

## Bernd Fischer

Division of Computer Science, University of Stellenbosch  
Private Bag X1, 7602 Matieland, South Africa  
bfischer@cs.sun.ac.za

### Career Summary:

- **Core competency** in software engineering, formal methods, program analysis, generation, and verification, and artificial intelligence.
- **110 reviewed publications**, including 12 journal papers and 67 conference contributions, h-index 22, **Paper / Presentation awards** at ICSE'11, Safecomp'10, and ASE'98. **ASE 2012 Most Influential Paper Award**.
- NRF-rated researcher, **B2** rating
- **Member of IFIP WG 2.11, SC member** of ASE and GPCE conferences, **General chair** ICTAC'18, GPCE'16, SPIN'15, and LPAR'13 conferences, **PC co-chair** GPCE'09 and ASE'07 conferences, regular event co-organizer, PC member and reviewer, **invited speaker or panelist** at FTP'07, ASE'06, ESSCaSS'06, GPCE'05, ASE'99, 25 invited presentations.
- **Developer** of HAMMR, AutoBayes, AutoFilter, AutoCert, ESBMC, and CSeq **tools**.
- Active **international research collaborations** with NASA Ames, USA, U Southampton, UK, and UFAM, Brazil.
- Proven track record of securing **research funding** from NASA, EPSRC, Royal Society, and NRF (**career total in excess of US\$ 2,800,000**).

### Professional Experience:

**Professor**, Division of Computer Science, University of Stellenbosch (since 1/2014)

**Associate Professor**, Division of Computer Science, University of Stellenbosch (2/2013-12/2013)

- Research in software engineering, formal methods, program generation, and program verification.
- Continued work on program generation and software safety certification.
- Continued work on software model checking.
- Secured competed research funding from NRF.

**Senior Lecturer**, Electronics and Computer Science, University of Southampton (2/2006-10/2013)

- part-time appointment 2/2013-10/2013.
- Continued work on program generation and software safety certification, developed approach to automatically derive safety cases.
- Developed ESBMC, award-winning SMT-based bounded model checker for C.
- Developed and delivered undergraduate and graduate modules on programming, software engineering, compiler engineering, and code generation. Excellent teaching evaluations.
- Leader of a research term with three post-doctoral researchers and six PhD students. Graduated two PhD students.
- Sabbatical leave at NASA Ames and University of Auckland (7/2009-1/2010)
- Responsible for four other academics as Deputy Head of Group (10/2011-1/2013). ECS Erasmus co-ordinator (international student exchanges) (7/2007-1/2013).
- Secured competed research funding from EPSRC and the Royal Society.

**Computer Scientist**, RIACS, NASA Ames Research Center (11/1998-1/2006)

- Developed a schema-based synthesis approach and implemented two systems AutoBayes (for statistical data analysis) and AutoFilter (for state estimation) that have been applied successfully to applications within NASA.

- Developed a Hoare-style approach which automatically certifies safety properties of the generated code and extends proof-carrying code to the source level.
- Secured competed research funding from NASA.

**Research/Teaching Associate**, Dept. of Software Technology, TU Braunschweig (11/1990-9/1998)

- Responsible for the NORA/HAMMR project to exploit semantic information and automated reasoning to retrieve components from software libraries. Developed and evaluated HAMMR tool.
- Supervised and delivered undergraduate- and graduate labs, classroom exercises, and seminars. Developed and delivered graduate module on unification theory.

**Education:**

**PhD** in Computer Science, *summa cum laude*, 6/2001, TU Braunschweig/U Passau

Thesis: *Deduction-Based Software Component Retrieval* (Supervisor: Prof. Dr. G. Snelting)

Nominated for GI Dissertationspreis for best German PhD Thesis in Computer Science.

**MSc** in Computer Science, *with distinction*, 9/1990, TU Braunschweig,

including 7 months industrial experience at Siemens Corporate Research Lab, Munich

**Publications - book chapters:**

1. S. H. Ghotbi and B. Fischer. “Fine-Grained Role- and Attribute-Based Access Control for Web Applications”. In: *Software and Data Technologies, Communications in Computer and Information Science 411*, Selected papers from ICSOFT 2012, pp. 171–187. Springer, 2013.
2. B. Fischer and E. Visser. “Retrofitting the AutoBayes Program Synthesis System with Concrete Syntax”. In C. Lengauer, D. Batory, C. Consel, and M. Odersky (Eds.), *Domain-Specific Program Generation, LNCS 3016*, pp. 239–253. Springer, 2004.
3. B. Fischer. “Deduction-Based Software Component Retrieval”. In D. Wagner *et al.* (Eds.), *Ausgezeichnete Informatikdissertationen 2001*, pp. 19–28. *Lecture Notes in Informatics D-2*, Köllen Verlag, Bonn, 2002. In German.
4. B. Fischer, J. Schumann, and G. Snelting. “Deduction-Based Software Component Retrieval”. In W. Bibel and P. Schmitt (Eds.), *Automated Deduction - A Basis for Applications*, pp. 265–292. Kluwer, Dordrecht, 1998.

**Publications - journal articles:**

5. M. Dunaiski, G. J. Greene, and B. Fischer. “Exploratory Search of Academic Publication and Citation Data using Interactive Tag Cloud Visualizations.” *Scientometrics*, 2017, to appear.
6. G. J. Greene, M. Esterhuizen, and B. Fischer. “Visualizing and Exploring Software Version Control Repositories using Interactive Tag Clouds over Formal Concept Lattices.” *Information and Software Technology*, 2017, to appear.
7. T. Aubrey-Jones and B. Fischer. “Synthesizing MPI Implementations from Functional Data-Parallel Programs”. *Intl. J. Parallel Programming*, 44(3): 552-573, 2016.
8. J. Morse, L. Cordeiro, D. Nicole, and B. Fischer. “Model Checking LTL Properties over C Programs with Bounded Traces”. *J. Software and Systems Modelling*, 14(1):65-81, 2015.
9. J. Morse, L. Cordeiro, D. Nicole, and B. Fischer. “Applying Symbolic Bounded Model Checking to the 2012 RERS Greybox Challenge”. *Intl. J. Software Tools for Technology Transfer*, 16:519-529, 2014.

