solvers. LMU Munich, 2020.

Amena Abdulla. Reale anforderungen für die software-analyse. LMU Munich, 2020.

Simon Lund. Code complexity measures in software engineering: A systematic comparison and evaluation on software-component level. LMU Munich, 2020.

Angelos Kafounis. Fault localization in model checking. implementation and evaluation of fault-localization techniques with distance metrics. LMU Munich, 2020.

Schindar Ali. Test-based fault localization in the context of formal verification: Implementation and evaluation of the tarantula algorithm in cpachecker. LMU Munich, 2020.

Petros Isaakidis. Energy consumption prediction of verification work. LMU Munich, 2020.

Matthias Kettl. Fault localization for formal verification: An implementation and evaluation of algorithms based on error invariants and unsat-cores. LMU Munich, 2020.

Sonja Münchow. A web frontend for visualization of computation steps and their results in cpachecker. LMU Munich, 2020.

Adrian Leimeister. A language server and ide plugin for CPAchecker. LMU Munich, 2020.

Michael Obermeier. Extending the framework JavaSMT with the SMT solver Yices2. LMU Munich, 2020.

Alexander Ried. Design and implementation of a cluster-based approach for software verification. LMU Munich, 2020.

Maximilian Hailer. Measuring and optimizing energy consumption of verification work on clusters. LMU Munich, 2019.

Daniel Baier. Integration des SMT-Solvers Boolector in das Framework JavaSMT und Evaluation mit CPAchecker. LMU Munich, 2019.

Laura Bschor. Modern architecture and improved UI for tables of BenchExec. LMU Munich, 2019.

Maximilian Wiesholler. Correctness witness validation using predicate analysis. LMU Munich, 2019.

Krutav Shah. Counterexample-guided abstraction refinement for interval domain. LMU Munich, 2019.

Raphael Hagl. Hybrid testcase generation with CPAchecker. LMU Munich, 2019.

Andrea Kreppel. Implementation and evaluation of backwards analyses in the software-verification framework CPAchecker. LMU Munich, 2019.

Matthias Gerlach. Newton refinement as alternative to Craig interpolation in CPAchecker. LMU Munich, 2018.

Flutura Estler. Heuristics-based selection of verification configurations. LMU Munich, 2018.

Balthasar Schuess. Flexible online job scheduling in a multi-user environment. LMU Munich, 2018.

Dominik Friedrich. Konzeption, Umsetzung und Visualisierung von statistischen Daten in CPAchecker. LMU Munich, 2018.

Moritz Buhl. Application of software verification to OpenBSD network modules. LMU Munich, 2018.

Nicholas Reyes. Integrating a witness store into a distributed verification system. LMU Munich, 2018.

Dominik Pastau. Implementation of a generic cloud-based file-storage solution and its integration into a web-based distributed verification system. LMU Munich, 2018.

Karam Shabita. String analysis for Java programs in CPAchecker. LMU Munich, 2018.

Evgeny Dunaev. Entwurf und Implementierung einer Abstraktionsschicht für Zuweisungs-basierte Analysen. LMU Munich, 2017.

Deyan Ivanov. Interactive visualization of verification results from CPAchecker with D3. LMU Munich, 2017.

Nils Steinger. Measuring, visualizing, and optimizing the energy consumption of computer clusters. University of Passau, 2017.

Gernot Zoerneck. Implementing PDR in CPAchecker. University of Passau, 2017.

Stefan Weinzierl. Configurable pointer-alias analysis in CPAchecker. University of Passau, 2016.

Maximilian Syri. Verification of concurrent programs by CFA sequentialization. University of Passau, 2016.

Stephan Lukasczyk. Unbounded heap support for CPAchecker's predicate analysis using SMT arrays. University of Passau, 2016.

Magdalena Murr. Towards understandable CPAchecker counterexamples. University of Passau, 2016.

Thomas Lemberger. Efficient symbolic execution using CEGAR over two abstract domains. University of Passau, 2015.

Sebastian Ott. VerifierCloud: Implementierung eines web-service zur software-verifikation. University of Passau, 2014.

Thomas Stieglmaier. Octagon-based software verification with CPAchecker. University of Passau, 2014.

Georg Dresler. A google-app-engine implementation for CPAchecker. University of Passau, 2014.

