Victor Winter
Curriculum Vitae*

March 13, 2009

Current Position: Associate Professor
Department of Computer Science
University of Nebraska at Omaha

Contact Information:

Address: PKI 174C
1110 South 67^{th} Street
Omaha, Nebraska 68182

Homepage: `http://faculty.ist.unomaha.edu/winter/`

Email: vwinter@mail.unomaha.edu
Phone: (402) 554-2003

Education:

Degree	Area	Institution
Ph.D.	CS	University of New Mexico(1994)
M.S.	CS	University of New Mexico(1988)
B.S.	CS	University of New Mexico(1985)

*This document as well as most of my publications can be found in electronic form on my homepage.

Employment History

University of Nebraska at Omaha — **Omaha, Nebraska**

Associate Professor in the Department of Computer Science — 8/2005—present
Assistant Professor in the Department of Computer Science — 8/2001—8/2005

Sandia National Laboratories — **Albuquerque, New Mexico**

Principal Member of the Technical Staff (PMTS) in the High Assurance Software Systems Engineering Department — 9/1996—7/2001
Postdoctoral Position in the Intelligent Systems and Robotics Center — 8/95—9/96

Arizona State University — **Tempe, Arizona**

Adjunct Faculty — 8/99—12/99

Department of Computer Science, University of New Mexico — **Albuquerque, New Mexico**

Adjunct Faculty — 8/2000–5/2001
Adjunct Faculty — 8/1994–5/1995
Research Assistant — 8/1989-9/1990
Instructor — 1/1989–5/1989

Argonne National Laboratory — **Argonne, Illinois**

Special Term Appointment in the Department of Mathematics and Computer Science — 5/1995—8/1995
Thesis Parts Appointment in the Department of Mathematics and Computer Science — 6/1991—7/1994
Thesis Parts Appointment in the Department of Mathematics and Computer Science — 6/1990—8/1990

Chapman University — **Albuquerque, New Mexico**

Adjunct Faculty — 8/94–10/94

Summary of Teaching Experience

Courses Taught at the University of Nebraska at Omaha (UNO)

Course Title	01	02	03	04	05	06	07	08
Introduction to the Theory of Computation					F			S, F
Principles of Programming Languages	F	F	F	F	F	F	F	F
Special Topics in Program Transformation		S,F	S					
Automata, Computability, and Formal Languages				F	F	F		
Advanced Concepts in Programming Languages		S	S	S	S	S	S	S
Trusted Systems: Design, Analysis, and Development							F	
Type Systems Behind Programming Languages							S	
Rewriting and Program Transformation						S		

F = fall semester, S = spring semester

Courses Taught at Other Institutions

Course Title	Institution	Time Period
Software Engineering	University of New Mexico (UNM) Department of Computer Science	fall 2000 and spring 2001
Topics on Engineering of High Assurance Systems	Arizona State University (ASU) Department of Computer Science	fall 1999
Programming Languages	University of New Mexico (UNM) Department of Computer Science	spring 1995
Discrete Math	University of New Mexico (UNM) Department of Computer Science	spring 1995 and fall 1994
Artificial Intelligence	Chapman University	fall 1994
Introduction to Programming	University of New Mexico (UNM) Department of Computer Science	spring 1989

Grants

Source		Amount	Year
Sandia National Laboratories	Victor Winter (PI)	$\$70,000$	$2008-2009$
Sandia National Laboratories	Victor Winter (PI)	$\$65,000$	$2007-2008$
Sandia National Laboratories	Victor Winter (PI)	$\$44,000$	$2006-2007$
Sandia National Laboratories	Victor Winter (PI)	$\$25,000$	$2005-2006$
Sandia National Laboratories	Victor Winter (PI)	$\$20,000$	$2004-2005$
Sandia National Laboratories	Victor Winter (PI)	$\$80,000$	$2003-2004$
National Science Founctation (CCR-0313691)	Victor Winter (co-PI)	$\$2,128,000$	$2003-2007$
Sandia National Laboratories	Victor Winter	$\$55,000$	$2002-2003$
Sandia National Laboratories	Victor Winter	$\$61,000$	$2001-2002$
National Science Foundation (CCR-0209187)	Victor Winter	$\$90,000$	$2001-2003$

Additional comments relating to grant activity.