10. L. Cordeiro, B. Fischer, and J. Marques-Silva. “SMT-Based Bounded Model Checking for Embedded ANSI-C Software”. *IEEE Trans. Software Engineering*, 38(4):957–974, 2012.
11. E. Denney, B. Fischer, and J. Schumann. “An Empirical Evaluation of Automated Theorem Provers in Software Certification”. *International Journal on Artificial Intelligence Tools*, 15(1):81–107, 2006.
12. B. Fischer and J. Schumann. “AutoBayes: A System for Generating Data Analysis Programs from Statistical Models”. *J. Functional Programming*, 13(3):483–508, 2003.
13. B. Fischer. “Specification-Based Browsing of Software Component Libraries”. *Automated Software Engineering*, 7(2):179–200, 2000.
14. T. Baar, B. Fischer, and D. Fuchs. “Integrating Deduction Techniques in a Software Reuse Application”. *J. Universal Computer Science*, 5(3):52–72, 1999.
15. G. Snelling, B. Fischer, F.-J. Grosch, M. Kievernagel, and A. Zeller. “Die inferenzbasierte Softwareentwicklungsumgebung NORA”. *Informatik–Forschung und Entwicklung*, 9(3):116–131, 1994. In German.
16. B. Fischer, C. Hammer, and W. Struckmann. “ALADIN: A Scanner Generator for Incremental Programming Environments”. *Software–Practice & Experience*, 22(11):1011–1025, November 1992.

**Publications - conference contributions:**

17. G. J. Greene and B. Fischer. “CVExplorer: identifying candidate developers by mining and exploring their open source contributions”. *Proc. 31st Intl. Conf. Automated Software Engineering (ASE), Tools track*, pp. 804–809, 2016.
18. E. Tomasco, T. L. Nguyen, O. Inverso, B. Fischer, S. La Torre, and G. Parlato. “Lazy Sequentialization for TSO and PSO via Shared Memory Abstractions”. *Proc. 16th Intl. Conf. Formal Methods in Computer-Aided Design (FMCAD)*, pp. 193–200, 2016.
19. T. L. Nguyen, B. Fischer, S. La Torre, and G. Parlato. “Lazy Sequentialization for the Safety Verification of Unbounded Concurrent Programs”. *Proc. 14th Intl. Symp. Automated Technology for Verification and Analysis (ATVA). LNCS 9938*, pp. 174–191, 2016.
20. G. Birch, B. Fischer, and M. Poppleton. “Using Fast Model-Based Fault Localisation to Aid Students in Self-Guided Program Repair and to Improve Assessment”. *Proc. 2016 ACM Conf. Innovation and Technology in Computer Science Education (ITiCSE)*, pp. 168–173, 2016.
21. E. Tomasco, T. L. Nguyen, O. Inverso, B. Fischer, S. La Torre, and G. Parlato. “MU-CSeq 0.4: Individual Memory Location Unwindings - (Competition Contribution)”. *Proc. 22nd Intl. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS). LNCS 9636*, pp. 938–941, 2016.
22. O. Inverso, T. L. Nguyen, B. Fischer, S. La Torre, and G. Parlato. “Lazy-CSeq: A Context-Bounded Model Checking Tool for Multi-Threaded C-Programs”. *Proc. 30th Intl. Conf. Automated Software Engineering (ASE), Tools track*, pp. 807–812, 2015.
23. G. Greene and B. Fischer. “Interactive Tag Cloud Visualization of Software Version Control Repositories”. *Proc. 3rd IEEE Working Conf. Software Visualization (VISSOFT)*, pp.56–65, 2015.
24. G. Birch, B. Fischer, and M. Poppleton. “Fast Model-Based Fault Localisation with Test Suites”. *Proc. 9th Intl. Conf. Testing and Proofs (TAP), LNCS 9154*, pp. 38–57, 2015.
25. E. Tomasco, O. Inverso, B. Fischer, S. La Torre, and G. Parlato. “Verifying Concurrent Programs by Memory Unwinding”. *Proc. 20th Intl. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS). LNCS 9035*, pp. 551–565, 2015.

