Transformation Technology Bibliography, 1998

Ira Baxter
Semantic Designs
12636 Research Blvd, Suite C214
Austin, Texas 78759
idbaxter@semdesigns.com

The *fundamental ideas* for transformation systems are:

domains — where/when does one define a notation for a problem area;

semantics — how the meaning of a notation is defined;

implementation — what it means for one spec fragment to "implement" another; and

control — how to choose among possible implementations.

The following references describe various ways of implementing one or more of the above ideas. Unfortunately, there aren't any easy, obviously good articles or books on the topic, and there is a lot of exotic technology.

I recommend:

the Agresti book (a number of short papers) for overview,

the Feather survey paper for conciseness and broadness.

These are the minimum required reading to get a sense of the field. The Partsch book is excellent for a view of transformational development using wide spectrum languages, but fails to address domains and control. The Turski book is very good for some key theory, but you have to be determined to wade through it.

Bibliography

[AGRESTI86] *What are the New Paradigms?,* William W. Agresti, **New Paradigms for Software Development**, William W. Agresti, IEEE Press, 1986, ISBN 0-8186-0707-6.

[ALLEMANG90] **Understandings Programs as Devices,** Dean T. Allemang, Ph.D. Thesis, Computer Science Department, Ohio State University, Columbus, Ohio, 1990.

[ALLEN90] **Readings in Planning,** James Allen, James Hendler and Austin Tate, Morgan Kaufmann, San Mateo, California,1990, ISBN 1-55860-130-9.

[ASTESIANO87] *An Introduction to ASL,* Egidio Astesiano and Martin Wirsing**,** pp. 343-365, **Program Specification and Transformation**, L. Meertens, ed., Elsevier Science Publishers B. V. (North-Holland, 1987.

[BALASUNDARAM89] *The ParaScope Editor: An Interactive Parallel Programming Tool,* Vasanth Balasundaram, Ken Kennedy, Ulrich Kremer, Kathryn McKinley and Jaspal Subhlok, pp. 540-550, **Proceedings of Supercomputing '89**, ACM Press, November 1989, ACM Order Number 415892.

[BALZER85] *A 15 Year Perspective on Automatic Programming*, Robert Balzer, pp. 1257-1268, **IEEE Transactions on Software Engineering**, SE-11, November, 1985.

[BARSTOW87] *Artificial Intelligence and Software Engineering*, David Barstow, pp. 200-211, **Proceedings of 9th International Conference on Software Engineering,** IEEE, Monterey, California, April, 1987, ISBN 0-89791-216-0.

[BARSTOW89] *Automatic Programming for Device Control Software,* David Barstow, Schlumberger Laboratory for Computer Science, PO Box 200015, Austin, Texas 78720-0015, 1989.

[BAXTER86] *TMM: Software Maintenance by Transformation,* Guillermo Arango, Ira Baxter, Peter Freeman and Christopher Pidgeon, pp. 27-39, **IEEE Software**, 3(3), May, 1986.

[BAXTER90] **Transformational Maintenance by Reuse of Design Histories**, Ira D. Baxter, Ph.D. thesis, Information and Computer Science Department, University of California at Irvine, Technical Report 90-36, November 1990. Also available from University Microfilms International, catalog #9109634.

[BAXTER92] *Design Maintenance Systems*, Ira D. Baxter, pp. 73-89, **Communications of the ACM** 35(4), April 1992.

[BAXTER97A] *Reverse Engineering is Reverse Forward Engineering*. Ira D. Baxter and Micheal M Mehlich**. Working Conference on Reverse Engineering**, 1997, IEEE.

[BAXTER97B] *Software Change Through Design Maintenance*. Ira D. Baxter and Christopher Pidgeon**. Proceedings of International Conference on Software Maintenance** '97, 1997, IEEE Press. Postscript

[BAUER87] **(Lecture Notes in Computer Science 292): The Munich Project CIP,** F. L. Bauer, H. Ehler, A. Horsch, B. Moller, H. Partsch, O. Paukner and P. Pepper, Springer-Verlag, 1987.

[BAUER89] *Formal Program Construction by Transformations-- Computer-Aided, Intuition-Guided Programming,* Fredrich Ludwig Bauer, Bernhard Moller, Helmut Partsch and Peter Pepper, pp. 165-180, **IEEE Transactions on Software Engineering**, 15(2), February 1989.

[BERGSTRA89] **Algebraic Specification,** J. A. Bergstra, J. Heering and P. Klint, ACM Press, New York, 1989, ISBN 0-201-41635-2.

[BIGGERSTAFF98] *Anticipatory Optimization in Domain Specific Translation ,*T. Biggerstaff**, 5th International Conference on Software Reusability**, 1998, IEEE.

[BONDORF91] **Similix Manual, System version 4.0,** Anders Bondorf, DIKU, University of Copenhagen, Denmark, September 1991, (included in Similix distribution).

[BOYLE84] *Program Reuseability through Program Transformation*, J. M. Boyle and M. N. Muralidharan, pp. 575-588, **IEEE Transactions on Software Engineering**, SE-10(5), 1984.

[BROY80] *Semantic Relations in Programming Lanaguages,* M. Broy, H. Partsch, P. Pepper and M. Wirsing, pp. 101-106, **Information Processing 80,** S. H. Lavington, ed., North-Holland Publishing Company, New York, 1980.

