

Elaine J. Weyuker
phone: +1 732 549 8118
cell:: +1 908 612 4906
email: weyuker@gmail.com

Education

Ph.D., Computer Science, Rutgers University

Dissertation: Program Schemas with Semantic Restrictions

M.S.E., Computer & Information Sciences, Moore School Electrical Engineering, University of Pennsylvania

Thesis: A Question Answering System in the Growing Machine

B.A., Mathematics, Harpur College, State University of New York at Binghamton

Thesis: Integral Distances and Rational Angles

Experience

2013-present, Visiting Professor, Mälardalen University, Västerås, Sweden and Visiting Scholar, DIMACS, Rutgers University.

1996–2012, Distinguished Member of Technical Staff and AT&T Fellow, AT&T Labs, Florham Park, NJ.

1993–1996, Distinguished Member of Technical Staff, AT&T Bell Laboratories, Murray Hill, NJ.

1977–1995, Professor of Computer Science, Courant Institute of Mathematical Sciences, New York University.

1969–1975, Lecturer in Computer Science, Richmond College, The City University of New York.

1968–1969, Systems Engineer, I.B.M.

June 1968–August 1968, Instructor, N.S.F. Summer Institute in Computer Science, University of Pennsylvania

December 1967–August 1968, Designed and Implemented an information retrieval system for archaeological data, University of Pennsylvania Museum.

February 1966–August 1966, Programmer, Texaco, Inc.

Major Honors

- Member of the US National Academy of Engineering
- Fellow of the ACM
- Fellow of the IEEE

Major Awards

- ACM President's Award - June 2010
- ACM SIGSOFT Retrospective Impact Paper Awards - September 2009 For a highly influential paper that has continued to have impact on the field for more than 25 years.
- Anita Borg Institute, Technical Leadership Award for outstanding research and technical leadership - Oct 2008.
- ACM SIGSOFT Outstanding Research Award for deep and lasting contributions and impact to software engineering as a discipline - Sept 2007.
- IEEE Harlan D. Mills Award for long-standing, sustained, and meaningful contributions to the theory and practice of the information sciences - 2004
- AT&T Chairman's Diversity Award -2004

- Rutgers University 50th Anniversary Distinguished Alumni Award - 2003
- YWCA Woman of Achievement Award - 2001.
- AT&T Fellow - 2000
- AT&T Bell Labs Diversity Award - 1994

Patent

Tool for Predicting Fault-Prone Software Files. No. 8151123, 3 April 2012, US.

Professional Activities - Leadership Positions

- Chair ACM Women's Council (2004-2012)
- Member ACM Council (2008-2012)
- Member of the Rutgers University Graduate School Advisory Board. (2004-present)
- Member of the Executive Board of the Coalition to Diversify Computing (2003-present)
- Member of the Board of Directors of the Computing Research Association (2000-2005)
- Association for Computing Machinery (ACM) Fellow Selection Committee (1998 - 2003), Chair (2002-3).
- Secretary/Treasurer ACM SIGSOFT (1989-1992)
- Member of the Executive Committee of the IEEE Computer Society Technical Committee on Software Engineering (1984-1987, 1991-1995)
- Editorial Board Member: J. Empirical Software Engineering, J. Software and Systems, IEEE Trans. Software Engineering, IEEE Trans. Dependable and Secure Computing, IEEE Spectrum, ACM Trans Software Engineering and Methodology

Publications: Journals and Refereed Conference Proceedings

1. Identifying Fault-prone Files in Large Industrial Software Systems. in *Perspectives on Data Science for Software Engineering*, ed. T. Menzies, L. Williams and T. Zimmerman, published by Morgan Kaufmann, Cambridge, MA, 2016. (with T. Ostrand)
2. Experience Report: Automated system level regression test prioritization using multiple factors. *Proc. 27th Int. Symp. on Software Reliability Engineering (ISSRE 2016)*, Ottawa, Canada, October 2016. (with P. Strandberg, D. Sundmark, W. Afzal and T. Ostrand)
3. Transitioning Fault Prediction Models to a New Environment. *Proc. 12th European Dependable Computing Conference (EDCC 2016)*, Gothenburg, Sweden, September 2016. (with J. Derhag, T. Ostrand, and D. Sundmark)
4. Automated test generation using model checking: an industrial evaluation. *Int. Journal on Software Tools for Technology Transfer*, Springer, Nov. 2014. (with E. Enoiu, A. Čaušević, T. Ostrand, D. Sundmark, P. Pettersson)
5. The Limited Impact of Individual Developer Data on Software Defect Prediction. *Empirical Software Eng. Journal*, Vol 18, No 3, June 2013. (with T. Ostrand, and R. Bell)
6. On the use of calling structure information to help predict software fault proneness. *Empirical Software Eng. Journal*, Vol. 17, No. 4-5, Aug 2012. (with Y. Shin, T. Ostrand, and R. Bell)
7. Can File Level Characteristics Help Identify System Level Fault-Proneness? *Proc. Haifa Verification Conference 2011 (HVC11)*, Haifa, Israel, December 2011. (with T. Ostrand)