26. E. Tomasco, O. Inverso, B. Fischer, S. La Torre, and G. Parlato. “MU-CSeq 0.3: Sequentialization by Read-Implicit and Coarse-Grained Memory Unwindings (Competition Contribution)”. *Proc. 20th Intl. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. LNCS 9035, pp. 436-438, 2015.
27. T. L. Nguyen, B. Fischer, S. La Torre, and G. Parlato. “Unbounded Lazy-CSeq: A Lazy Sequentialization Tool for C Programs with Unbounded Context Switches (Competition Contribution)”. *Proc. 20th Intl. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. LNCS 9035, pp. 461-463, 2015.
28. Gillian J. Greene and B. Fischer. “ConceptCloud: A Tagcloud Browser for Software Archives”. *Proc. 14th Intl. Conf. Foundations of Software Engineering (FSE)*, Research tool demonstrations track. pp. 759-762, 2014.
29. O. Inverso, E. Tomasco, B. Fischer, S. La Torre, and G. Parlato. “Bounded Model Checking of Multi-Threaded C Programs via Sequentialization”. *Proc. 26th Intl. Conf. Computer Aided Verification (CAV)*. LNCS 8559, pp. 585-602, 2014.
30. T. Aubrey-Jones and B. Fischer. “Automatic Data Distribution for Data-Parallel Programming”. *Proc. 7th Intl. Symposium on High-level Parallel Programming and Applications*. 2014.
31. J. Morse, M. Ramalho, L. Cordeiro, D. Nicole, and B. Fischer. “ESBMC 1.22 (Competition Contribution)”. *Proc. 20th Intl. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. LNCS 8413, pp. 405-407, 2013.
32. E. Tomasco, O. Inverso, B. Fischer, S. La Torre, and G. Parlato. “MU-CSeq: Sequentialization of C Programs by Shared Memory Unwindings (Competition Contribution)”. *Proc. 20th Intl. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. LNCS 8413, pp. 402-404, 2014.
33. O. Inverso, E. Tomasco, B. Fischer, S. La Torre, and G. Parlato. “Lazy-CSeq: A Lazy Sequentialization Tool for C (Competition Contribution)”. *Proc. 20th Intl. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. LNCS 8413, pp. 398-401, 2014.
34. B. Fischer, O. Inverso, and G. Parlato. “CSeq: A Concurrency Pre-Processor for Sequential C Verification Tools”. *Proc. 28th Intl. Conf. Automated Software Engineering (ASE), Tools track*, pp. 710-713, 2013.
35. N. Grech, B. Fischer, and J. Rathke. “Preemptive Type Checking in Dynamically Typed Languages”. *10th Intl. Colloquium Theoretical Aspects of Computing (ICTAC)*, LNCS, pp. 195-212, 2013.
36. M. Ramalho, M. Freitas, F. Sousa, H. Marques, L. Cordeiro, and B. Fischer. “SMT-Based Bounded Model Checking of C++ Programs”. *Proc. 20th IEEE Intl. Conf. Workshops Engineering of Computer-Based Systems (ECBS)*, pp. 147-156, 2013.
37. J. Morse, L. Cordeiro, D. Nicole, and B. Fischer. “Handling Unbounded Loops with ESBMC 1.20 (Competition Contribution)”. *Proc. 19th Intl. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. LNCS 7795, pp. 619-622, 2013.
38. B. Fischer, O. Inverso, and G. Parlato. “CSeq: A Sequentialization Tool for C (Competition Contribution)”. *Proc. 19th Intl. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. LNCS 7795, pp. 616-618, 2013.
39. S. H. Ghotbi and B. Fischer. “A Declarative Fine-Grained Role-Based Access Control Model and Mechanism for the Web Application Domain”. *Proc. 7th Intl. Conf. Software Paradigm Trends (IC-SOFT)*, pp. 80–91. SciTePress, 2012.

[Presentation, 11%]

40. J. Morse, L. Cordeiro, D. Nicole, and B. Fischer. “Context-Bounded Model Checking with ESBMC 1.17 (Competition Contribution)”. *Proc. 18th Intl. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. LNCS 7214, pp. 534-537, 2012.
41. J. Morse, L. Cordeiro, D. Nicole, and B. Fischer. “Context-Bounded Model Checking of LTL Properties for ANSI-C Software”. *Proc. 9th Intl. Conf. Software Engineering and Formal Methods (SEFM)*. LNCS 7431, pp. 302-317, 2011. [Presentation, 29%]
42. S. Esmailsabzali, B. Fischer and J. Atlee. “Monitoring Aspects for the Customization of Automatically Generated Code for Big-Step Models”. *Proc. 10th Intl. Conf. Generative Programming and Component Engineering (GPCE)*, pp. 117-126, 2011. ACM Press. [Presentation, 31%]
43. B. Fischer, R. Laemmel and V. Zaytsev. “Comparison of Context-free Grammars Based on Parsing Generated Test Data”. *Proc. 4th Intl. Conf. Software Language Engineering (SLE 2011)*, LNCS 6940. [36%]
44. L. Cordeiro and B. Fischer. “Bounded Model Checking for Multi-threaded Software using SMT-Solvers”. *Proc. 33rd Intl. Conf. Software Engineering (ICSE)*, pp. 331-340. ACM Press, 2011. **ACM Distinguished Paper Award**. [Presentation, 14%]
45. A. Darbari, B. Fischer, and J. Marques-Silva. “Industrial-Strength Certified SAT Solving through Verified SAT Proof Checking”, *Proc. 7th Intl. Colloquium Theoretical Aspects of Computing (ICTAC)*, LNCS 6255, pp. 260-274. Springer, 2010. [Presentation, 29%]
46. N. Grech, B. Fischer, and J. Rathke. “Generating correct and efficient equality and hashing methods using JEqualityGen”. *Proc. 9th Intl. Conf. Generative Programming and Component Engineering (GPCE)*, pp. 177-186. ACM Press, 2010. [Presentation, 31%]
47. N. Basir, E. Denney and B. Fischer. “Deriving Safety Cases for Hierarchical Systems in Model-based Development”. *Proc. 29th Intl. Conf. Computer Safety, Reliability and Security (SAFECOMP)*, LNCS 6351, pp. 68-81. Springer, 2010. **Best Presentation Award**. [Presentation]
48. L. Cordeiro, B. Fischer, and J. Marques-Silva. “Continuous Verification of Large Embedded Software Using SMT-Based Bounded Model Checking”. *Proc. 17th IEEE Intl. Conf. Workshops Engineering of Computer-Based Systems (ECBS)*, pp. 160-169, 2010. [Presentation]
49. N. Basir, E. Denney and B. Fischer. “Deriving Safety Cases from Automatically Constructed Proofs”. *Proc. 4th IET International Conference on System Safety*, 2009. [Presentation]
50. L. Cordeiro, B. Fischer, and J. Marques-Silva. “SMT-Based Bounded Model Checking for Embedded ANSI-C Software”. *Proc. 24th Intl. Conf. Automated Software Engineering (ASE)*, pp. 137-148, 2009. [Presentation, 17%]
51. E. Denney and B. Fischer. “A Verification-Driven Approach to Traceability and Documentation for Auto-Generated Mathematical Software”. *Proc. 24th Intl. Conf. Automated Software Engineering (ASE)*, pp. 560-564, 2009. [Poster, 32%]
52. P. Matos, B. Fischer, J. Marques-Silva. “A Lazy Unbounded Model Checker for Event-B”. *Proc. 11th Intl. Conf. Formal Engineering Methods (ICFEM)*, LNCS 5885, pp. 485-503, 2009. Springer. [Presentation, 30%]
53. E. Denney and B. Fischer. “Generating code review documentation for auto-generated mission-critical software”. *Proc. Third IEEE Intl. Conf. Space Mission Challenges for Information Technology*, 2009.
54. B. Fischer, A. Saabas, and T. Uustalu. “Program Repair as Sound Optimization of Broken Programs”. *Proc. 3rd Intl. Conf. Theoretical Aspects of Software Engineering (TASE)*, pp. 165-173, 2009. [Presentation, 27%]

