December, 2006

Curriculum Vitae for David R. Musser

Personal Data

- **Name: David R. Musser**
- **Title:** Professor
- **Mailing Address:**

  Department of Computer Science
  Rensselaer Polytechnic Institute
  Troy, NY 12180-3590
  USA
- **Home page:** http://www.cs.rpi.edu/~musser
- **Email address:** musser@cs.rpi.edu

Education

Ph.D.	(Computer Sciences)	University of Wisconsin	1971
M.S.	(Computer Sciences)	University of Wisconsin	1970
M.A.	(Mathematics)	University of Wisconsin	1968
B.A.	(Mathematics)	Austin College	1966

Professional Experience

1988-present	Rensselaer Polytechnic Institute 1993-present Professor, Computer Science 1998-present Professor, Faculty of Information Technology Program 1988-93 Research Professor, Computer Science
2006	Visiting Researcher, Google, Inc., New York, NY (January-August, during sabbatical leave)
1998	Visiting Scientist, Technical University of Vienna (January-February, during sabbatical leave)
1998	Associate Professor, Tübingen University (March-July, during sabbatical leave)
1979-87	Computer Scientist, General Electric Research and Development Center, Schenectady, NY
1974-79	Research Staff Member, USC Information Sciences Institute
1973-74	(academic year) Visiting Assistant Professor, Computer Sciences Department and Research Staff Member, Mathematics Research Center, University of Wisconsin
1970-73	Assistant Professor, Computer Sciences Department, University of Texas
1967-70	Computer Programmer, Mathematics Research Center, University of Wisconsin (part-time)

Research Interests

Concept- and proof-based methods of software development

The goal of my research in this area is to provide a solid foundation for development and deployment of software for systems with stringent requirements for functional correctness, efficiency, security, and safety, The long range goal is to meet new challenges that arise from distributed software components, embedded system software updates, Web services, and other software for modern, pervasive computing.

A principal focus is generic programming methodology. Generic programming is largely an activity of "lifting" of specific computer code to a more widely useful level, while maintaining high standards of efficiency and other required properties. This process is aided by conceptual classification of software components according to rigorously specified requirements. Results include the C++ Standard Template Library (STL), which is based on joint research with colleagues in industry; new generic sorting and searching algorithms; and new algorithm concept taxonomies.

This work is 'concept-based,' where 'concept' is meant in the specific, technical sense of a set of abstractions, such as abstract data types or algorithms, whose membership is defined by a set of axioms. In this technical sense, concepts are the main abstraction and organization mechanism in generic programming. The key operation on concepts is *refinement*: incrementally adding requirements to a concept description, thereby reducing the number of abstractions it contains but at the same time enabling more efficient algorithms or more refined analysis to be applied to the abstractions that remain. Repeated refinement with different choices of additional requirements results in a *concept taxonomy* or *hierarchy*, such as the STL container concept taxonomy.

Such concept taxonomies can also serve as the foundation for a 'proof-based approach' to high assurance of software quality. By associating theorems and their proofs with each concept, we can systematically build and use libraries of theories (axioms + theorems). In the proofs we can use both the axioms and the other theorems of the concept, including those from ancestor concepts. A principal focus of current work is the 'proof-carrying code' approach, including extending it to higher level languages and exploring a new variant called 'code-carrying theory' in which code is extracted directly from a theory. There are major problems with mobile code, such as code downloaded from the Web or software updates to consumer products like cellphones. The code can be corrupted in a variety of ways, some originating at the code producer's end (such as insufficient testing by its developers, or infection with viruses), and others occurring during delivery (such as network hacking or undetected transmission errors). Such problems can cause considerable harm to the consumer's software and data, but the currently employed means of protecting against them are woefully inadequate. The key idea of proof-carrying code and code-carrying theory is to use a proof-checker, a small program that can be embedded in an operating system and is capable of checking mathematical proofs presented to it. A code consumer who receives new (or replacement) code simply doesn't install it unless it is accompanied by a proof that (1) is accepted by the proof-checker, and (2) proves the code meets certain requirements. A major component of the research involves learning how to state and prove requirements that, if met, are sufficient to protect against careless programmers, hackers, and network errors.