8. The Impact of Individual Developers on Software Defect Prediction. *Empirical Software Eng. Journal*, to appear. (with T. Ostrand and R. Bell)
9. SoftwareFaultPrediction. *AT&T Software Symposium* Middletown, NJ, October, 2011. (with T. Ostrand, R. Bell, A. Nocera, A. Gauld)
10. Does Measuring Code Change Improve Fault Prediction? *Proc. 7th International Conference on Predictive Models in Software Engineering (Promise2011)*, Banff, Canada, September 2011. (with T. Ostrand and R. Bell)
11. Empirical Software Engineering Research - The Good, The Bad, The Ugly. *Proc. of the IEEE International Symposium on Empirical Software Engineering and Measurement*, Banff, Canada, September 2011.
12. Replicate, Replicate, Replicate. *Proc. Third Internat'l Conf on Software Testing, Verification*, Banff, Canada, September 2011. (with T. Ostrand and R. Bell)
13. On the use of calling structure information to help predict software fault proneness. *Empirical Software Eng. Journal*, online version July 2011, to appear in print version, (with Y Shin, T. Ostrand and R. Bell)
14. Assessing the Impact of Using Fault-Prediction in Industry. *Proc Testing: Academic & Industrial Conference (TAIC 2011)*, Berlin, Germany, March 2011. (with T. Ostrand and R. Bell)
15. Software Testing Research and Software Engineering Education. *Workshop on the Future of Software Engineering and Research* Santa Fe, NM, November 2010. (with T. Ostrand)
16. Programmer-based Fault Prediction. *Proc. International Conference on Predictive Models (PROMISE10)*, Romania, September, 2010. (with T. Ostrand and R. Bell)
17. Generating Test Cases Using using a Performability Model, *Special Issue of IET Journal*, 2010. (with Alberto Avritzer, Rosa Maria Meri Ledmundo de Souza e Silvaa)
18. Methods and Opportunities for Rejuvenation in Aging Distributed. *Journal of Systems and Software*, vol 83, no 9, 2010, pp. 1568 - 1578 (with A. Avritzer and R. Cole)
19. Software Fault Prediction Tool. *Proc. Internat'l Symp on Software Testing and Analysis (ISSTA2010)*, Trento, Italy, July 2010. (with Tom Ostrand)
20. Comparing the Effectiveness of Several Modeling Methods for Fault Prediction *Empirical Software Eng. Journal*, June 2010. (with T. Ostrand and R. Bell)
21. We're Finding Most of the Bugs, but What Are We Missing? *Proc. Third Internat'l Conf on Software Testing, Verification and Validation (ICST2010)*, Paris, France, April 2010. (with T. Ostrand and R. Bell)
22. Monitoring for Security Intrusion using Performance Signatures *Proc. ACM/WOSP-SIPEW 2010*, Jan 2010. (with A. Avritzer, R. Cole, K. James and R. Tanikella)
23. The Automated Generation of Test Cases using an Extended Domain Based Reliability Model *Proc. Workshop on the Automation of Software Test (AST 2009)*. Vancouver, BC, Canada, May 2009 (with A. Avritzer)
24. Does Calling Structure Information Improve the Accuracy of Fault Prediction? *Proc. Sixth International Working Conference on Mining Software Repositories(MSR 2009)*. Vancouver, BC, Canada, May 2009 (with Y. Shin, T. Ostrand and R. Bell)
25. Methods and Opportunities for Rejuvenation in Aging Distributed Software Systems. *Proc. International Workshop on Software Aging and Rejuvenation*, Seattle, Washington, November, 2008. (with A. Avritzer and R. Cole)
26. Progress in Automated Software Defect Prediction. *Proc. Haifa International Conference*, Haifa, Israel October 2008. (with T. Ostrand)