55. L. Cordeiro, B. Fischer, H. Chen and J. Marques-Silva “Semiformal Verification of Embedded Software in Medical Devices Considering Stringent Hardware Constraints”. *Proc. 6th Intl. Conf. Embedded Software and Systems*, pp. 396–403. 2009. [Presentation]
56. E. Denney and B. Fischer. “Generating Customized Verifiers for Automatically Generated Code”. *Proc. 7th Intl. Conf. Generative Programming and Component Engineering (GPCE)*, pp. 77–88, 2008. ACM Press. [Presentation, 30%]
57. N. Basir, E. Denney and B. Fischer. “Constructing a Safety Case for Automatically Generated Code from Formal Program Verification Information”, *Proc. 27th Intl. Conf. Computer Safety, Reliability and Security (SAFECOMP), LNCS 5219*, pp. 249–262. Springer, 2008. [Presentation, 38%]
58. E. Denney and B. Fischer. “Explaining Verification Conditions”, *Proc. 12th Intl. Conf. Algebraic Methodology and Software Technology, LNCS 5140*, pp. 145–159. Springer, 2008. [47%]
59. E. Denney and B. Fischer. “Extending Source Code Generators for Evidence-based Software Certification”. *Proc. 2nd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation*. 2006.
60. E. Denney and B. Fischer. “A Generic Annotation Inference Algorithm for the Safety Certification of Automatically Generated Code”. *Proc. Conf. Generative Programming and Component Engineering (GPCE)*, pp. 121–130. ACM Press, 2006. [35%]
61. E. Denney and B. Fischer. “Annotation Inference for Safety Certification of Automatically Generated Code (Extended Abstract)”. *Proc. 21st Intl. Conf. Automated Software Engineering (ASE)*, pp. 265–268. IEEE Comp. Soc. Press, 2006. [Poster, 22%]
62. A. Srivastava, J. Schumann, B. Fischer. “An Ensemble Approach to Building Mercer Kernels with Prior Information”. *Proc. IEEE Conf. Systems, Man, and Cybernetics*, 2005.
63. E. Denney and B. Fischer. “Certifiable Program Generation”. *Proc. 4th Intl. Conf. Generative Programming and Component Engineering (GPCE), LNCS 3676*, pp. 17–28, Springer, 2005. Invited paper. [Presentation]
64. J. Richardson, J. Schumann, B. Fischer, and E. Denney. “Rapid Exploration of the Design Space During Automatic Generation of Kalman Filter Code”. *Proc. IEEE Aerospace Conf.*, March 2005.
65. E. Denney, B. Fischer, J. Schumann, and J. Richardson. “Automatic Certification of Kalman Filters for Reliable Code Generation”. *Proc. IEEE Aerospace Conf.*, March 2005.
66. K. Huyser, B. Fischer, K. Knuth, J. Schumann, D. Granquist-Fraser, and A. Hajian. “Discovering Planetary Nebula Geometries: Explorations with a Hierarchy of Models”. *Proc. 24rd Conf. Bayesian Maximum Entropy Methods*, pp. 135–142. AIP Press, 2004.
67. E. Denney, B. Fischer, and J. Schumann. “Using Automated Theorem Provers to Certify Auto-Generated Aerospace Software”. *Proc. 2nd Intl. Joint Conf. Automated Reasoning (IJCAR), LNCS 3097*, pp. 198–212. Springer, 2004. [Presentation, 35%]
68. E. Denney, B. Fischer, and J. Schumann. “Adding Assurance to Automatically Generated Code (Fast Abstract)”. *Proc. 8th IEEE Intl. Symp. High Assurance Systems Engineering (HASE)*, pp. 297–299. IEEE Comp. Soc. Press, 2004.
69. B. Fischer and J. Schumann. “Applying AutoBayes to the Analysis of Planetary Nebulae Images (Extended Abstract)”. *Proc. 18th Intl. Conf. Automated Software Engineering (ASE)*, pp. 337–342. IEEE Comp. Soc. Press, 2003. [Poster, 25%]