- Sandia National Laboratories. All oft the Sandia grants support research related to the development of the SCORE processor, a hardware implementation of the JVM developed at Sandia National Laboratories for use in high-consequence embedded systems.
- National Science Foundation (NSF)
  - Grant Number: CCR-0209187. Research related to this grant involves developing and enhancing the HATS transformation system.
  - Grant Number: 0313691. Burnham, B. W. (PI), Craiger, J. P. (co-PI), and Winter, V. (co-PI). Cyber Corp Scholarships for the University of Nebraska at Omaha Information Assurance Program. Award date: June 2003.

Additional Remarks

- Primary author of the BART Case Study. In his keynote address at ICSE 2000, Axel van Lamswerde stated that the BART case study was the benchmark for the formal methods community. In 2001, a week-long seminar on the BART case study was held at Dagstuhl (Seminar No. 01221). The topic of this seminar was "Can Formal Methods Cope with Software-Intensive Systems?". This seminar focused on the BART Case Study. In 2003, Fabrice Kordon and Michael Lemoine were the Editors of a book published by Kluwer Academic Press. This book is titled "Formal Methods for Embedded Distributed Systems: How to Master the Complexity" and is devoted entirely to the BART case study.
- I am the creator and primary developer of Transformation system called HATS. HATS provides an integrated development environment (IDE) for strategic programming. The HATS interface is written in Java and the execution engine is written in SML. The Java interface provides support for file management, editors for various file types including text highlighting of keywords for the strategic programming language as well as term highlighting. The interface also provides control over execution of various portions of the execution engine (i.e., the parser, the interpreter, and the prettyprinter), and supports graphical display of term structures.

  The HATS engine consists of three components: a parser, an interpreter, and a prettyprinter. A domain of discourse can be defined by defining a suitable grammar and lexer. The HATS parser supports an extended-BNF grammar and additionally supports precedence and associativity of operators and rules. The parser is an LR parser with the capability to do backtracking as needed in order to resolve local ambiguities. One can think of such a parser as an LR(K) parser for an arbitrary K. Backtracking brings the parser's capability close to that of a scannerless generalized LR parser.

  HATS is freely available and currently only on Windows systems.

Publications in Chronological Order

[1] V. Winter, G. Kniesel, H. Siy, and M. Zand. Making Aspect-Orientation Accessible through Syntax-based Language Composition. *IET Software*, TBA(TBA):TBA, 2009.

[2] D. Resler and V. Winter. A Higher-Order Strategy for Eliminating Common Subexpressions. *Journal of Computer Languages, Systems and Structures*, TBA(TBA):TBA, 2008.

[3] Eric Lindahl and Victor L. Winter. Pattern Matching Information Flow using GADT. In *Proceedings of the* 3^{rd} *International Conference on Information Warfare and Security (ICIW)*, pages 255–262. ACI, 2008.

[4] Fares Fraij and Victor L. Winter. A Suggested Generic Intelligent Tutoring Framework. In *Proceedings of the International Conference on Modelling and Simulation (MS)*, pages 178–182, 2008.

[5] Victor L. Winter, Azamat Mametjanov, S. E. Morrison, J. A. McCoy, and G. L. Wickstrom. Transformation-based Library Adaptation for Embedded Systems. In *Proceedings of the* 10^{th} *IEEE International Symposium on High Assurance Systems Engineering (HASE)*, 2007.

[6] Victor L. Winter and Azamat Mametjanov. Generative Programming Techniques for Java Library Migration. In *Proceedings of the Sixth International Conference on Generative Programming and Component Engineering (GPCE)*, October 2007.

[7] V. Winter, Harvey Siy, Mansour Zand, and Gunter Kniesel. A Prototype of a Generic Weaver. Technical Report UNO2007-10, Department of Computer Science, Univerity of Nebraska at Omaha, August 2007.