Earlier research

My earlier research included discovery of the fundamental proof method known as 'inductionless induction' or 'proof by consistency,' and its application to rigorous specification and verification of data abstractions. This work was part of the basis of the Affirm, Affirm-85, and Tecton program verification systems. My research on generic programming with Alexander Stepanov provided the foundation for the Standard Template Library (STL), now part of the C++ Standard Library.

Publications

Books

STL Tutorial and Reference Guide: C++ Programming with the Standard Template Library, Second Edition, Addison-Wesley, Reading, MA, 550 pages, 2001 (with G. Derge and A. Saini).

STL Tutorial and Reference Guide: C++ Programming with the Standard Template Library, Addison-Wesley, Reading, MA, 432 pages, 1996 (with A. Saini).

The C++ Standard Template Library, Prentice-Hall, Upper Saddle River, NJ, 484 pages, 2000 (with P. Plauger, A. Stepanov, and M. Lee).

Generic Programming—International Seminar on Generic Programming, Dagstuhl Castle, Saarbrucken, Germany, April/May 1998, Selected Papers, co-editor, Lecture Notes in Computer Science, Vol. 1766, Springer-Verlag, Heidelberg, 268 pages, 2000 (with M. Jazayeri and R. Loos, co-editors).

The Ada Generic Library: Linear List Processing Packages, Springer-Verlag, New York, 278 pages, 1989 (with A. Stepanov).

Refereed Book Chapters

"The Tecton Proof System," chapter in *Formal Methods in Databases and Software Engineering*, V. Alagar, L. Lakshmanan, F. Sadri, eds., *Workshop in Computing Series*, Springer-Verlag, London, 1993 (with D. Kapur and X. Nie).

"Automated Theorem Proving for Analysis and Synthesis of Computations," chapter in *Current Trends in Hardware Verification and Automated Theorem Proving*, G. Birtwistle and P. Subrahmanyam, eds., Springer-Verlag, New York, 1989.

"BIDS: A Method of Specifying and Verifying Bidirectional Hardware Devices," chapter in *VLSI Specification, Verification, and Synthesis*, G. Birtwistle and P. Subrahmanyam, eds., Kluwer Academic Publishers, Boston, 1988 (with P. Narendran and W. Premerlani).

"Rewrite Rule Theory and Abstract Data Type Analysis," chapter in *Computer Algebra, EUROSAM*, 1982, *Lecture Notes in Computer Science*, Vol. 144, J. Calmet, ed., Springer-Verlag, Berlin, April, 1982, pp. 77-90 (with D. Kapur).

"Tecton: A Language for Manipulating Generic Objects," *Proc. of Program Specification Workshop*, J. Staunstrup, ed., Aarhus, Denmark, August 1981, chapter in *Lecture Notes in Computer Science*, Vol. 134, Springer-Verlag, Berlin, 1982 (with D. Kapur and A. Stepanov).

"The Design of Data Type Specifications," chapter in *Current Trends in Programming Methodology, Vol. IV*, R.T. Yeh, ed., Prentice-Hall, Englewood Cliffs, NJ, 1978 (with J. Guttag and E. Horowitz).

Refereed Journal and Proceedings Articles

"Generic programming and high-performance libraries," *International Journal of Parallel Programming*, 33(2-3):145-164, June 2005 (with D. Gregor, J. Järvi, M. Kulkarni, A. Lumsdaine, and S. Schupp).

"Design patterns for library optimizations," *Scientific Programming*, 11(4):309–320, 2003 (with D. P. Gregor and S. Schupp).

"Concept Use or Concept Refinement: An Important Distinction in Building Generic Specifications," *Proc. 4th International Conference on Formal Engineering Methods*, Shanghai, China, LNCS, Vol. 2495, Springer-Verlag, pp. 132 ff, 2002 (with Z. Shao).

"Complete Traversals as General Iteration Patterns," *Proc. IFIP TC2/WG2.1 Working Conference on Generic Programming*, pp. 187–206, Kluwer, Jul. 11-12, 2002, Dagstuhl Castle, Saarbrucken, Germany (with W. Klostermeyer and A. Sánchez-Ruíz).

"Semantic and Behavioral Library Transformations," *Information and Software Technology*, 44:797–810 (October 2002) (with S. Schupp, D. P. Gregor, and S.-M. Liu).