70. E. Denney and B. Fischer. “Correctness of Source-Level Safety Policies”. *Proc. Formal Methods (FM), LNCS 2805*, pp. 894–913. Springer, 2003. [31%]
71. B. Fischer, K. Knuth, A. Hajian, and J. Schumann. “Automatic Derivation of Statistical Data Analysis Programs: Planetary Nebulae and Beyond”. *Proc. 23rd Conf. Bayesian Maximum Entropy Methods*, pp. 276–291. AIP Press, 2003. [Presentation]
72. J. Schumann, B. Fischer, M. Whalen, and J. Whittle. “Certification Support for Automatically Generated Programs”. *Proc. 36th Hawaii International Conference on System Sciences*. 2003.
73. A. Gray, B. Fischer, J. Schumann, and W. Buntine. “Automatic Derivation of Statistical Algorithms: The EM Family and Beyond”. *Proc. Advances in Neural Information Processing Systems 15 (NIPS\*2002)*, pp. 689–696. MIT Press, 2003. [Poster, 31%]
74. M. Whalen, J. Schumann, and B. Fischer. “AutoBayes/CC — Combining Program Synthesis with Automatic Code Certification (System Description)”. *Proc. 18th Intl. Conf. Automated Deduction (CADE), LNAI 2392*, pp. 290–294. Springer, 2002. [Presentation]
75. M. Whalen, J. Schumann, and B. Fischer. “Synthesizing Certified Code”. *Proc. Formal Methods Europe (FME), LNCS 2391*, pp. 431–450. Springer, 2002. [Presentation, 33%]
76. B. Fischer, T. Pressburger, G. Rosu, and J. Schumann. “The AutoBayes Program Synthesis System—System Description”. *Proc. 9th Symp. Integration of Symbolic Computation and Mechanized Reasoning*, pp. 118–125. 2001.
77. J. Penix, B. Fischer, J. Whittle, G. Pour, and J. Van Baalen. “Automating Component Integration for Web-Based Data Analysis Applications”. *Proc. IEEE Aerospace Conf.*, March 2000.
78. B. Fischer and J. Whittle. “An Integration of Deductive Retrieval into Deductive Synthesis”. *Proc. 14th Intl. Conf. Automated Software Engineering (ASE)*, pp. 52–61. IEEE Comp. Soc. Press, 1999. [Presentation, 20%]
79. W. Buntine, B. Fischer, and T. Pressburger. “Towards Automated Synthesis of Data Mining Programs”. *Proc. 5th Intl. Conf. Knowledge Discovery and Data Mining (KDD)*, pp. 372–376. ACM Press, 1999. [Poster]
80. B. Fischer. “Specification-Based Browsing of Software Component Libraries”. *Proc. 13th Intl. Conf. Automated Software Engineering (ASE)*, pp. 74–83, IEEE Comp. Soc. Press, 1998. **Best Paper Award** and **ASE 2012 Most Influential Paper Award**. [Presentation, 16%]
81. J. Schumann and B. Fischer. “NORA/HAMMR: Making Deduction-Based Software Component Retrieval Practical”. *Proc. 12th Intl. Conf. Automated Software Engineering (ASE)*, pp. 246–254. IEEE Comp. Soc. Press, 1997. [30%]
82. B. Fischer and J. Schumann. “SETHEO Goes Software Engineering: Application of ATP to Software Reuse”. *Proc. 14th Intl. Conf. Automated Deduction (CADE), LNAI 1249*, pp. 65–68. Springer, 1997. [Presentation]
83. B. Fischer, M. Kievernagel, and W. Struckmann. “High-precision Retrieval for High-Quality Software”. *Proc. 4th Software Quality Conf.*, pp. 80–88, University of Abertay Dundee, 1995.

#### **Publications - reviewed workshop contributions:**

84. G. J. Greene and B. Fischer. “Single-Focus Broadening Navigation in Concept Lattices”. *Proc. 3rd Workshop Concept Discovery in Unstructured Data*, pp. 32–43, 2016.
85. R. Barreto, L. Cordeiro, and B. Fischer. “Verifying Embedded C Software with Timing Constraints using an Untimed Bounded Model Checker”. *Proc. SBESC Workshop on Real-Time Systems*, 2011.

86. R. Economopoulos and B. Fischer. “Higher-order Transformations with Nested Concrete Syntax”. *Proc. 11th Workshop on Language Descriptions, Tools and Applications (LDTA)*, 2011.
87. A. Jorgensen, R. Economopoulos and B. Fischer. “VLex: Visualizing a Lexical Analyzer Generator”. *Proc. 11th Workshop on Language Descriptions, Tools and Applications (LDTA)*, 2011.
88. N. Basir, E. Denney and B. Fischer. “Building Heterogeneous Safety Cases for Automatically Generated Code” *Proc. Infotech@AIAA*, Mar. 2011.
89. L. Cordeiro and B. Fischer. “Bounded Model Checking for Multi-threaded Software using SMT-Solvers” *Proc. 8th Intl. Workshop on Satisfiability Modulo Theories*, 2010.
90. N. Grech, B. Fischer, and J. Rathke. “Generating correct and efficient equality and hashing methods using JEqualityGen” *Proc. ETAPS Workshop on Generative Techniques*, 2010.
91. A. Darbari, B. Fischer, and J. Marques-Silva. “Formalizing a SAT Proof Checker in Coq” *Proc. First Intl. Coq Workshop*, 2009.
92. M. Poppleton, B. Fischer, C. Franklin, A. Gondal, C. Snook, and J. Sorge “Towards Reuse with Feature-Oriented Event-B” *Proc. GPCE/OOPSLA Workshop on Modularization, Composition and Generative Techniques for Product Line Engineering*, 2008.
93. N. Basir, E. Denney and B. Fischer. “Deriving Safety Cases for the Formal Certification of Automatically Generated Code” *Proc. ETAPS Workshop on Software Certification*, 2008.
94. E. Denney and B. Fischer. “Formal Safety Certification of Auto-Generated Aerospace Software”. *Proc. Infotech@AIAA*, Sep. 2005.
95. E. Denney and B. Fischer. “Software Certificate Management (Position Paper)” *Proc. ASE Workshop on Software Certificate Management*, 2005.
96. G. Sutcliffe, E. Denney and B. Fischer. “Practical Proof Checking for Program Certification”, *Proc. CADE-20 Workshop on Empirically Successful Classical Automated Reasoning*. Tallinn, July 2005.
97. E. Denney and B. Fischer. “A Program Certification Assistant Based on Fully Automated Theorem Provers”, *Proc. Workshop User Interfaces for Theorem Provers*. Edinburgh, April 2005.
98. E. Denney, B. Fischer, and J. Schumann. “An Empirical Evaluation of Automated Theorem Provers in Software Certification”. *Proc. IJCAR Workshop on Empirically Successful First Order Reasoning*. Cork, Ireland, July 2004. *Electronic Notes Theoretical Comp. Sci.*
99. B. Fischer and E. Visser. “Adding Concrete Syntax to a Prolog-Based Program Synthesis System (Extended Abstract)”, *Proc. Intl. Symp. Logic-based Program Synthesis and Transformation 2003*, pp. 39-48, 2003. Report CW 365, Katholieke Universiteit Leuven.
100. B. Fischer and G. Rosu. “Interpreting Abstract Interpretations in Membership Equational Logic”. *Proc. Intl. Workshop Rule-Based Programming, Electronic Notes Theoretical Comp. Sci.* 59.4, pp. 118–125. Elsevier, 2001.
101. B. Fischer, J. Schumann, and T. Pressburger. “Generating Data Analysis Programs from Statistical Models (Position Paper)”. *Proc. Intl. Workshop Semantics Applications, and Implementation of Program Generation, LNCS 1924*, pp. 212–229. Springer, 2000.
102. T. Kaiser, B. Fischer, and W. Struckmann. “MOPS: Verifying Modula-2 programs specified in VDM-SL”. *Proc. 4th Workshop Tools for System Design and Verification*, pp. 163–167, Reisingburg, July 2000.