27. Do Too Many Cooks Spoil the Broth? Using the Number of Developers to Enhance Defect Prediction Models. *Empirical Software Eng. Journal*, October 2008. (with T. Ostrand and R. Bell)
28. Comparing the Effectiveness of Testing Techniques. in *Formal Methods and Testing*, R. Hierons, J. Bowen, M. Harman, eds. Published as Lecture Notes in Computer Science 4949, Springer, 2008.
29. Comparing Methods to Identify Defect Reports in a Change Management Database. *Proc. Defects in Large Software Systems Workshop (DEFECTS08)*, Seattle, Washington, July 2008. (with T. Ostrand)
30. Comparing Negative Binomial and Recursive Partitioning Models for Fault Prediction. *Proc. Predictive Models in Software Engineering (PROMISE'08)*, Leipzig, Germany, May 2008. (with T. Ostrand)
31. What Can Fault Prediction Do For YOU? *Proc. Second International Conference on Tests and Proofs (TAP08)*, Prato, Italy, April 2008. Published as Lecture Notes in Computer Science 4966, Springer. (with T. Ostrand)
32. Software Engineering Research - From Cradle to Grave. *Proc. 6th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering*, Dubrovnik, Croatia, Sept 2007.
33. How to Measure Success of Software Prediction Models. *Proc. Fourth International Workshop on Software Quality Assurance*, Dubrovnik, Croatia, Sept 2007. (with T. Ostrand)
34. Automating Algorithms for the Identification of Fault-Prone Files. *Proc. ACM/International Symposium on Software Testing and Analysis (ISSTA07)*, London, England, July 2007. (with T. Ostrand and R. Bell)
35. Using Developer Information as a Factor for Fault Prediction. *Proc. IEEE/Third International Promise Workshop*, Minneapolis, May 2007. (with T. Ostrand and R. Bell)
36. Ensuring System Performance for Cluster and Single Server Systems. *Journal of Systems and Software*, Vol 80, No. 4, April, 2007. (with A. Avritzer and A. Bondi)
37. An Industrial Research Program in Software Fault Prediction. *Proc. Workshop on Software Testing - From Research to Practice*, Hamburg, Germany, March 2007. (with T. Ostrand)
38. Using Performance Signatures and Software Rejuvenation for Worm Mitigation in Tactical MANETs. *Proc. ACM Sixth International Workshop on Software and Performance (WOSP2007)*, Buenos Aires, Argentina, Feb 2007. (with A. Avritzer and R. Cole)
39. Experience Developing Software Using a Globally Distributed Workforce. *Proc. IEEE/First International Conference on Global Software Engineering (ICGSE06)*, Florianopolis, Brazil, October 2006. (with A. Avritzer and T. Ostrand)
40. Adapting a Fault Prediction Model to Allow Widespread Usage *Proc. IEEE/Second International Promise Workshop*, Philadelphia, September, 2006. (with T. Ostrand and R. Bell)
41. On the Automation of Software Fault Prediction *Proc. IEEE/Testing: Academic and Industrial Conference - Practice and Research Techniques (TAIC PART)*, Windsor, England, August 2006. (with T. Ostrand)
42. Looking for Bugs in All the Right Places. *Proc. ACM/International Symposium on Software Testing and Analysis (ISSTA2006)*, Portland, Maine, July 2006. (with R. Bell and T. Ostrand)
43. Performance Assurance via Software Rejuvenation: Monitoring, Statistics and Algorithms *Proc. IEEE Performance and Dependability Symposium (PDS)*, Philadelphia, PA, June 2006. (with A. Avritzer, A. Bondi, M. Grottke, and K. Trivedi)
44. Empirical Studies as a Basis for Technology Transfer. *Empirical Software Engineering Issues*. June 2006. Published as Lecture Notes in Computer Science 4336, Springer.

45. Locating Where Faults Will Be. *Proc. ACM/IEEE/CRA Richard Tapia Celebration of Diversity in Computing*, Albuquerque, NM, Oct 2005. (with T. Ostrand and R. Bell)
46. Optimizing the Performance of a Cluster of Hosts. *Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PacRim 2005)* Victoria, Canada, August 2005. (with A. Avritzer and A. Bondi)
47. A Different View of Fault Prediction *Proc. 29th IEEE Annual International Computer Software and Applications Conference (COMPSAC 2005)*, Edinburgh, Scotland, July, 2005. (with T. Ostrand and R. Bell)
48. Ensuring Stable Performance for Systems that Degrade. *Proc. ACM Fifth International Workshop on Software and Performance (WOSP2005)*, Palma, Spain, July 2005. (with A. Avritzer and A. Bondi)
49. Predicting the Location and Number of Faults in Large Software Systems *IEEE Trans. on Software Engineering*, Vol 31, No 4, April 2005. (with T. Ostrand and R. Bell)
50. The Role of Modeling in the Performance Testing of an E-Commerce Application. *IEEE Trans. on Software Engineering*, Vol 30, No 12, December 2004, pp.1072-1083. (with A. Avritzer)
51. Difficulties Encountered Doing Empirical Studies in an Industrial Environment *Proc. 15th IEEE International Symposium of Software Reliability Engineering* Saint-Malo, France, November 2004. (with T. Ostrand)
52. Challenges in Predicting the Location of Faults in Large Software Systems. *IEEE/Proc. Workshop on Predictive Software Models (PSM04)*, Chicago, September 2004. (with R. Bell and T. Ostrand)
53. Where the Bugs Are. *Proc. ACM/International Symposium on Software Testing and Analysis (ISSTA2004)*, Boston, MA, July 2004. (with T. Ostrand and R. Bell)
54. A Tool for Mining Defect-Tracking Systems to Predict Fault-Prone Files. *Proc. IEE International Workshop on Mining Software Repositories (MSR2004)*, Edinburgh, May 2004. (with T.J. Ostrand)
55. Using Static Analysis to Determine Where to Focus Dynamic Testing Effort. *Proc. IEE Second International Workshop on Dynamic Analysis (WODA 2004)*, Edinburgh, May 2004. (with T. Ostrand and R. Bell)
56. How to Judge Testing Progress. *Journal Information and Software Technology*, Vol.46, No.5, Elsevier, 2004, pp.323-328.
57. An AGENDA for Testing Relational Database Applications. *J. Software Testing, Verification and Reliability*, Vol14(10), 2004, pp.17-44. (with D. Chays, Y. Deng, P. Frankl, S. Dan, and F. Vokolos)
58. Estimating the CPU Utilization of a Rule-Based System. *Proc. ACM/Fourth International Workshop on Software and Performance (WOSP2004)*, Redwood City, CA, Jan 2004. (with A. Avritzer and J. Ros)
59. Identifying Fault-Prone Files in Large Software Systems. *Proc. 7th IASTED International Conf. on Software Engineering and Applications, (SEA2003)*, Marina del Rey, CA, Nov. 2003. (with T.J. Ostrand)
60. Prediction = Power. *Proc. 15th IFIP International Testing of Communicating Systems Conference, (TESTCOM 2003)*, Sophia, Antipolis, France, May, 2003, pp. 1-9.
61. Using Operational Distributions to Judge Testing Progress. *Proc. ACM/Eighteenth Symposium on Applied Computing (SAC2003)*, Melbourne, FL, March 2003.
62. Thinking Formally About Testing Without a Formal Specification. *Proc. FATES Workshop*, Brno, Czech Republic, August 2002, pp.1-10.
63. Software Performance Testing Based on Workload Characterization. *Proc. ACM/Third International Workshop on Software and Performance (WOSP2002)*, Rome, Italy, July 2002. (with A. Avritzer, J. Kondek, and D. Liu)