"Base class injection," in J. Bosch, editor, *Proc. of the Third Intern. Conf. on Generative and Component-Based Software Engineering,* LNCS 2186, pp. 106–117, Springer-Verlag, Berlin, 2001 (with D. P. Gregor and S. Schupp).

"Library transformations," in Mark Hamann, editor, *Proc. IEEE International Conference on Source Code Analysis and Manipulation, Florence*, pages 109–121, 2001 (with S. Schupp, D. P. Gregor, and S.-M. Liu).

"User-extensible Simplification—Type-based Optimizer Generators," *International Conference on Compiler Construction,* Lecture Notes in Computer Science, 2001 (with S. Schupp, D. Gregor, and S.-M. Liu).

"Complete Traversals and Their Implementation Using the Standard Template Library," *CLEI Electronic Journal*, `http://www.dcc.uchile.cl/~rbaeza/clei/`, Vol. 1, Number 2, December 1998 (Special Issue of Best Papers presented at CLEI'97, Valparaìso, Chile) (with E. Gamess and A. Sánchez-Ruíz).

"Generic Gram-Schmidt Orthogonalization by Exact Division," *ISSAC Proceedings*, July, 1997 (with Ulfar Erlingsson and Erich Kaltofen).

"Introspective Sorting and Selection Algorithms," *Software—Practice and Experience*, Vol. 27(8), 983–993 (August 1997).

"Dynamic Verification of C++ Generic Algorithms," *IEEE Transactions on Software Engineering*, Vol. 23, No. 5, May, 1997 (with C. Wang).

"Dynamic Verification of C++ Generic Components: A Practical Method and Its Support System," *Proc. Formal Methods in Software Practice*, San Diego, CA, January 10-11, 1996 (with C. Wang).

"An Overview of the Tecton Proof System," *Theoretical Computer Science*, Vol. 133, pp. 307–339, October 24, 1994 (with D. Kapur and X. Nie).

"Algorithm-Oriented Generic Libraries," *Software—Practice and Experience*, Vol. 24, No. 7, pp. 623–642, July 1994. (with A. Stepanov).

"Semi-Unification," *Theoretical Computer Science*, Vol. 81, pp. 169–187, April 30, 1991 (with D. Kapur, P. Narendran, and J. Stillman).

"Generic Programming," *Proc. 1988 International Symposium on Symbolic and Algebraic Computation*, P. Gianni, ed., Rome, Italy, July, 1988, *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1989 (with A. Stepanov).

"Only Prime Superpositions Need Be Considered in the Knuth-Bendix Completion Procedure," *J. of Symbolic Computation*,, Vol. 6, pp. 19–36, August 1988 (with D. Kapur and P. Narendran).

"Proof by Consistency," *Artificial Intelligence*, Vol. 31, pp. 125–157, 1987 (with D. Kapur).

"A Library of Generic Algorithms in Ada," *Proc. ACM SIGAda Conference*, Boston, December 9-11, 1987 (with A. Stepanov).

"Inductive Reasoning with Incomplete Specifications," *Proc. IEEE Symposium on Logic in Computer Science*, June, 1986 (with D. Kapur).

"Reasoning about Three Dimensional Space," *Proc. IEEE Robotics Conference*, St. Louis, March 1985 (with D. Kapur, J.L. Mundy, and P. Narendran).

"An Application of the AFFIRM System to Digital Flight Control System Design," *Proc. of the Fifth GE Software Engineering Conference*, Datona Beach, FL, April 1983; also selected for presentation at *Joint GE/Hitachi Symposium on Software Engineering*, Scottsdale, AZ, October 30-31 1984 (with R.W. Szcech).

"A Generalized Class of Polynomials That are Hard to Factor," *SIAM Journal on Computing*, Vol. 12, No. 3, August 1983 (with E. Kaltofen and B.D. Saunders).

"On Proving Uniform Termination and Restricted Termination of Rewriting Systems," *SIAM Journal on Computing*, Vol. 12, No. 1, February 1983 (with J. Guttag and D. Kapur).