103. B. Fischer, M. Lowry, and J. Penix. “Intelligent Component Retrieval via Automated Reasoning”. *Proc. AAAI-99 Workshop on Intelligent Software Engineering*, Orlando, FL, July 1999.
104. W. Buntine, B. Fischer, K. Havelund, M. Lowry, T. Pressburger, S. Roach, P. Robinson, and J. Van Baalen. “Transformation Systems at NASA Ames”. *Proc. ICSE-21 Intl. Workshop Software Transformation Systems*, pp. 8–13, Los Angeles, CA, May 1999.
105. T. Baar and B. Fischer. “Solving Software Reuse Problems with Theorem Provers”. *Proc. CADE-15 Workshop on Problem Solving Methodologies with Automated Deduction*, pp. 217–247, Lindau, July 1998.
106. T. Baar, B. Fischer, and D. Fuchs. “Experiments with ATP Integration in a Software Engineering Application”. *Proc. CADE-15 Workshop on Integration of Deduction Systems*, pp. 19–27, Lindau, July 1998.
107. B. Fischer and G. Snelling. “Reuse by Contract”. *Proc. ESEC-FSE Workshop on Foundations of Component-Based Systems*, pp. 91–100, Zürich, September 1997.
108. B. Fischer, M. Kievernagel, and G. Snelling. “Deduction-Based Software Component Retrieval”. *Working Notes of the IJCAI-95 Workshop: Formal Approaches to the Reuse of Plans, Proofs, and Programs*, pp. 1–5, Montréal, August 1995.
109. B. Fischer, M. Kievernagel, and W. Struckmann. “VCR: A VDM-based Software Component Retrieval Tool”. *Working Notes of the ICSE-17 Workshop on Formal Methods Application in Software Engineering Practice*, pp. 30–38, Seattle, Wash., April 1995.
110. B. Fischer. “Resolution for Feature Logics”. *Proc. Workshop GI-Fachgruppe “Alternative Konzepte für Sprachen und Rechner”*, pp. 23–34. Bad Honnef, Germany, August 1993.

**Publications - other:**

111. J. Schumann, H. Jafari, T. Pressburger, E. Denney, W. Buntine, and B. Fischer. “AutoBayes Program Synthesis System—Users Manual”. Technical Report NASA/TM-2008-215366, Ames Research Center.

**Dissertations:**

112. B. Fischer. “Deduction-Based Software Component Retrieval”. PhD Thesis, Universität Passau, defended on June 1, 2001.
113. B. Fischer. “A Scanner Generator for Incremental Programming Environments”. MSc Thesis, TU Braunschweig, 1990. In German.

**Edited volumes:**

1. B. Fischer, I. Schaefer (Eds.). *Proc. 2016 ACM SIGPLAN Intl. Conf. Generative Programming: Concepts and Experiences (GPCE '16)*. Amsterdam, 2016.
2. B. Fischer, J. Geldenhuys (Eds.). *Model Checking Software - 22nd International Symposium (SPIN 2015)*. LNCS 9232, Springer, 2015.
3. B. Fischer, O. Nierstrasz (Eds.). *Eighth International Conference on Generative Programming and Component Engineering (GPCE'09)*. Special Section of the *Journal of Object Technology*, 10, 2011.
4. B. Fischer, J. Siek (Eds.). *Proc. Eighth International Conference on Generative Programming and Component Engineering (GPCE '09)*. Denver, 2009.

5. K. Stirewalt, A. Egyed, B. Fischer (Eds.). Special Issue of the *Journal of Automated Software Engineering*, 16(1), 2009.
6. K. Stirewalt, A. Egyed, B. Fischer (Eds.). *Proc. 22nd IEEE/ACM International Conference on Automated Software Engineering (ASE '07)*. Atlanta, 2007.
7. C. Benzmüller, B. Fischer, S. Sutcliffe (Eds.). *Proc. 6th International Workshop on the Implementation of Logics*. Phnom Penh, 2006.
8. B. Fischer, S. Schulz, S. Sutcliffe (Eds.). *Empirically Successful Automated Reasoning: Systems Issue*. Special Issue of the *Journal of Automated Reasoning*, 37(4), 2006.
9. B. Fischer, S. Schulz, S. Sutcliffe (Eds.). *Empirically Successful Automated Reasoning: Applications Issue*. Special Issue of the *Journal of Automated Reasoning*, 37(1-2), 2006.
10. E. Denney, B. Fischer, M. Jones, D. Hutter (Eds.). *Proc. 2005 Automated Software Engineering Workshop on Software Certificate Management (SoftCeMent'05)*. Long Beach, 2005.
11. B. Fischer, S. Schulz, S. Sutcliffe (Eds.). *Proc. CADE-20 Workshop on Empirically Successful Classical Automated Reasoning*. Tallinn, 2005.
12. B. Fischer and E. Visser (Eds.). *Proc. 2002 ACM SIGPLAN Workshop on Rule-Based Programming (RULE'02)*. ACM Press, 2002.
13. B. Fischer and D. Smith (Eds.). *Logic-Based Program Synthesis: State of the Art and Future Trends. Papers from 2002 AAAI Spring Symposium*. The AAAI Press, 2002. Summary in *AI Magazine*, 23(4), pp. 101-108, 2002.