64. The Distribution of Faults in a Large Industrial Software System. *Proc. ACM/International Symposium on Software Testing and Analysis (ISSTA2002)*, Rome, Italy, July 2002, pp.55-64. (with T. Ostrand)
65. A Metric to Predict Software Scalability. *Proc. IEEE/Eighth International Symposium on Software Metrics (Metrics2002)*, Ottawa, Canada, June 2002, pp.152-158. (with A. Avritzer)
66. How to Decide if Your Software is Scalable. *Proc. of the AT&T Software Symposium*, Feb 2002. (with A. Avritzer)
67. A Metric For Predicting the Performance of an Application Under a Growing Workload. *IBM Systems Journal*, Vol 41, No1, pp.245-254, 2002. (with A. Avritzer)
68. Transitioning from Academia to Industrial Research. *Journal of Systems and Software*, Vol 59(2001)1-2, December 2001.
69. Difficulties Measuring Software Risk in an Industrial Environment. *Proc. IEEE/International Conference on Dependable Systems and Networks (DSN2001)*, Goteberg, Sweden, July 2001, pp. 15-24.
70. Confessions of a Lapsed Academic. *Software Testing and Quality Engineering*, Vol 3, No 3, May/June 2001.
71. Confessions of an Empirical Researcher, or What Some People Won't Do. *Proc. 5th International Conference on Empirical Assessment and Evaluation of Software Engineering (EASE2001)*, U. Keele, England, April, 2001. (with T.J. Ostrand)
72. Empirical Studies of a Prediction Model for Regression Test Selection. *IEEE Trans. on Software Engineering*, Vol 27, No 3, March 2001, pp.248-263.. (with M.J. Harrold, D. Rosenblum, and G. Rothermel)
73. Experience with Performance Testing of Software Systems - Issues, An Approach, and Case Study. *IEEE Trans. on Software Engineering*, Vol 26, No 12, December 2000, pp.1147-1156. (with F.I. Vokolos)
74. Testing Software to Detect and Reduce Risk. *Journal of Systems and Software*, Vol 53, No 3, Sept 15, 2000, pp. 275-286. (with P. Frankl)
75. A Framework for Testing Database Applications, *Proc. ACM/International Symposium on Software Testing and Analysis (ISSTA2000)*, Portland, Oregon, Aug 2000, pp147-157. (with D. Chays, S. Dan, P. Frankl, and F. Vokolos)
76. Updating Software Reliability using Conditional Probabilities. *Proc. of the Section on Physical and Engineering Sciences, American Statistical Association Proc. Joint Statistical Meeting, 1999*, July 2000, pp141-145. (with Tamraparni Dasu)
77. A Learning Environment for Software Testers at AT&T. *Proc. Workshop on Learning Software Organizations*, Oulu, Finland, June 2000, pp.47-53. (with T.J. Ostrand)
78. Issues in Interoperability and Performance Verification in a Multi-ORB Telecommunications Environment, *Proc. Workshop on Dependability of IP Applications, Platforms, and Networks*, New York, June 2000. (with C.J. Lin, A. Avritzer, and S-L Lo)
79. CORBA Interoperability Testing using Open Group VSORB Test Suites. *Proc. ATTEST2000*, Middletown, NJ, June 2000. (with C.J. Lin, A. Avritzer, and S-L Lo)
80. Clearing a Career Path for Software Testers. *IEEE/Software*, March/April 2000, pp.76-82. (with T. Ostrand, R. Prasad, J. Brophy)
81. Quality of Service Enforcement for Distributed Objects. *IEEE/Proc. on Software*, Vol 146, No 5, October, 1999, pp.232-244. (with A. Avritzer)