"Derived Pairs, Overlap Closures, and Rewrite Dominoes: New Tools for Analyzing Term Rewriting Systems," *Proc. of ICALP Conference*, Aarhus, Denmark, July 1982 (with J. Guttag and D. Kapur).

"Operators and Algebraic Structures," *Proc. of Conference on Functional Programming Languages and Computer Architecture*, Portsmouth, New Hampshire, October 1981 (with D. Kapur and A. Stepanov).

"The AFFIRM Theorem Prover: Proof Forests and Management of Large Proofs," *Proc. Fifth Conference on Automated Deduction*, W. Bibel and R. Kowalski, eds., Les Arc, France *Lecture Notes in Computer Science*, Vol. 87, Springer-Verlag, Berlin, 1980 (with R.W. Erickson).

"An Overview of AFFIRM: A Specification and Verification System," *Proc. of IFIP Congress 80 (Eighth World Computer Congress)*, S. H. Lavington, ed., Melbourne and Tokyo, October 1980 (with S.L. Gerhart, D.H. Thompson, D.A. Baker, R.L. Bates, R.W. Erickson, R.L. London, D.G. Taylor, and D.S. Wile).

"Abstract Data Type Specification in the AFFIRM System," *IEEE Trans. on Software Engineering*, Vol. SE-6, No. 1, January 1980.

"On Proving Inductive Properties of Abstract Data Types," *Proc. of the Seventh ACM Symposium on Principles of Programming Languages*, Las Vegas, Nevada, January 1980.

"Abstract Data Types and Software Validation," *Comm. ACM* 21, pp. 1049–1064, December 1978 (with J. Guttag and E. Horowitz).

"On the Efficiency of a Polynomial Irreducibility Test," *Journal of the ACM*, Vol. 25, No. 2, pp. 271–282, April 1978.

"A Data Type Verification System Based on Rewrite Rules," *Proc. of Sixth Texas Conference on Computing Systems*, Austin, Texas, Section 1A, pp. 22–31, November 1977.

"Analysis of the Pope-Stein Division Algorithm," *Information Processing Letters*, Vol. 6, No. 2, pp. 151–155, October 1977 (with G.E. Collins).

"Some Extensions to Algebraic Specifications," *Proc. of Conference on Language Design for Reliable Software*, D. B. Wortman, ed., Raleigh, North Carolina, pp. 63–67, March 1977 (with J. Guttag and E. Horowitz).

"Multivariate Polynomial Factorization," *Journal of the ACM*, Vol. 22, No. 2, pp. 291-308, April 1975.

"The Application of a Symbolic Mathematical System to Program Verification," *Proc. of ACM Annual Conference*, San Diego, CA, November 1974 (with R.L. London).

"New Directions in Teaching the Fundamentals of Computer Science–Discrete Structures and Computational Analysis," *Proc. of the Third SIGCSE Symposium*, Columbus, Ohio, February 1973 (with R.T. Yeh and D.I. Good).

Other publications

"A Metaprogramming Approach to Aspect Oriented Programming in C++," *MPOOL 04 Workshop, ECOOP 2004 Conference*, Oslo, Norway, June 2004 (with S. Sunder).

Concept-Based Component Libraries and Optimizing Compilers, RPI Computer Science Technical Report 02-2, Troy, NY, January, 2002, also Proc. of Next Generation Software Workshop, International Parallel and Distributed Processing Symposium, April 2002 (with S. Schupp, D. P. Gregor, B. Osman, J. G. Siek, L.-Q. Lee, and A. Lumsdaine).

Algebraic Concepts Represented in C++, RPI Computer Science Technical Report 01-08, Troy, NY, January, 2001 (with S. Schupp and D. Gregor).

A Fast Generic Sequence Matching Algorithm, RPI Computer Science Technical Report 97-11 (without appendices) 1997; full version available at `http://www.cs.rpi.edu/~musser/gp/gensearch1.pdf` (revised Feb. 2, 2001) (with Gor V. Nishanov).

Depth-Limited Partitioning in Sorting and Selection Algorithms, RPI Computer Science Technical Report 96-11, Troy, NY, May, 1996.

Rationale for Adding Hash Tables to the C++ Standard Template Library, RPI Computer Science Technical Report 95-8, Troy, NY, April, 1995.