**Note:** Electronic versions of most publications are available via the DBLP at

[www.informatik.uni-trier.de/~ley/pers/hd/f/Fischer\\_0002:Bernd.html](http://www.informatik.uni-trier.de/~ley/pers/hd/f/Fischer_0002:Bernd.html).

**Software available:**

1. ConceptCloud (<http://conceptcloud.herokuapp.com/>)
2. CSeq (<http://users.ecs.soton.ac.uk/gp4/cseq.html>)
3. ESBMC ([www.esbmc.org](http://www.esbmc.org))
4. AutoBayes (<http://ti.arc.nasa.gov/opensource/projects/autobayes/>)

**Grants:**

STIAS Doctoral Scholarship (for G. Greene), 2015–2017. ZAR 400,000.

*Browsing Semi-Structured Data with Tag Clouds over Concept Lattices*, NRF Competitive Programme for Rated Researchers, 2015–2017. ZAR 815,000.

NRF Incentive Funding for Rated Researchers, 2014–2022. ZAR 600,000.

Small Equipment Grant, Stellenbosch University, 2014. ZAR R125,624.

NRF Knowledge Interchange and Collaboration Grant, 2013. ZAR 20,000.

*Model-based Fault Localisation and Repair of Multi-Threaded Software using Software Model Checking*, Stellenbosch University, 2013-15. ZAR 140,000.

*SMT-Based Bounded Model Checking Timed LTL Properties for Embedded Software*, Royal Society International Exchange Grant, 2011-13. £11,600.

NASA Travel Grant, 2009. \$ 14,400.

*Customization and Adaptation of Automatically Generated Code*, EPSRC-grant EP/F052669/1, 2008-2011, £355,477 (FEC).

*NOTOS: New Algorithms for LTL model checking with satisfiability*, EPSRC-grant EP/E012973/1, 2006-2010, £208,613. Principal investigator from 02/2009.

*21st IEEE/ACM International Conference on Automated Software Engineering*. £1,640. Royal Society Travel Grant, 2006.

*A Software Safety Certification Plug-in for Automated Code Generators*. \$72,397. NASA Office of Safety and Mission Assurance Software Assurance Research Program funding, 2006. Proposal co-author and consultant.

Principal investigator and co-investigator on four NASA-funded projects, \$1,800,000 (1999-2006).

## Teaching:

Undergraduate level:

- *Software Engineering*, 3rd year, Stellenbosch, 2014–2016. Module leader. Module rating 74.5%, Lecturer rating 83.3%.
- *Software Engineering*, 1st year, U Southampton, 2008, 2010–2012. Lecturer rating 3.9/5.
- *Algorithms and Programming*, 1st year, U Southampton, 2007. Lecturer rating 3.7/5.
- *Compiler Engineering*, 2nd year, U Southampton, 2007–2012. Module leader. Module rating 3.8/5. Lecturer rating 4.3/5.

Graduate level:

- *Compiler and Software Language Engineering*, Stellenbosch, 2015.
- *From Aspect-Oriented Programming to Program Generation*, U Southampton, 2009–2012. Module leader. Module rating 4.5/5. Lecturer rating 4.8/5.
- *Unification Theory*, TU Braunschweig, 1994.

## Students:

PhD Students (graduated):

1. J. Morse (2010-2015), *Model Checking Embedded Systems Software* (external examiner: M. Tautschnig, Queen Mary University London, UK). First position at University of Bristol, UK
2. T. Aubrey-Jones (2010-2015), *Synthesizing Imperative Distributed-Memory Implementations from Functional Data-Parallel Programs* (external examiner: S.-B. Scholz, Herriot-Watt, UK). First position at BluePoint, Southampton, UK
3. S. Ghotbi (2008-2014), *A declarative and fine-grained policy language for the web application domain* (external examiner: K. Lano, King's College, UK). First position at Prince of Songkla University, Thailand.
4. N. Grech (2009-2013), *Preemptive Type Checking in Dynamically Typed Programs* (external examiner: G. Bierman, Microsoft, UK). First position at University of Bristol, UK.
5. L. Cordeiro (2008-2011), *SMT-Based Bounded Model Checking of Multi-threaded Software in Embedded Systems* (external examiner: R. Majumdar, Max Planck Institute, Germany). First position at University Federal de Amazonia, Manaus, Brazil.
6. N. Basir (2007-2010), *Safety Cases for the Formal Verification of Automatically Generated Code* (external examiner: Tim Kelly, U York, UK). First position at Universiti Sains Islam, Malaysia.

PhD Students (current):

7. G. Greene (2014), *Using Concept Lattices to Analyze Software Repositories*.
8. G. Birch (2012), *Automated Program Debugging via Model Checking*.
9. M. Tian (2010), *Adaptation and Customization of Automatically Generated Code*.