Matthias Dittrich. Bit-precise predicate analysis with CPAchecker. University of Passau, 2013.

Alexander Driemeyer. Software-verifikation von java-programmen in cpachecker. University of Passau, 2012.

Karlheinz Friedberger. Ein typbasierter Ansatz zur Kombination verschiedener Verifikationstechniken. University of Passau, 2012.

Internship students:

Emanuele De Angelis (from University of Chieti-Pescara), Uni Passau, 2013;

Przemyslaw Daca (from TU Denmark), Uni Passau, 2011;

Philipp Wendler (from Uni Passau), SFU, 2009;

Michael Tautschnig (from TU Darmstadt), SFU, 2008;

Andreas Holzer (from TU Darmstadt), SFU, 2008;

Alberto Griggio (from Uni Trento), SFU, 2008;

Damien Zufferey (from EPFL), SFU, 2007;

Sudhanshu Narang (from IIT Delhi), SFU, 2007;

Rajhans Samdani (from IIT Bombay), EPFL, 2006;

Nitesh Kumar (from IIT Kanpur), EPFL, 2005

PhD thesis referee:

Nian-Ze Lee; National Taiwan University, 2021

Marie-Christine Jakobs, University of Paderborn, 2016

Daniel Dietsch, University of Freiburg, 2016

George Karpenkov, University of Grenoble, 2016

Evren Ermis, University of Freiburg, 2014

Jiri Slaby, Masaryk University, 2014

Andreas Holzer, TU Vienna, 2013

PhD thesis defense chair:

Jan Seedorf, Uni Passau, 2013

Roozbeh Farahbod, SFU, 2009

PhD depth examination chair:

Brian Fraser, SFU, 2007

MSc thesis referee:

Siegfried Rasthofer, Uni Passau, 2013

Stephan Huber, Uni Passau, 2012

Hendrik Speidel, Uni Passau, 2011

Kathrin Hanauer, Uni Passau, 2010

George Ma, SFU, 2007

MSc thesis defence examiner:

Wolfgang Haas, SFU, 2007;

George Ma, SFU, 2007

MSc thesis defense chair:

Kaiyan Jin, SFU, 2009;

Edward Glen, SFU, 2007;

Majid Bagheri, SFU, 2007;

Chiyoko Kawano, SFU, 2006

Departmental and University Committees

Chair Prüfungsausschuss Informatik IfI, LMU Munich, since 2019

Chair of the Department of Computer Science, LMU Munich, 2020–2022

Chair Berufungskommission W3 “Programmierung and AI”, LMU Munich, 2021–2022

Chair Berufungskommission W3 “Theoretische Informatik (2)”, LMU Munich, 2021–2022