Hashtables for the Standard Template Library, Hewlett-Packard Report, Palo Alto, CA, January 29, 1995, revised February 20, 1995 (with J. Barreiro and R. Fraley)

A Basis for Formal Specification and Verification of Generic Algorithms in the C++ Standard Template Library, RPI Computer Science Department Technical Report 95-1, Troy, NY, January 1995, (with C. Wang).

EnDoc: Enhanced Documentation of Software Libraries Using Hypermedia, RPI Computer Science Department Technical Report 94-21, Troy, NY, October 1994 (with R. Cook, Jr. and K. Zalewski).

Block Shuffling for Improved Read Performance in a Loge Disk Controller, RPI Computer Science Department Technical Report 92-21, Troy, NY, July 1992 (with N. Schimke).

Tecton: A Framework for Specifying and Verifying Generic System Components, RPI Computer Science Department Technical Report 92-20, Troy, NY, July 1992 (with D. Kapur).

Algorithm-Oriented Generic Software Library Development, RPI Computer Science Department Technical Report 92-13, Troy, NY, July 1992 (with A. Stepanov).

Block Shuffling in a Loge Disk-Controller, Hewlett-Packard Laboratories Technical Report HPL-CSP-91-18, Palo Alto, CA, July 31, 1991.

"The Tecton Proof System, System Abstract," in *Proc. International Conference on Rewriting Techniques and Applications*, R. Book, ed., Como, Italy, April 10-12, 1991 (with R. Agarwal, D. Kapur and X. Nie).

An Ada Generic Data Structure Test Package, RPI Computer Science Department Technical Report 90-22, Troy, NY, July 1990 (with T. Quinn).

Elements of a Pragmatic Approach to Program Verification, RPI Computer Science Department, Technical Report 89-24, Troy, NY, Dec. 1989.

A Prototyping Language for Rapid Reuse, Hewlett-Packard Laboratories Technical Report STL-89-10, Palo Alto, CA, June 1989 (with D. Kapur, W. Olthoff, A. Snyder, A. Stepanov, B. Szymanski).

Report on the HOL (Higher Order Logic) Proof Checker, Final Report for NASA-sponsored Hardware Verification Assessment Project, RPI Computer Science Department, Troy, NY, November 1989.

Higher Order Imperative Programming, RPI Computer Science Department Technical Report 88-10, Troy, NY, April 1988 (with A. Kershenbaum and A. Stepanov).

Ada Generic Library Linear Data Structure Packages, Volume Two, General Electric Corporate Research and Development Technical Report 88CRD113, Schenectady, NY, April 1988.

Ada Generic Library Linear Data Structure Packages, Volume One, General Electric Corporate Research and Development Technical Report 88CRD112, Schenectady, NY, April 1988.

"Critique of Enhanced HDM," in *Verification Assessment Study Final Report, Vol. V: The Enhanced HDM System*, R. Kemmerer, ed., National Computer Security Center, Baltimore, MD, pp. 163–174, March 27, 1986.

"Critique of the FDM,"in *Verification Assessment Study Final Report, Vol. IV: The FDM System*, R. Kemmerer, ed., National Computer Security Center, Baltimore, MD, pp. 151–164, March 27, 1986.

"Introduction to the Affirm Verification System," in *Verification Assessment Study Final Report, Vol. III: The Affirm Verification System*, R. Kemmerer, ed., National Computer Security Center, pp. 1–12, March 27, 1986.

"Critique of the Gypsy Verification Environment," in *Verification Assessment Study Final Report, Vol. II: The Gypsy System*, R. Kemmerer, ed., National Computer Security Center, Baltimore, MD, pp. 75–84, March 27, 1986.

"Aids to Hierarchical Specification Structuring and Reusing Theorems in AFFIRM-85," *Proc. of Verification Workshop III*, Pahara Dunes, Calif., February 1985, also in *Software Engineering News*, August 1985.

Proceedings of a Workshop on the Rewrite Rule Laboratory, General Electric Report No. 84GEN008, 348 pages, Schenectady, NY, April 1984 (Co-editor with J. Guttag and D. Kapur)

On Semideciding First Order Validity and Invalidity, General Electric Report No. 84CRD042, Schenectady, NY, April 1984 (with D.S. Lankford).