[8] V. L. Winter. Stack-based Strategic Control. In *Preproceedings of the Seventh International Workshop on Reduction Strategies in Rewriting and Programming*, June 2007.

[9] Harvey Siy, Prasanna Aryal, Victor Winter, and Mansour Zand. Aspectual Support for Specifying Requirements in Software Product Lines. In *Proceedings of Early Aspects Workshop in Aspect-Oriented Requirements Engineering and Architecture Design*, May 2007.

[10] Yan Wu, Mansour Zand, Harvey Siy, and Victor Winter. Construction of Ontology-Based Software Repositories by Text Mining. In *International Conference on Computational Science (ICCS)*, volume 4489 (Part III) of *LNCS*, 2007.

[11] William L. Sousan, Victor L. Winter, Mansour Zand, and Harvey Siy. ERTSAL: A Prototype of a Domain-Specific Aspect Language for Analysis for Embedded Real-Time Systems. In *Second Workshop on Domain Specific Aspect Languages (DSAL)*, 2007.

[12] Victor L. Winter and Deepak Kapur. Towards Dynamic Partitioning of States of a Reactive System: Train Controller Case Study. In Fabrice Kordon and Janos Sztipanovits, editors, *Reliable Systems on Unreliable Networked Platforms (*12^{th} *Monterey Workshop)*, volume 4322 of *LNCS*, pages 47–69, 2007.

[13] Victor L. Winter. Model-driven Transformation-based Generation of Java Stress Tests. *Electronic Notes in Theoretical Computer Science (ENTCS)*, 174(1):99–114, April 2007.

[14] Victor L. Winter, Christopher Scalzo, Arpit Jain, Brent Kucera, and Azamatbek Mametjanov. Comprehension of Generative Techniques. In *Software Transformation Systems Workshop (STS)*, 2006.

[15] Victor L. Winter, Harvey Siy, Mansour Zand, and Prasanna R. Aryal. Aspect Traceability through Invertible Weaving. In *Proceedings of Early Aspects Workshop: Traceability of Aspects in the Early Life Cycle*, March 2006.

[16] Victor L. Winter, Jason Beranek, Fares Fraij, Steve Roach, and Greg Wickstrom. A Transformational Perspective into the Core of an Abstract Class Loader for the SSP. *ACM Trans. on Embedded Computing Sys.*, 5(4):773–818, 2006.

[17] Victor L. Winter. Program Transformation: What, How, Why? In *Encyclopedia of Computer Science and Engineering*. Wiley and Sons, 2006.

[18] Victor L. Winter and Jason Beranek. Program Transformation Using HATS 1.84. In Ralf Lämmel, João Saraiva, and Joost Visser, editors, *Generative and Transformational Techniques in Software Engineering (GTTSE)*, volume 4143 of *LNCS*, pages 378–396, 2006.

[19] Victor L. Winter, Jason Beranek, Azamatbek Mametjanov, Fares Fraij, Steve Roach, and Greg Wickstrom. A Transformational Overview of the Core Functionality of an Abstract Class Loader for the SSP. In *Proceedings of the Tenth IEEE International Workshop on Object-oriented Real-time Dependable Systems (WORDS)*, 2005.

[20] Harvey Siy, Mansour Zand, and Victor Winter. The Role of Aspects in Domain Engineering. In *Aspects and Software Product Lines (An Early Aspects Workshop at SPLC-Europe)*, 2005.

[21] Victor L. Winter. Strategy Construction in the Higher-Order Framework of TL. *Electronic Notes in Theoretical Computer Science (ENTCS)*, 124:149–170, March 2005.

[22] Victor L. Winter, S. Roach, and F. Fraij. Higher-Order Strategic Programming: A Road to Software Assurance. In *The* 8^{th} *IASTED International Conference on Software Engineering and Applications (SEA)*, pages 350–355, 2004.

[23] Victor L. Winter and M. Subramaniam. The Transient Combinator, Higher-order Strategies, and the Distributed Data Problem. *Science of Computer Programming (Special Issue on Program Transformation)*, 52:165–212, 2004.