Berufungskommission W3 “Technology-Enhanced Learning”, LMU Munich, 2019

Berufungskommission W1 “Medieninformatik”, LMU Munich, 2019

Chair Berufungskommission W3 “Theoretische Informatik”, LMU Munich, 2018–2019

Berufungskommission W1 “Software Engineering”, LMU Munich, 2018–2019

Prüfungsausschuss IfI, LMU Munich, 2016–2019

Promotionsausschuss FIM, Uni Passau, 2015–2017

Faculty Council, Uni Passau, 2013–2015

Studiengangsverantwortlicher MSc Informatik, Uni Passau, 2015–2016

Studienberater Lehramt Informatik, Uni Passau, 2011–2016

Promotionsausschuss FIM, Uni Passau, 2013–2015

Berufungskommission W3 “Theoretische Informatik”, Uni Passau, 2014–2015

Berufungskommission W3 “Betriebliche Informationssysteme”, Uni Passau, 2014

Chair Berufungskommission W3 “Complex-Systems Engineering”, Uni Passau, 2012

Berufungskommission W3 “Embedded Systems”, Uni Passau, 2012

Promotionsausschuss FIM, Uni Passau, 2010–2013

Berufungskommission W2 “Medieninformatik”, Uni Passau, 2011

Berufungskommission W3 “Bildverarbeitung”, Uni Passau, 2009–2011

Hardware and Capital Resources Committee, SFU, 2008–2009

Faculty Recruiting Committee, SFU, 2007–2008

Faculty Council, BTU Cottbus, 1998–2000

President of the CS Students’ Organization, BTU Cottbus, 1997–1998

Professional Activities

Conference Organizer

1. General Co-Chair, 25st European Joint Conferences on Theory and Practice of Software (ETAPS), Munich, Germany, 2022
2. Workshops Chair, 25st European Joint Conferences on Theory and Practice of Software (ETAPS), Munich, Germany, 2022
3. Co-Organizer, 14th Alpine Verification Meeting (AVM),
   Frauenwörth, Germany, 2022
4. Program Co-Chair, 21st Int. Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI),
   New Orleans, USA, January 19–21, 2020
5. Program Co-Chair, 13th Int. Conference on Tests and Proofs (TAP),
   Porto, Portugal, October 9–11, 2019
6. Co-Organizer, 4th Int. Workshop on CPAchecker (CPA),
   Frauenchiemsee, Germany, October 1–2, 2019
7. Program Co-Chair, 24th Int. Conference on Tools and Algorithms for the Construction and of Analysis Systems (TACAS),
   Thessaloniki, Greece, 2018
8. Co-Organizer, 1st Int. Workshop on CPAchecker (CPA),
   San Francisco (CA), USA, July 18–24, 2015
9. Workshops Chair, 27th International Conference on Computer Aided Verification (CAV),
   Munich, Germany, 2022

10. Co-Organizer, Dagstuhl Seminar 14171: Evaluating Software Verification Systems: Benchmarks and Competitions,
    Schloss Dagstuhl, April 21–25, 2014
11. Program Co-Chair, 32nd IFIP Int. Conference on Formal Techniques for Distributed Systems (FORTE/FMOODS),
    Florence, Italy, June, 2013
12. Organizer, 2st TACAS 2013 Competition on Software Verification (SV-COMP),
    Rome, Italy, March 21, 2013
13. Organization Co-Chair, 2nd Int. Workshop on Linux Driver Verification (LDV),
    Heraklion, Creta, October 15, 2012
14. General Chair, 20th IEEE Int. Conference on Program Comprehension (ICPC),
    Passau, Bavaria, Germany, June 11–13, 2012
15. Organizer, 7th Alpine Verification Meeting (AVM),
    Passau, Bavaria, Germany, May 21–22, 2012
16. Organizer, 1st TACAS 2012 Competition on Software Verification (SV-COMP),
    Tallinn, Estonia, March 29, 2012
17. Organization Chair, 1st Int. Workshop on Linux Driver Verification (LDV),
    Passau, Germany, February 13–17, 2012
18. Local Organization Chair, 31st IEEE Int. Conference on Software Engineering (ICSE),
    Vancouver, BC, Canada, May 16–24, 2009
19. Local Organization Chair, 17th IEEE Int. Conference on Program Comprehension (ICPC),
    Vancouver, BC, Canada, May 17–19, 2009
20. Local Organization Chair, 6th IEEE Working Conference on Mining Software Repositories (MSR),
    Vancouver, BC, Canada, May 16–17, 2009
21. Proceedings Chair, 25th IEEE Int. Conference on Software Maintenance (ICSM),
    Edmonton, AB, Canada, September 20–26, 2009
22. Local Organization Chair, 14th IEEE Working Conference on Reverse Engineering (WCRE),
    Vancouver, BC, Canada, October 28–31, 2007
23. Founder and Organizer, First Alpine Verification Meeting (AVM'05),
    Lausanne, Switzerland, October 6, 2005

Member of Conference Steering Committees

1. Fundamental Approaches to Software Engineering (FASE), since 2022
2. European Joint Conferences on Theory and Practice of Software (ETAPS), since 2021
3. Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), since 2015
4. IFIP Int. Conference on Formal Techniques for Distributed Objects, Components and Systems (FORTE), 2013–2016
5. IEEE Int. Conference on Program Comprehension (ICPC), 2012–2015

Member of Conference Program Committees

1. Int. Conference on Software Engineering (ICSE), 2024
2. Int. Conference on Computer-Aided Verification (CAV), 2023
3. Annual NASA Formal Methods Symposium (NFM), 2023
4. Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 2023
5. Int. Conference on Software Engineering (ICSE), 2022
6. Int. Colloquium on Theoretical Aspects of Computing (ICTAC), 2022
7. Annual NASA Formal Methods Symposium (NFM), 2022
8. IEEE/ACM Int. Conference on Automated Software Engineering (ASE), 2022
9. ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), 2022
10. Int. Conference on Software Engineering and Formal Methods (SEFM), 2022
11. Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 2022
12. Int. Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI), 2022

13. ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), 2021
14. Int. Conference on Software Engineering (ICSE), 2021
15. Int. Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI), 2021
16. Int. Conference on Tests and Proofs (TAP), 2021
17. Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 2021
18. Int. Conference on Software Engineering and Formal Methods (SEFM), 2021
19. Int. Conference on Software Engineering and Formal Methods (SEFM), 2020
20. Int. Conference on Tests and Proofs (TAP), 2020
21. ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), 2020
22. Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 2020
23. Int. Conference on Software Engineering and Formal Methods (SEFM), 2019
24. Annual NASA Formal Methods Symposium (NFM), 2019
25. Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 2019
26. Interactive Workshop on the Industrial Application of Verification and Testing (InterAVT), 2019
27. Int. Conference on Formal Methods in Software Engineering (FORMALIZE), 2019
28. Int. Conference on Software Engineering and Formal Methods (SEFM), 2018
29. Annual NASA Formal Methods Symposium (NFM), 2018
30. Int. Conference on Formal Techniques for Distributed Objects, Components, and Systems (FORTE), 2018
31. Int. Conference on Computer-Aided Verification (CAV), 2018
32. Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 2017
33. ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), 2017
34. Int. Conference on Software Engineering, tools track (ICSE), 2017
35. Annual NASA Formal Methods Symposium (NFM), 2017
36. Int. SPIN Symposium on Model Checking of Software (SPIN), 2017
37. Int. Conference on Software Testing, Machine Learning and Complex Process Analysis (TMPA), 2017
38. 31st IEEE/ACM Int. Conference on Automated Software Engineering (ASE), 2016
39. 9th Indian Software Engineering Conference (ISEC), 2016
40. 9th Int. Conference on Tests and Proofs (TAP), 2015
41. 7th NASA Formal Methods Symposium (NFM), 2015
42. 31st IEEE Int. Conference on Software Maintenance and Evolution (ICSME), 2015
43. 23rd IEEE Int. Conference on Program Comprehension (ICPC), 2015
44. 5th Int. Workshop on Formal Methods and Analysis in Software Product-Line Engineering (FMSPLE), 2015
45. 14th Int. Conference on Formal Methods in Computer-Aided Design (FMCAD), 2014
46. 6th Int. Symposium On Leveraging Applications of Formal Methods, Verification and Validation (ISoLA), 2014
47. 36th ACM/IEEE Int. Conference on Software Engineering (ICSE), 2014
48. 34nd IFIP Int. Conference on Formal Techniques for Distributed Objects, Components, and Systems (FORTE), 2014
49. 21st Int. Symposium on Model Checking of Software (SPIN), 2014
50. 22nd IEEE Int. Conference on Program Comprehension (ICPC), 2014
51. 8th Int. Conference on Tests and Proofs (TAP), 2014
52. 18th European Conference on Software Maintenance and Reengineering and 21st Working Conference on Reverse Engineering (CSMR/WCRE), 2014
53. 13th Int. Conference on Formal Methods in Computer-Aided Design (FMCAD), 2013
54. 21st IEEE Int. Conference on Program Comprehension (ICPC), 2013