A Proof Rule for Functions, USC Information Sciences Institute Technical Report No. ISI/RR-77-72, Marina del Rey, CA, October 1977.

Analysis of a Degree Compatibility Test for Polynomial Irreducibility, Mathematics Research Center (University of Wisconsin) Technical Summary Report, Madison, WI, May 1974.

The SAC-1 Polynomial Factorization System, University of Wisconsin Computer Sciences Department Technical Report No. 157, Madison, WI, March 1972 (with G.E. Collins).

Algorithms for Polynomial Factorization, University of Wisconsin Computer Sciences Department Technical Report No. 134 (Ph.D. Thesis), Madison, WI, September 1971.

The SAC-1 Modular Arithmetic System, University of Wisconsin Computing Center Technical Report No. 10, Madison, WI, June 1969 (with G.E. Collins, L.E. Heindel, E. Horowitz, and M.T. McClellan).

Professional and Public Lectures[1]

Invited Lectures at Conferences and Workshops

"Generic Programming and Formal Methods," *Verification Grand Challenge Workshop*, SRI International, Menlo Park, February 21-23, 2005.

"Expressing and Checking Semantic Concepts in a Formal Language and Proof System" *Workshop on Concepts: a Linguistic Foundation of Generic Programming*, Adobe Systems, San Jose, CA, April 29, 2004.

"Generic Programming in the STL and Beyond," Keynote Address, Adobe Tech Summit 2004, San Jose, CA, April 28, 2004.

"Generic Programming Concepts: STL and Beyond," DARPA Workshop on Program Composition for Embedded Systems, Portland, Oregon, July 8-10, 2002.

"Algorithm Concepts for Standard Libraries," Tübingen University on the occasion of the awarding of an honorary doctorate to Donald E. Knuth, Professor Emeritus, Stanford University, October 3, 2001.

"Dynamic Verification of C++ Generic Algorithms," Formal Methods and Automated Reasoning Day, Institute for Programming and Logics, University at Albany, October 14, 1996.

Colloquium Talks

"Generic Programming in the STL and Beyond" Computer Science Department, Union College, Schenectday, NY, October 5, 2004.

"Toward Strong Guarantees of Algorithm Performance: A Concept-Based Approach," Computer Science Department, The University at Albany, Albany, New York, May 1, 2003.

"Designing Generic Algorithms and Data Structures," Computer and Information Science Department, University of Delaware, Newark, Delaware, September, 1998.

"The Tecton Concept Description Language," Artificial Intelligence Department, Ulm University, Ulm, Germany, June, 1998.

[1]Past 10 years, and not including presentations of contributed conference or workshop papers.

"Fast Generic Search Algorithms," Distributed Computing Department, Technical University of Vienna, Vienna, Austria, February, 1998.

"Generic Programming and the Standard Template Library," Computer Science Department, Centrale Univ. Venezuela, Caracas, Venezuela, July 18, 1996.

"Generic Programming and the Standard Template Library," University of Delaware Computer Science Department, Newark, DE, May 10, 1996.

Seminar Presentations

"Generic Programming in the STL and Beyond," BlackRock Solutions, Inc., New York, NY, July 7, 2006.

"Generic Programming in the STL and Beyond," Google, Inc., New York, NY, November 21, 2005.

"Formal Methods for Generic Libraries," Workshop on Software Libraries: Design and Evaluation, Dagstuhl Castle, Germany, March 9-11, 2005.

"Towards Generic Theorems and Proofs About Generic Algorithms," Theory Seminar, RPI, October 22, 2003.

"A Distributed Algorithms Concept Taxonomy," Open Systems Lab Seminar, Indiana University, August 14, 2003.

"Towards Generic Theorems and Proofs About Generic Algorithms," Open Systems Lab Seminar, Indiana University, August 13, 2003.

"The Ada Standard Generic Library," seminar at Tübingen University, Tübingen, Germany, four lectures, June, 1998.

"The Tecton Concept Description Language," Seminar at Tübingen University, Tübingen, Germany, ten lectures, March - May, 1998.

"Generic Programming and the Standard Template Library," Rensselaer ACM Student Chapter Meeting, October 28, 1996.