[24] G. L. Wickstrom, J. Davis, S. E. Morrison, S. Roach, and V. L. Winter. The SSP: An Example of High-Assurance System Engineering. In *Proceedings of the* 8^{th} *IEEE International Symposium on High Assurance Systems Engineering (HASE)*, pages 167–177, 2004.

[25] Victor L. Winter, F. Kordon, and M. Lemoine. *Formal Methods for Embedded Distributed Systems: How to master the complexity*, chapter The BART Case Study, pages 3–22. Kluwer Academic Publishers, 2004.

[26] V. Winter, D. Kapur, and G. Fuehrer. *Formal Methods for Embedded Distributed Systems: How to master the complexity*, chapter Formal Specification and Refinement of a Safe Train Control Function, pages 25–64. Kluwer Academic Publishers, 2004.

[27] V.L. Winter, S. Roach, and F. Fraij. Dependable Software through Higher-Order Strategic Programming. Technical Report SAND2004-0868, Sandia National Laboratories, March 2004.

[28] Victor Winter. Strategy Application, Observability, and the Choice Combinator. Technical Report SAND2004-0871, Sandia National Laboratories, March 2004.

[29] G. Wickstrom, V. L. Winter, J. Beranek, F. Fraij, and S. Roach. An Abstract Classloader for the SSP and its Implementation in TL. Technical Report SAND2004-3225, Sandia National Laboratories, August 2004.

[30] V. L. Winter, S. Roach, and G. Wickstrom. Transformation-Oriented Programming: A Development Methodology for High Assurance Software. In M. Zelkowitz, editor, *Advances in Computers: Highly Dependable Software*, volume 58, pages 47 – 116, 2003.

[31] V. Winter and M. Subramaniam. Higher-order Transformation and the Distributed Data Problem. Technical Report SAND2003-4587, Sandia National Laboratories, December 2003.

[32] A. Azadmanesh, V. Winter, and B. Ghahramani. Authenticated Consensus with Diagnosis. In *The* 34^{th} *Annual Symposium on Performance Evaluation of Computer and Telecommunication Systems (SCSC'02)*, 2002.

[33] V. L. Winter. Artificial Intelligence. In Flynn and Roger, editors, *Computer Sciences (Macmillan Science Library)*. Macmillan Reference USA, 2002.

[34] V. L. Winter. Virtual Reality. In Flynn and Roger, editors, *Computer Sciences (Macmillan Science Library)*. Macmillan Reference USA, 2002.

[35] V. L. Winter, D. Desovski, and B. Cukic. Virtual Environment Modeling for Requirements Validation of High Consequence Systems. In *Proceedings of the Fifth International Symposium on Requirements Engineering*, 2001.

[36] A. de Groot, J. Hooman, F. Kordon, E. Paviot-Adet, I. Mounier, M. Lemoine, G. Gaudiere, V. Winter, and D. Kapur. A Survey: Applying Formal Methods to a Software Intensive System. In *Proceedings of the* 6^{th} *International Symposium on High Assurance Systems Engineering (HASE)*, pages 55–64, 2001.

[37] D. Kapur, V. L. Winter, and R. S. Berg. Designing a Controller for a Multi-Train Multi-Track System. *Electronic Notes in Theoretical Computer Science*, 50(1):65–79, August 2001.

[38] Deepak Kapur and Victor Winter. On the Construction of a Domain Language for a Class of Reactive Systems. In Winter and Bhattacharya, editors, *High Integrity Software*, pages 169–196, 2001.

[39] Victor Winter, Deepak Kapur, and Ray Berg. A Refinement-based Approach for Developing Software Controllers for Train Systems. In Winter and Bhattacharya, editors, *High Integrity Software*, pages 197–240, 2001.

[40] Victor Winter and Tom Caudell. Using Virtual Reality to Validate System Models. In Winter and Bhattacharya, editors, *High Integrity Software*, pages 301–320, 2001.

[41] Victor Winter, Raymond Berg, and Jim Ringland. Bay Area Rapid Transit District Advance Automated Train Control System Case Study Description. In Winter and Bhattacharya, editors, *High Integrity Software*, pages 115–135, 2001.