55. 20th Working Conference on Reverse Engineering (WCRE), 2013
56. 7th Int. Conference on Tests and Proofs (TAP), 2013
57. 23rd Annual Int. Conference on Computer Science and Software Engineering (CASCON), 2013
58. 4rd Int. Workshop on Formal Methods and Analysis in Software Product-Line Engineering (FMSPLE), 2013
59. 19th Working Conference on Reverse Engineering (WCRE), 2012
60. 5th Int. Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA), 2012
61. 3rd Int. Workshop on Formal Methods and Analysis in Software Product-Line Engineering (FMSPLE), 2012
62. 1st Int. Workshop on Comparative Empirical Evaluation of Reasoning Systems (COMPARE), 2012
63. 19th Int. Workshop on Model Checking of Software (SPIN), 2012
64. 6th Int. Conference on Tests and Proofs (TAP), 2012
65. 32nd IFIP Int. Conference on Formal Techniques for Networked and Distributed Systems (FORTE), 2012
66. 28th IEEE Int. Conference on Software Maintenance (ICSM), 2012
67. 6th IEEE Int. Symposium on Theoretical Aspects of Software Engineering (TASE), 2012
68. 18th Working Conference on Reverse Engineering (WCRE), 2011
69. 6th Int. Workshop on Systems Software Verification (SSV), 2011
70. 31st IFIP Int. Conference on Formal Techniques for Networked and Distributed Systems (FORTE), 2011
71. 27th IEEE Int. Conference on Software Maintenance (ICSM), 2011
72. 19th IEEE Int. Conference on Program Comprehension (ICPC), 2011
73. 5th Int. Conference on Tests and Proofs (TAP), 2011
74. 17th Working Conference on Reverse Engineering (WCRE), 2010
75. 26th IEEE Int. Conference on Software Maintenance (ICSM), 2010,
Industrial Track
76. 4th IEEE Int. Symposium on Theoretical Aspects of Software Engineering (TASE), 2010
77. 4th Int. Conference on Tests and Proofs (TAP), 2010
78. 18th IEEE Int. Conference on Program Comprehension (ICPC), 2010
79. 19th Annual Int. Conference on Computer Science and Software Engineering (CASCON), 2009
80. 16th Working Conference on Reverse Engineering (WCRE), 2009
81. 17th IEEE Int. Conference on Program Comprehension (ICPC), 2009
82. 18th Annual Int. Conference on Computer Science and Software Engineering (CASCON), 2008
83. 15th Working Conference on Reverse Engineering (WCRE), 2008
84. 24th IEEE Int. Conference on Software Maintenance (ICSM), 2008
85. 19th Int. Conference on Concurrency Theory (CONCUR), 2008
86. 16th IEEE Int. Conference on Program Comprehension (ICPC), 2008
87. 23rd IEEE Int. Conference on Software Maintenance (ICSM), 2007
88. 15th IEEE Int. Conference on Program Comprehension (ICPC), 2007
89. 11th European Conference on Software Maintenance and Reengineering (CSMR), 2007,
Doctoral Symposium
90. 22nd IEEE Int. Conference on Software Maintenance (ICSM), 2006

Journal Editor

1. Editorial Board of Journal PeerJ, since 2015
2. Int. Journal on Software Tools for Technology Transfer (STTT), since 2015, Editor-in-chief for CoCha Theme

Journal Referee

Transactions on Software Engineering, ACM, 2021;

Journal on Software: Practice and Experience, Springer, 2020;
Journal of Automated Reasoning, Springer, 2019;
Transactions on Software Engineering and Methodology, ACM, 2018;
Communications of the ACM, ACM, 2018;
Transactions on Software Engineering and Methodology, ACM, 2018;
Transactions on Programming Languages and Systems, ACM, 2018;
Software Testing, Verification, and Reliability, Springer, 2018;
Journal of Automated Reasoning, Springer, 2018;
Innovations in Systems and Software Engineering, Springer, 2018
Int. Journal on Software Tools for Technology Transfer, Springer, 2017
Journal of Systems and Software, Elsevier, 2017;
Automated Software Engineering, Springer, 2016;
IEEE Software, IEEE, 2015;
Int. Journal on Software Tools for Technology Transfer (STTT), Springer, 2015;
IEEE Transactions on Software Engineering (TSE), 2014;
Int. Journal on Software Tools for Technology Transfer (STTT), Springer, 2014;
Computer Science Review (COSREV), Elsevier, 2014;
Formal Methods in System Design (FMSD/FORM), Springer, 2014;
IEEE Transactions on Software Engineering (TSE), 2013;
Software and Systems Modeling (SOSYM), Springer, 2013;
Formal Methods in System Design (FMSD/FORM), Springer, 2013;
Communications of the ACM, 2012;
Int. Journal on Software Tools for Technology Transfer (STTT), Springer, 2012;
Computing (COMP), Springer, 2012;
ACM Transactions on Software Engineering and Methodology (TOSEM), 2012;
Int. Journal on Software Testing, Verification and Reliability, 2011;
IEEE Software, 2011;
ACM Transactions on Software Engineering and Methodology (TOSEM), 2011;
Int. Journal on Software Tools for Technology Transfer (STTT), Springer, 2011;
Science of Computer Programming (SCICO), Elsevier, 2011;
Empirical Software Engineering (EMSE), Springer, 2011;
Automated Software Engineering (ASE), Springer, 2011;
Arabian Journal for Science and Engineering (AJSE), 2011;
IEEE Transactions on Software Engineering (TSE), 2010;
ACM Transactions on Software Engineering and Methodology (TOSEM), 2010;
Software Quality Journal, Special Issue on ICPC'09, 2010;
Journal for Graph Algorithms and Applications (JGAA), 2010;
IEEE Software, Special Issue on Software Evolution, 2009;
Journal of Systems and Software (JSS), 2009;
Journal on Formal Aspects of Computing (FACJ), 2009;
Int. Journal on Software Tools for Technology Transfer (STTT), 2009;
IEEE Transactions on Software Engineering (TSE), 2009;
Journal of Software Maintenance and Evolution: Research and Practice (JSME), 2009;
IEEE Systems Journal, 2008;
IEEE Transactions on Software Engineering (TSE), 2008;
ACM Transactions on Design Automation of Electronic Systems (TODAES), 2008;
Simulation Modelling Practice and Theory, 2008;
ACM Transactions on Software Engineering and Methodology (TOSEM), 2007;