82. Predicting Project Risk from Architecture Reviews. *Proc. IEEE/Sixth International Symposium on Software Metrics (Metrics99)*, Boca Raton, Fla, Nov. 1999, pp.82-90.
83. Quality of Service Issues for Distributed Objects. *Proc. AT&T Software Symposium*, Middletown, NJ, Oct. 1999. (with A. Avritzer)
84. Metrics to Assess the Likelihood of Project Success Based on Architecture Reviews. *Empirical Software Eng. Journal*, Vol. 4, No. 3, Sept. 1999, pp.197-213. (with A. Avritzer)
85. How Can We Encourage Practitioners to Respond to Empirical Evidence? *Proc. Fifth IEEE Workshop on Empirical Studies of Software Maintenance (WESS99)*, Oxford, Eng., Sept. 1999. (with T. Ostrand)
86. Evaluation Techniques For Improving the Quality of Very Large Software Systems in a Cost-Effective Way, *Journal of Systems and Software*, Vol. 47, Nos.2/3, July 1999, pp.97-103.
87. What Has Software Testing Research Done For You Lately? And What Should It Do For You In the Future? *Proc. ATTEST99 Conf.*, Middletown, NJ, June 1999.
88. Certifying Software Testers, *Proc. First International Software Assurance Certification Conf. (ISACC99)*, Dulles, Va., March 1999.
89. Investigating Metrics for Architectural Assessment, *Proc. IEEE/Fifth International Symposium on Software Metrics (Metrics98)*, Bethesda, Md., Nov. 1998, pp.4-10. (with A. Avritzer)
90. Performance Testing of Software Systems, *Proc. ACM Workshop on Software and Performance (WOSP-98)*, Santa Fe, N.M., Oct. 1998, pp80-87. (with F. Vokolos)
91. Testing Issues for Component-Based Software - A Cautionary Tale, *IEEE/Software*, Sept-Oct, 1998, pp 48-53.
92. Experience Testing Very Large Software Systems, *Proc. Fifteenth International Conference on Testing Computer Software*, Wash, D.C., June 1998, pp. 127-133.
93. Facilitating the Enforcement of Quality of Service Objectives By Using Software Testing Artifacts, *Proc. of QualityWeek98*, May 1998. (with A. Avritzer)
94. Testing the Performance of the New InfoWorx Customer Host Gateway, *Proc. Services and Infrastructure Performance Symposium 1997 (SIPS-97)*, November 1997, pp. 122-131. (with R. Deblock, C. Murali, and F. Vokolos)
95. An Approach to Quality of Service Enforcement for Distributed Objects, *Proc. Services and Infrastructure Performance Symposium 1997 (SIPS-97)*, November 1997, pp. 90-97. (with A. Avritzer)
96. Automatically Generating Test Data From a Boolean Specification, in *Automating Specification-Based Software Testing*, Robert M. Poston, ed., IEEE Computer Society, 1997, pp. 75-85.
97. Analyzing Partition Testing Strategies, in *Automating Specification-Based Software Testing*, Robert M. Poston, ed., IEEE Computer Society, 1997, pp. 64-72.
98. Enforcing Quality of Service of Distributed Objects, *Proc. IEEE International Symposium on Software Reliability Engineering (ISSRE-97)*, November 1997, pp 390-401. (with A. Avritzer)
99. Lessons Learned from a Regression Testing Case Study, *Empirical Software Eng. Journal*, 1997, Vol 2, No 2, pp.188-191. (with D. Rosenblum)
100. Quality of Service Issues for Distributed Objects *Proc. of IEEE/AT&T Software Reliability Engineering Users Group Workshop*, June, 1997. (with A. Avritzer)
101. Re-estimation of Software Reliability After Maintenance, *Proc. of IEEE/ACM Nineteenth International Conference on Software Engineering (ICSE)*, May 1997, pp. 79-85. (with A. Podgurski)