[42] Farokh B. Bastani, I-Ling Yen, John Linn Kashi Rao, and Victor Winter. Design for Independent Composition and Evaluation of High-Confidence Embedded Software Systems. In *Proceedings of the Monterey Workshop*, 2001.

[43] V. L. Winter. Program Transformation in HATS. In *Proceedings of the Software Transformation Systems (STS) Workshop (part of ICSE)*, pages 26–32, 1999.

[44] Victor L. Winter. An Overview of HATS: A Language Independent High Assurance Transformation System. In *Proceedings of the IEEE Symposium on Application-Specific Systems and Software Engineering Technology (ASSET)*, pages 222–229, 1999.

[45] J. M. Boyle, R. D. Resler, and V. L. Winter. Do You Trust Your Compiler? *IEEE Computer*, 32(5):65–73, 1999.

[46] Victor Winter. A Synchronous Paradigm for Modeling Stable Reactive Systems. In *Proceedings of the* 4^{th} *International Symposium on High Assurance Systems Engineering (HASE)*, pages 257–265, 1999.

[47] F. B. Bastani, V. Winter, and I-Ling Yen. Dependability of Relational Safety-Critical Programs. In *Proceedings of the* 10^{th} *International Symposium on Software Reliability Engineering (ISSRE)*, pages 47–48, 1999. (fast abstract).

[48] V. L. Winter, J. M. Covan, and L. J. Dalton. Passive Safety in High Consequence Systems. *IEEE Computer*, 31(4):35–36, 1998.

[49] Victor L. Winter, R. S. Berg, and L. J. Dalton. Risk-based System Refinement. In *Proceedings of the* 16^{th} *International System Safety Conference*, pages 61–69, 1998.

[50] R. S. Berg and Victor L. Winter. Risk Assessment and Integrity in System Design. In *Proceedings of the* 4^{th} *International Conference on Engineering of Complex Computer Systems (ICECCS)*, pages 115–120, 1998.

[51] R. S. Berg and V. Winter. Use of Integrity Techniques and Risk Assessment in System Design. In *Proceedings of the IEEE Workshop on Application-Specific Software Engineering and Technology (ASSET)*, pages 60–62, Washington, DC, USA, 1998. IEEE Computer Society.

[52] V. L. Winter. Software Construction via Abstraction, Synthesis, and Transformation (AST). In *Proceedings of the IEEE High Integrity Software Conference*, pages 30–39, 1997.

[53] James M. Boyle, Terence J. Harmer, and Victor L. Winter. The TAMPR Program Transformation System: Simplifying the Development of Numerical Software. In Erland Arge, Are Magnus Bruaset, and Hans Petter Langtangen, editors, *Modern Software Tools for Scientific Computing*, pages 353–372. Birkhäuser Boston, Inc., 1997.

[54] V. L. Winter and J. M. Boyle. Proving Refinement Transformations Using Extended Denotational Semantics. In *Proceedings of the Durham Transformation Workshop*, 1996.

[55] J. M. Boyle, Terence J. Harmer, and V. L. Winter. The TAMPR Program Transformation System: Design and Applications. In *Proceedings of the Durham Transformation Workshop*, 1996.

[56] V. L. Winter and J. M. Boyle. Proving Refinement Transformations for Deriving High-Assurance Software. In *Proceedings of the IEEE High-Assurance Systems Engineering Workshop*, pages 68–77, 1996.

[57] V. L. Winter. Visualization and Animation as a Technique to Assist in the Construction of High Assurance Software. In *CADE Workshop on Visual Reasoning*, pages 11–17, 1996.

[58] Software with Partial Functions: Automating Correctness Proofs via Nonstrict Explicit Domains. In *CADE Workshop on Mechanization of Partial Functions*, 1996.

[59] V. L. Winter. *Proving the Correctness of Program Transformations.* PhD thesis, University of New Mexico, 1994.

[60] V. L. Winter, G. H. Chisholm, B. T. Smith, and A. J. Wojcik. A Formal Model for Verification of Abstract Properties. Technical Report ANL-92/10, Argonne National Laboratory, 1992.