PhD Students (examined):

10. M. Mews, Technische Universität Berlin, Germany, 2016. *Kontrollflusssensible Quelltext-Umgestaltung mithilfe von Constraint-Programmierung*.
11. N. Rosner, University of Buenos Aires, Argentina, 2015. *Técnicas distribuidas para verificación acotada eficiente*.
12. I. Maamria, University of Southampton, UK, 2012. *Towards a Practically Extensible Event-B Methodology*.
13. K. Olmos, Utrecht University, The Netherlands, 2009. *Strategies for Context Sensitive Program Transformation*.
14. A. Saabas, Tallinn Technical University, Estonia, 2008. *Logics for Low-Level Code and Proof-Preserving Program Transformations*.
15. S. Lämmermann, KTH Stockholm, Sweden, 2002. *Runtime Service Composition via Logic-Based Program Synthesis*.

MSc Research Students (current):

16. J. Breytenbach (2016-). Formula Caching for Bounded Model Checking.
17. M. Esterhuizen (2016-). Grammar-based Test Case Generation.

Project Students:

- 8 honours project theses, U Stellenbosch, 2013-.
- 12 MSc theses and 11 senior project theses, U Southampton, 2006-2013.
- 9 MSc theses and 13 senior project theses, TU Braunschweig, 1990-1998.
- 8 graduate summer research students, NASA Ames, 2000-2004.

### **Professional Activities:**

Member of steering committee, *Automated Software Engineering* (2007-),  
*Generative Programming and Component Engineering* (2011-).

Member, IFIP TC-2 Working Group 2.11, “Program Generation” (2004-).

Member of editorial board, *Science of Computer Programming – Software Section* (2015-).

General co-chair, *Intl. Colloquium on Theoretical Aspects of Computing* (2018),  
*Generative Programming and Component Engineering Conf.* (2016),  
*Intl. SPIN Workshop on Model Checking of Software* (2015),  
*Intl. Conf. Logic for Programming, Artificial Intelligence and Reasoning* (2013).

PC co-chair, *Generative Programming and Component Engineering Conf.* (2009),  
*Automated Software Engineering Conf.* (2007).

Guest Editor, *J. Object Technology*, Special Section on “Selected Papers of GPCE 2009” (2011),  
*J. Automated Software Engineering*, Special Issue on “Best Papers of ASE 2007” (2009),  
*J. Automated Reasoning*, Special Issues on “Empirically Successful Automated Reasoning” (2006).

Co-organizer / session organizer for *Automated Software Engineering Conf., Tool Demos* (2010, 2012), *International Conf. Software Engineering, Tool Demonstrations* (2009), *6th International Workshop on the Implementation of Logics* (2006), *Automated Software Engineering Conf., Doctoral Symposium* (2006), *ASE-20 Workshop on Software Certification Management* (2005), *CADE-20 Workshop on Empirically Successful Classical Automated Reasoning* (2005), *Intl. Society for Bayesian Analysis, World Meeting* (2004), *Intl. Workshop on Rule-Based Programming* (2002), *AAAI Spring Symp. on Logic-Based Program Synthesis* (2002), *33rd Symp. on the Interface of Computing Science and Statistics* (2001), *NIPS\*2000 Workshop on Software Support for Bayesian Analysis Systems*.

PC member for *Intl. Conf. Software Engineering* (2016–2018), *European Software Engineering Conf. / Symp. Foundations of Software Engineering* (2017), *Conf. Fundamental Approaches to Software Engineering* (2015–2017), *Intl. Conf. Current Trends in Theory and Practice of Computer Science* (2017), *Intl. Conf. Model-Driven Engineering and Software Development* (2015, 2017), *Automated Software Engineering Conf.* (1999–2003, 2005–2016), *TACAS Software Verification Competition* (2012–2016), *Conf. Formal Aspects of Component Software* (2016), *Intl. Conf. Software Language Engineering* (2016), *Intl. SPIN Workshop on Model Checking of Software* (2016), *Intl. Conf. Logic for Programming, Artificial Intelligence, and Reasoning* (2005–2006, 2015), *Conf. Generative Programming and Component Engineering* (2011, 2013, 2015), *Intl. Conf. Predictor Models in Software Engineering* (2012), *Conf. Model Driven Engineering Languages and Systems* (2008), *ACM Symposium on Partial Evaluation and Program Manipulation* (2007), *Intl. Conf. Software and Data Technologies* (2006–2007), *Intl. Conf. Algebraic Methodology and Software Technology* (2002), *ACM Symp. Software Reusability* (1999),  
23 workshops.

Reviewer for major journals and conferences, including *IEEE Trans. Software Engineering Methodology*, *J. Symbolic Computation*, *J. Automated Software Engineering*.

30 invited presentations at international institutions.

Reviewer for NSF (USA), NRF (South Africa), NSERC (Canada), NSC (Poland), ETAg (Estonia), Claude Leon Foundation (South Africa).

#### **Awards:**

Invited Speaker, *Intl. Workshop First-Order Theorem Proving (FTP)* (2007), *Estonian Summer School in Computer and System Science (ESSCaSS)*, 2006, *Generative Programming and Component Engineering* (2005).

Invited Panelist, *30th Intl. Conf. Automated Software Engineering* (2015), *21st Intl. Conf. Automated Software Engineering* (2006), *IJCAR Workshop on Empirically Successful First Order Reasoning* (2004), *14th Intl. Conf. Automated Software Engineering* (1999).

Bronze Medal for *ESBMC*, *TACAS Software Verification Competition* (2012, 2013).

Most Influential Paper Award, *27th Intl. Conf. Automated Software Engineering* (2012).

Distinguished Paper Award, *33rd Intl. Conf. Software Engineering* (2011).

Best Presentation Award, *29th Intl. Conf. Computer Safety, Reliability and Security* (2010).

Best Paper Award, *13th Intl. Conf. Automated Software Engineering* (1998).

RIACS Performance Award (2004).

**Personal:**

Citizenship: Germany, South African resident

Languages: German (native), English (native standard), French (high-school level), Afrikaans (basic)

**References:** on request.