102. Comments on "Toward a Framework for Software Measurement Validation", *IEEE Trans. on Software Engineering*, March, 1997, pp. 187-188. (with S. Morasco, L.C. Briand, V.R. Basili, and M.V. Zelkowitz)
103. Using Coverage Information to Predict the Cost-Effectiveness of Regression Testing Strategies, *IEEE Trans. on Software Engineering*, March, 1997, pp. 146-156. (with D. Rosenblum)
104. Monitoring Smoothly Degrading Systems for Increased Dependability, *Empirical Software Eng. Journal*, Vol.2, No1, 1997, pp.59-77. (with A. Avritzer)
105. Predicting the Cost-Effectiveness of Regression Testing Strategies, *Proc. ACM Foundations of Software Engineering Conf (FSE4)*, Oct 1996, pp.118-126. (with D. Rosenblum)
106. Reliability Testing of Rule-Based Systems, *IEEE Software*, Sept 1996, pp.76-82. (with A. Avritzer and J. Ros) (Also presented at IEEE International Symposium on Software Reliability Engineering (ISSRE-96), Oct 1996.)
107. Detecting Failed Processes Using Fault Signatures, *Proc. of International Computer Performance and Dependability Symposium (IPDS)*, Sept 1996, pp.302-311. (with A. Avritzer)
108. Deriving Workloads for Performance Testing, *Software – Practice and Experience*, Vol26, No6, June 1996, pp.613-633. (with A. Avritzer)
109. Testing a Rule-Based System, *Proc. of QualityWeek96*, May 1996. (with A. Avritzer)
110. Using Failure Cost Information for Testing and Reliability Assessment, *ACM Transactions on Software Engineering and Methodology*, Vol5, No2, April 1996, pp.87-98.
111. Reply to Some Critical Remarks on a Hierarchy of Fault-Detecting Abilities of Test Methods *IEEE Trans. on Software Engineering*, Vol21, No10, Oct 1995, pp.861-863. (with P. Frankl)
112. Using the Consequence of Failures for Testing and Reliability Assessment, *Proc. ACM Foundations of Software Engineering Conf (FSE3)*, Oct 1995, pp.81-91.
113. The Automatic Generation of Load Test Suites and the Assessment of the Resulting Software, *IEEE Trans. on Software Engineering*, vol21, No9, Sept 1995, pp.705-716. (with A. Avritzer)
114. Automatically Generating Load Test Suites and Assessing the Software's Reliability, *Proc. Quality Week 95*, June 1995. (with A. Avritzer)
115. Testing Issues for Software Reuse, *Proc. AT&T Symposium on Software Reuse*, May 1995.
116. Using Software Load Testing Algorithms for Hardware Testing, *Proc. AT&T Conference on Electronic Testing (ACET95)*, April 1995, pp. 313-318. (with A. Avritzer)
117. A Simplified Domain Testing Strategy, *ACM Transactions on Software Engineering and Methodology*, Vol. 3, No. 3, 1994, pp.254-270. (with B. Jeng)
118. Estimating the Software Reliability of Smoothly Degrading Systems, *Proc. IEEE Fifth International Symposium on Software Reliability Engineering (ISSRE-94)*, Nov 1994, pp.168-177. (with A. Avritzer)
119. Generating Test Suites for Software Load Testing, *Proc. ACM International Symposium on Software Testing and Analysis (ISSTA-94)*, Aug 1994, pp.44-57. (with A. Avritzer)
120. Automatically Generating Test Data From a Boolean Specification, *IEEE Trans. on Software Engineering*, Vol. 20, No. 5, May, 1994, pp. 353-363. (with T. Goradia and A. Singh)
121. Provable Improvements on Branch Testing, *IEEE Trans. on Software Engineering*, Vol. 19, No. 10, Oct 1993, pp. 962-975. (with P. Frankl)

122. More Experience with Data Flow Testing, *IEEE Trans. on Software Engineering*, Vol. 19, No. 9, Sept 1993, pp. 912-919.
123. Can We Measure Software Testing Effectiveness?, *Proc. of IEEE-CS International Software Metrics Symposium*, May 1993, pp. 100-107.
124. An Analytical Comparison of the Fault-detecting Ability of Data Flow Testing Techniques, *Proc. of IEEE/ACM Fifteenth International Conference on Software Engineering (ICSE)*, May 1993, pp. 415-424. (with P. Frankl)
125. A Formal Analysis of the Fault-detecting Ability of Testing Methods, *IEEE Trans. on Software Engineering*, Vol. 19, No. 3, March 1993, pp.202-213. (with P. Frankl)
126. Comparing Fault Detecting Ability of Testing Methods, *Proc. of ACM SIGSOFT 91, Conference on Software for Critical Systems*, New Orleans, La., Dec 1991, pp.77-91. (with P. Frankl)
127. Data Flow-Based Test Adequacy Analysis for Languages with Pointers, *Proc. of the Fourth ACM SIGSOFT Symposium on Software Testing, Analysis and Verification (TAV4)*, Victoria, B.C., Canada, Oct 1991, pp.74-86. (with T. Ostrand)
128. Comparison of Program Testing Strategies, *Proc. of the Fourth ACM SIGSOFT Symposium on Software Testing, Analysis and Verification (TAV4)*, Victoria, B.C., Canada, Oct 1991, pp. 1-10. (with S. Weiss and D. Hamlet)
129. Analyzing Partition Testing Strategies, *IEEE Trans. on Software Engineering*, Vol. 17, No. 7, July 1991, pp. 703-711, (with B. Jeng)
130. Fault Detection Using Data Flow Testing, *Proc. of the Eighth Annual Pacific Northwest Software Quality Conference*, Portland, Oregon, October 1990.
131. Experience With Data Flow Testing, *Proc. Seventh International Conference on Testing Computer Software*, San Francisco, Ca., June 1990, pp. 219-224. (with B. Jeng)
132. The Cost of Data Flow Testing: An Empirical Study, *IEEE Trans. on Software Engineering*, Vol. 16, No. 2, Feb 1990, pp. 121-128.
133. Some Observations on Partition Testing, *Proc. Third Symposium on Testing, Analysis, and Verification*, Key West, Florida, Dec 1989, pp. 38-47. (with B. Jeng)
134. What Will It Cost to Test My Software?, *Proc. of the Seventh Annual Pacific Northwest Software Quality Conference*, Portland, Oregon, September 1989, pp.349-360.
135. In Defense of Coverage Criteria, *Proc. of the 11th ACM/IEEE International Conf. on Software Engineering (ICSE)*, Pittsburgh, Pa, May 1989.
136. An Extended Domain-Based Model of Software Reliability, *IEEE Trans. on Software Engineering*, Vol. 14, No. 10, Oct 1988, pp1512-1524. (with S. Weiss)
137. An Applicable Family of Data Flow Testing Criteria, *IEEE Trans. on Software Engineering*, Vol. 14, No. 10, Oct 1988, pp1483-1498. (with P. Frankl)
138. Using Data Flow Analysis for Regression Testing, *Proc. of the Sixth Annual Pacific Northwest Software Quality Conference*, Portland, Oregon, September 1988, pp233-247. (with T. Ostrand)
139. Evaluating Software Complexity Measures, *IEEE Trans. on Software Engineering*, Vol. 14, No. 9, Sept 1988, pp.1357-1365.
140. An Empirical Study of the Complexity of Data Flow Testing, *Proc. of the Second ACM Workshop on Software Testing, Verification, and Analysis (TAV)*, Banff, Canada, July 1988, pp.188-195.

