1 Research Interests

The main goal of my research is *to enable programmers to build reliable software efficiently*. This goal manifests in different research interests:

- Automatic Systematic Software Testing
- Software Verification
- Automated Fault Localization and Bug Fixing
- Semantics of Concurrent Data Structures
- Compiler Construction for Secure Computations
- Automated Information Leak Detection and Explanation

2 Ongoing Projects & Collaborators

Systematic Testing. I am working on test generation approaches for concurrent software that explore systematically combinations of code and global data-flow. The goal is to provide coverage information as feedback on the effectiveness of spent testing effort. In particular, I am continuing my work on con2colic testing, a concurrency variant of directed testing.

Automated Fault Explanation. I am working on automated fault explanation techniques for erroneous executions of concurrent programs such that programmers can understand an observed bug faster. In particular, I am interested in decision procedure based approaches for fault explanation.

Automated Bug Fixing. I am working on an approach for automatically fixing bugs in concurrent software.

Concurrent Data Structures. I am working on new ways to specify the semantics of concurrent data structures. The goal is to precisely specify the guarantees that concurrent data structures provide in a manner that is intuitive to programmers and enables efficient implementations that scale well for scenarios with many computing cores.

Debugging Tools for Secure Computations. Closely related to my work on fault explanation, I am also working on automated explanation techniques for information leaks in secure multi-party computations. Furthermore, I am working on the synthesis of attack scenarios that exploit information leaks in secure multi-party computations. This research is a continuation of my work on CBMC-GC, the first C compiler for secure two-party computations.

Collaborators. My current collaborators include Azadeh Farzan (University of Toronto), Georg Weissenbacher (TU Vienna), Thomas Wies (New York University), Daniel Schwartz-Narbonne (Amazon), Christoph Kirsch (University of Salzburg), Ana Sokolova (University of Salzburg), Stefan Katzenbeisser (TU Darmstadt).

3 Education

Ph.D. in Computer Science (with distinction) TU Vienna (June 2013)
Thesis title: *Query-Based Test Case Generation* Supervisor: Prof. Dr. Helmut Veith

Master of Science in Computer Science (with distinction) TU München (Oct. 2004 – Nov. 2006)
Thesis title: *Description Languages for Malicious Software* Supervisor: Prof. Dr. Helmut Veith

Diploma in Computer Science (with distinction) Landshut Univ. of Applied Sciences (Oct. 2000 – Sept. 2004)
Thesis title: *Optimization and Porting of the Software and Human-Computer Interface of a Cockpit Procedure Trainer for its Application in Virtual Reality* Supervisor: Prof. Dr. Gudrun Schiedermeier

4 Professional Activities

Service

ACISP'16 PC member

CAV'15 Artifact Evaluation PC member

PETShop: The Workshop on Language Support for Privacy Enhancing Technologies (Berlin, 2013) Co-organizer

Grants

The Austrian Science Fund (FWF) granted me the Erwin Schroedinger Fellowship J3696-N26 "Systematic Testing of Concurrent Software". This fellowship finances a two-year PostDoc position at the University of Toronto in the group of Prof. Azadeh Farzan and a follow-up one-year PostDoc position at TU Vienna.

PostDoc Positions

University of Toronto since March 2015
I am PostDoc in the group of Prof. Azadeh Farzan. My position is funded by the above mentioned fellowship.

TU Vienna July 2013 – December 2014
I was PostDoc in the group of Prof. Helmut Veith.

Research Assistant Positions

TU Vienna, TU Darmstadt, TU München November 2006 – June 2013
I was Research Assistant in the group of Prof. Helmut Veith. The group moved from TU München to TU Darmstadt and finally to TU Vienna.

Internships

University of Toronto, Toronto, Canada June & September 2012
Research collaboration with Prof. Azadeh Farzan.

Simon Fraser University, Vancouver, Canada November 2008 – December 2008
Research collaboration with Prof. Dirk Beyer.

EADS Deutschland GmbH, Munich, Germany March 2003 – July 2003
Intern at MT641 "Simulation Facilities".

Critical Reach AG, Munich, Germany August 2001 – January 2002
Development of a converter for CAD models.

Conference and Journal (Sub-)Referee

2015:	CONCUR
2014:	CAV
2013:	CAV
2012:	FMCAD, CSL, Journal of Systems and Software
2011:	FMCAD, HVC, ICCCT, FMICS, TASE, Special Issue of the Journal of Symbolic Computation on Invariant Generation and Advanced Techniques for Reasoning about Loops
2010:	POPL, CAV, CSR
2009:	FMCAD, ICTAC, CAV
2008:	ISoLA

5 Publications

[1] A. Holzer, C. Schallhart, M. Tautschnig, and H. Veith. Closure Properties and Complexity of Rational Sets of Regular Languages. *Theor. Comput. Sci.*, 605:62–79, 2015.

[2] J. Bürdek, M. Lochau, S. Bauregger, A. Holzer, A. von Rhein, S. Apel, and D. Beyer. Facilitating Reuse in Multi-Goal Test-Suite Generation for Software Product Lines. In *FASE'15*, pages 84–99.

[3] A. Farzan, A. Holzer, and H. Veith. Perspectives on White-Box Testing: Coverage, Concurrency, and Concolic Execution. In *ICST'15*, pages 1–11.