IEEE Transactions on Software Engineering (TSE), 2007;
Int. Journal of Computers and Their Applications (IJCA), 2007;
Int. Journal on Software Tools for Technology Transfer (STTT), 2006;
ACM Transactions on Software Engineering and Methodology (TOSEM), 2006;
ACM Transactions on Design Automation of Electronic Systems (TODAES), 2005

Conference Referee

Int. Symposium on Formal Methods (FM), 2012;
Int. Conf. on Verification, Model Checking, and Abstract Interpretation (VMCAI), 2012;
Festschrift Manfred Nagl (FMN), 2009;
Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 2009;
Int. Conference on Verification, Model Checking and Abstract Interpretation (VMCAI), 2009;
Int. Conference on Computer-Aided Verification (CAV), 2008;
Joint Eurographics - IEEE TCVG Symposium on Visualization (EUROVIS), 2008;
ACM Symposium on Principles of Programming Languages (POPL), 2007;
ACM Int. Symposium on Principles and Practice of Declarative Programming (PPDP), 2007;
ACM/IEEE Int. Conference on Software Engineering (ICSE), 2006;
Int. Conf. on Foundations of Software Science and Computation Structures (FOSSACS), 2006;
ACM Int. Symposium on Component-Based Software Engineering (CBSE), 2006;
Int. Symposium on Software Composition (SC), 2006;
Int. Conference on Computer-Aided Verification (CAV), 2005;
Int. Symposium on Formal Methods (FM), 2005;
ACM Int. Symposium on Component-Based Software Engineering (CBSE), 2005;
Monterey Workshop: Software Engineering Tools, 2005

Other Referee (Grants/Awards)

Swiss National Science Foundation (SNF), 2013;
South Africa's National Research Foundation (NRF), 2013;
German Science Foundation (DFG), 2012;
Czech Science Foundation (GACR), 2011;
German Science Foundation (DFG), 2011;
ACM Student Research Competition at ICSE (ICSE-SRC), 2011;
The Netherlands Organization for Scientific Research (NWO), 2010;
Austrian Science Fund (FWF), 2009;
The Villum Kann Rasmussen Foundation, 2008;
ACM SIGPLAN 2006 Dissertation Award Committee, 2007;
Natural Sciences and Engineering Research Council of Canada (NSERC), 2007;
The Netherlands Organization for Scientific Research (NWO), 2005;
The Netherlands Organization for Scientific Research (NWO), 2004

Consultant in Industry

Argus Technologies Ltd., Burnaby, 2009
Alpha Technologies Ltd., Burnaby, 2008

Member of Professional Organizations

Association for Computing Machinery (ACM)
IEEE, IEEE Computer Society
Deutscher Hochschulverband (DHV)

References

1. Thomas A. Henzinger, Professor, IST Austria, http://pub.ist.ac.at/∼tah
2. Claus Lewerentz, Professor, BTU Cottbus, Germany, http://www-sst.informatik.tu-cottbus.de
3. Carlo Ghezzi, Professor, Politecnico di Milano, Italy, http://home.dei.polimi.it/ghezzi

Additional references are available on request.