141. The Evaluation of Program-Based Software Test Data Adequacy Criteria, *Communications of the ACM*, June 1988, pp.668-675.
142. Metric Space-based Test-data Adequacy Criteria, *The Computer Journal*, Vol. 31, No. 1, 1988, pp. 17-24. (with M. Davis)
143. How To Decide When To Stop Testing, *Proc. of the Fifth Annual Pacific Northwest Software Quality Conference*, Portland, Oregon, Oct. 1987, pp. 145-154.
144. An Overview of Software Testing Research, *Proc. of the Fourth International Conference on Testing Computer Software*, Washington, D.C., June 1987, pp. 11-28.
145. Axiomatizing Software Test Data Adequacy, *IEEE Trans. on Software Engineering*, Vol. SE-12, No. 12, Dec 1986, pp.1128-1138.
146. Data Flow Testing in the Presence of Unexecutable Paths, *Proc. of the IEEE Workshop on Software Testing*, Banff, Canada, July 1986, pp. 4-13. (with P. Frankl)
147. A Generalized Domain-Based Definition of Software Reliability, *Proc. of the IEEE Workshop on Software Testing*, Banff, Canada, July 1986, pp. 98-107. (with S. Weiss)
148. Design for a Tool to Manage Specification-Based Testing, *Proc. of the IEEE Workshop on Software Testing*, Banff, Canada, July 1986, pp. 41-50. (with T. Ostrand and R. Sigal)
149. A Data Flow Testing Tool, *Proc. IEEE Softfair II*, San Francisco, Ca., Dec 1985, pp. 46-53. (with P. Frankl)
150. ASSET: A System to Select and Evaluate Tests, *Proc. of the Conference on Software Tools*, New York, N.Y. April 1985, pp. 72-79. (with P. Frankl and S. Weiss)
151. Selecting Software Test Data Using Data Flow Information, *IEEE Trans. on Software Engineering*, Vol.SE-11, Apr 1985, pp. 367-375. (with S. Rapps)
152. Collecting and Categorizing Software Error Data in an Industrial Environment, *J. Systems and Software*, Vol.4, 1984, pp.289-300. (with T.J. Ostrand)
153. The Complexity of Data Flow Criteria for Test Data Selection, *Information Processing Letters*, Vol.19, No.2, Aug 1984, pp.103-109.
154. Assessing Test Data Adequacy Through Program Inference, *ACM Transactions on Programming Languages and Systems*, Vol.5, No.4, Oct 1983, pp.641-655.
155. A Formal Notion of Program-Based Test Data Adequacy, *Information and Control*, Vol.56, Nos.1-2, Jan/Feb 1983, pp.52-71. (with M.D. Davis)
156. Software Error Data Collection and Categorization, *Proc. of the Seventh Annual Software Engineering Workshop*, Goddard Space Flight Center, Greenbelt, Maryland, December 1982. (with T.J. Ostrand)
157. On Testing Non-testable Programs, *The Computer Journal*, Volume 25, Number 4, November, 1982, pp.465-470.
158. Data Flow Analysis Techniques For Test Data Selection, *Proc. of the 6th IEEE/ACM International Conference on Software Engineering (ICSE)*, Tokyo, Japan, September 1982, pp.272-278. (with S. Rapps)
159. The Use of Revealing Subdomains, *Proc. of the IEEE Workshop on Effectiveness of Testing and Proving Methods*, Catalina Island, May 1982. (with T.J. Ostrand)
160. Experiments With Pseudo-Oracles, *Proc. of the IEEE Workshop on Effectiveness of Testing and Proving Methods*, Catalina Island, May 1982. (with M.D. Davis and T.J. Ostrand)