[4] M. Franz, A. Holzer, S. Katzenbeisser, C. Schallhart, and H. Veith. CBMC-GC: An ANSI C Compiler for Secure Two-Party Computations. In *CC'14*, pages 244–249.

[5] A. Holzer, C. Schallhart, M. Tautschnig, and H. Veith. On the Structure and Complexity of Rational Sets of Regular Languages. In *FSTTCS'13*, pages 377–388.

[6] A. Farzan, A. Holzer, N. Razavi, and H. Veith. Con2colic Testing. In *ESEC/FSE'13*, pages 37–47.

[7] D. Pötzl and A. Holzer. Solving Constraints for Generational Search. In *TAP'13*, pages 197–213.

[8] D. Beyer, A. Holzer, M. Tautschnig, and H. Veith. Information Reuse for Multi-goal Reachability Analyses. In *ESOP'13*, pages 472–491.

[9] K. Hoder, A. Holzer, L. Kovács, and A. Voronkov. Vinter: A Vampire-Based Tool for Interpolation. In *APLAS'12*, pages 148–156.

[10] A. Holzer, M. Franz, S. Katzenbeisser, and H. Veith. Secure Two-Party Computations in ANSI C. In *CCS'12*, pages 772–783.

[11] N. Razavi, A. Farzan, and A. Holzer. Bounded-Interference Sequentialization for Testing Concurrent Programs. In *ISoLA'12*, pages 372–387.

[12] A. Holzer, D. Kroening, C. Schallhart, M. Tautschnig, and H. Veith. Proving Reachability Using FShell - (Competition Contribution). In *TACAS'12*, pages 538–541.

[13] A. Holzer, V. Januzaj, S. Kugele, B. Langer, C. Schallhart, M. Tautschnig, and H. Veith. Seamless Testing for Models and Code. In *FASE'11*, pages 278–293.

[14] A. Holzer, C. Schallhart, M. Tautschnig, and H. Veith. How did You Specify Your Test Suite? In *ASE '10*, pages 407–416.

[15] A. Holzer, M. Tautschnig, C. Schallhart, and H. Veith. An Introduction to Test Specification in FQL. In *HVC'10*, pages 9–22.

[16] A. Holzer, V. Januzaj, S. Kugele, and M. Tautschnig. Timely Time Estimates. In *ISoLA'10*, pages 33–46.

[17] A. Holzer, V. Januzaj, and S. Kugele. Towards Resource Consumption-Aware Programming. In *ICSEA'09*, pages 490–493.

[18] A. Holzer, C. Schallhart, M. Tautschnig, and H. Veith. Query-Driven Program Testing. In *VMCAI'09*, pages 151–166.

[19] A. Holzer, C. Schallhart, M. Tautschnig, and H. Veith. FShell: Systematic Test Case Generation for Dynamic Analysis and Measurement. In *CAV'08*, pages 209–213.

[20] A. Holzer, J. Kinder, and H. Veith. Using Verification Technology to Specify and Detect Malware. In *EUROCAST'07*, pages 497–504.

6 Tool Development

ConCREST
`http://www.forsyte.at/software/concrest/`
Concolic testing tool for concurrent software
(design and main developer, C++)

CBMC-GC
`http://www.forsyte.at/software/cbmc-gc/`
C compiler for secure two-party computations
(design and main developer, C++)

Vermeer
`https://github.com/wies/vermeer/`
Automated debugging tool for C
(contributing developer, OCaml & Python)

CPA/Tiger
`http://www.forsyte.at/software/cpatiger/`
Automated test generator for sequential C programs
(design and main developer, Java)

CPAchecker
`http://cpachecker.sosy-lab.org/`
Software model checker
(contributing developer, Java)

7 University Activities

Supervised Bachelor's Theses

Author	*Title*
Alexander Reznicek	*CrestLA — Lazy Annotation for Concolic Testing*
Simon Lindner	*Translation of UML Activity Diagrams to C Code*
Wolfgang Zwirchmayr	*Activity Testing*

Supervised Master's Theses

Author	*Title*
Martin Dobiasch	*Concolic Testing of Concurrent Software in the Context of Weak Memory Models*
Daniel Pötzl	*Achieving High Coverage and Finding Bugs in Sequential and Concurrent Software*
Stephan Krall	*An FQL-Backend with Support for Complex Data Structures*
Vanessa Struve	*Realization of LIMA and Implementation of a Translator for C*

Teaching Assistantships

Course	*Nr. of Appointed Terms*
Formal Methods in Computer Science	7
Computer Aided Verification	3
Complexity Theory	2
Foundations of Digital Systems	1
Software Validation (seminar)	3
Program and Model Analysis (lab course)	3
Foundations of Programming (lab course)	1

8 OTHER

Progr. Languages:	C/C++, Java, OCaml, Python, JavaScript
Verification Tools:	CPAchecker, CBMC, NuSMV, SPIN
Testing Tools:	CREST, KLEE, FShell
Operating Systems:	Linux, Windows, VxWorks
Version Control:	git, SVN

9 REFERENCES

Prof. Dr. Helmut Veith (TU Vienna)
e-mail: veith@forsyte.at phone: +43-1-58801-18441

Prof. Dr. Stefan Katzenbeisser (TU Darmstadt)
e-mail: katzenbeisser@seceng.informatik.tu-darmstadt.de phone: +49-6151-16-5016

Prof. Dr. Christoph Kirsch (University of Salzburg)
e-mail: ck@cs.uni-salzburg.at phone: +43-662-8044-6411