161. The Generation of Test Data Using Data Flow Information, *Proc. of the IEEE Workshop on Effectiveness of Testing and Proving Methods*, Catalina Island, May 1982. (with S. Rapps)
162. Software Error Data Collection and Analysis, *Proc. of the Conference on Computers and Communications*, Phoenix, May 1982, pp.209-213. (with T.J. Ostrand)
163. Parsing Regular Grammars with Finite Lookahead, *Acta Informatica*, Volume 16, November 1981, pp.125-138. (with T.J. Ostrand and M.C. Paull)
164. Pseudo-Oracles for Non-Testable Programs, *Proc. ACM National Conference*, Los Angeles, November 1981, pp.254-257. (with M. Davis)
165. Security in Operating Systems: Separating the Roles of Rights, *Proc. of the Fifteenth Conference on Information Sciences and Systems*, The Johns Hopkins University, Baltimore, Maryland, March, 1981, pp.80-85.
166. Current Directions in the Theory of Testing, *Proc. CompSAC 80 Conference*, Chicago, October 1980, pp.386-389. (with T.J. Ostrand)
167. Theories of Program Testing and the Application of Revealing Subdomains, *IEEE Transactions on Software Engineering*, May, 1980, pp.236-245. (with T.J. Ostrand)
168. The Oracle Assumption of Program Testing, *Proc. of the Thirteenth International Conference on System Sciences*, Honolulu, Hawaii, January 1980, pp.44-49.
169. The Reduction of a Discrete Event Simulation to a Markov Chain, *Proc. of the Winter Simulation Conference*, December 1979, pp.391-397. (with A. Noetzel).
170. The Applicability of Program Schema Results to Programs, *Int. J. Comput. Inf. Sci.*, Volume 8, Number 5, Oct. 1979, pp.387-403.
171. Translatability and Decidability Questions for Restricted Classes of Program Schemas, *SIAM J. Comput.*, Volume 8, 1979, pp.587-598.
172. Modifications of the Program Schema Model, *J. Comput. Syst. Sci.*, Volume 18, Number 3, June 1979, pp.281-293.
173. Error Based Program Testing, *Proc. of Thirteenth Conference on Information Sciences and Systems*, The Johns Hopkins University, Baltimore, March 1979, pp.444-449. (with T.J. Ostrand)
174. Remarks on the Theory of Test Data Selection, *Proc. of the IEEE Workshop on Software Testing and Test Documentation*, Fort Lauderdale, Florida, December 1978, pp.1-18. (with T.J. Ostrand)
175. Fixed Lookahead Parsing of Regular Grammars by Finite State Machines, *Proc. of the Twelfth Annual Conference on Information Sciences and Systems*, The Johns Hopkins University Baltimore, Maryland, March 1978, pp.340-345. (with T.J. Ostrand and M.C. Paull)

Books

Computability, Complexity and Languages, Academic Press, December 1983. (with M.D. Davis)

Computability, Complexity and Languages, second edition, Academic Press, January 1994. (with M.D. Davis and R. Sigal)

Book Chapters

Theories of Program Testing and the Application of Revealing Subdomains, *Tutorial on Software Testing and Validation Methods*, ed. E.J. Miller and W.E. Howden, IEEE Computer Society, November 1981, pp.56-66. (with T.J. Ostrand)

An Extended Domain-Based Model of Software Reliability, *Software Reliability Models: Theoretical Developments, Evaluations, and Applications*, ed. Y.K. Malaiya and P.K. Srimani, IEEE Computer Society Press, 1990, pp98-110. (with S. Weiss)

Analyzing Partition Testing Strategies, *Automating Specification-Based Software Testing*, ed. Robert M. Poston, IEEE Computer Society, 1996, pp. 64-72. (with B. Jeng)

Automatically Generating Test Data From a Boolean Specification, *Automating Specification-Based Software Testing*, ed. Robert M. Poston, IEEE Computer Society, 1996, pp. 75-85. (with T. Goradia and A. Singh)

The Trouble With Testing Components. *Component-Based Software Engineering*, ed. G.T. Heineman and W.T. Council, Addison-Wesley, New York, 2001.

An Automated Fault Prediction System. *Making Software: What Really Works and Why We Believe It* ed. A. Oram & G. Wilson, O'Reilly Media, Oct 2010. (with T.J. Ostrand)

Proceedings Edited

ISSTA '93: International Symposium on Software Testing and Analysis, ACM Press, June 1993. (with Thomas Ostrand)

Encyclopedia Entries

Data Flow Testing, in *Encyclopedia of Software Engineering*, Wiley, pp.247-249, 1994. (with P. Frankl)

Data Flow Testing, in *Encyclopedia of Software Engineering, 2nd edition*, Wiley, pp.280-283, 2002.