[BULL95] *A Language for Expressing Program Transformations*. Tim Bull. ICSE-17 Workshop on Program Transformations, 1995.

[CHEATHAM79] *A System for Program Refinement,* T. E. Cheatham, Jr., J. A. Townley and G. H. Holloway, pp. 53-63, **Proceedings of the 4th International Conference on Software Engineering**, September 1979.

[CHEATHAM81] *Program refinement by transformation,* T. E. Cheatham, Jr., J. A. Townley and G. H. Holloway, pp. 430-437, **Proceedings of the Fifth International Conference on Software Engineering,** San Diego, California, March 1981, reprinted in [AGRESTI86].

 [CHEATHAM84] *Reusability Through Program Transformation*, T. E. Cheatham, Jr., pp. 589-594, **IEEE Transactions on Software Engineering,** SE-10(5), September 1984.

[CLAUS79] **(Lecture Notes in Computer Science 73) Graph Grammars and Their Application to Computer Science and Biology**, V. Claus, H. Ehrig and G. Rozenberg, Springer Verlag, New York, 1979.

[CYTRON90] *Efficiently Computing Static Single Assignment Form and the Control Dependence Graph,* R. Cytron, J. Ferrante, B. Rosen, and M. Wegman, ACM Transactions on Programming Languages and Systems, Vol 13 No. 4, October 1991 pp. 451-490.

[DEWAR81] *Higher Level Programming: Introduction to the use of the set-theoretic programming language SETL,* R. B. K. Dewar, E. Schonberg and J. T. Schwartz, Courant Institute of Mathematical Sciences, New York University, 1981.

[DIJKSTRA76] **A Discipline of Programming,** Edsger W. Dijkstra, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976, ISBN 0-13-215871-X.

[EHRIG80] *A Case Study of Abstract Implementations and Their Correctnes,* H. Ehrig, H.-J. Kreowski and P. Padawitz, pp. 209-263, **(Lecture Notes in Computer Science 83) Proceeding of the Fourth 'Colloque International sur la Programmation',** 1982.

[ERSHOV82] *Mixed computation: potential applications and problems for study,* A. Ershov, pp. 41-67, **Theoretical Computer Science,** vol 18, 1982.

[FEATHER83] *Reuse in the Context of a Transformation Based Methodology,* Martin S. Feather, pp. 50-58, **ITT Proceedings of the Workshop on Reusability in Programming**, ITT, Newport, Rhode Island, 1983, reprinted in **Tutorial: Software Reusability,** Peter Freeman, ed., IEEE Computer Society, 1987.

[FEATHER84] *Specification and Transformation: Automated Implementation,* Martin S. Feather, pp. 223-230, **Program Transformation and Programming Environments: Report on a Workshop directed by F. L. Bauer and H. Remus**, P. Pepper, ed., Springer-Verlag, New York, 1984.

[FEATHER86] *A Survey and Classification of some Program Transformation Approaches and Techniques,* Martin S. Feather, Information Sciences Institute, University of Southern California, April 1986, presented at IFIP WG2.1 Working Conference on Program Specification and Transformation, Bad Toelz, Germany, April 1986.

[FEATHER89A] *Constructing Specifications by Combining Parallel Elaborations*, Martin S. Feather, pp. 198-208, **IEEE Transactions on Software Engineering**, 15(2), February 1989.

[FEATHER89B] *Detecting Interference when Merging Specification Evolutions,* Martin S. Feather, pp. 169-176, **Proceedings, Fifth International Workshop on Software Specification and Design**, Pittsburgh, Pennsylvania, May 1989, published as **ACM SIGSOFT Engineering Notes** 14(3), May 1989.

[FREEMAN87] *A Conceptual Analysis of the Draco Approach to Constructing Software Systems*, Peter Freeman, pp. 830-844, **IEEE Transactions on Software Engineering**, SE-13(7), July 1987.

[GALLIER87] *A General Complete E-Unification Procedure*, Jean H. Gallier and Wayne Snyder, pp. 216-227, **(Lecture Notes in Computer Science 256) Rewriting Techniques and Applications**, Bordeaux, France, Springer-Verlag, May 1987.

 [GEHANI86] **Software Specification Techniques**, Narain Gehani and Andrew McGettrick, eds., Addison-Wesley, Reading, Massachusetts, 1986.

 [GEORGEFF87B] **Planning**, Michael P. Georgeff, Annual Review Computer Science, volume 2, pps. 359-400, 1987, reprinted in **Readings in Planning**, J. Allen, J. Hendler and Austin Tate, eds., 1990, Morgan Kaufmann, San Mateo, California, ISBN 1-55860-130-9.

[GOGUEN83A] *INSTITUTIONS: Abstract Model Theory for Program Specification*, J. A. Goguen and R. M. Burstall, SRI International and University of Edinburgh, January 1983.

[GOGUEN86] *Reusing and Interconnecting Software Components,* J. Goguen, **IEEE Computer** 19(2), February 1986.

[GORDON85A] *Why Higher-order Logic is a Good Formalism for Specifying and Verifying Hardware,* Michael Gordon, Computer Laboratory, University of Cambridge, England, Report 77, 1985.

[GOLDBERG86] *Knowledge-Based Programming: A Survey of Program Design and Construction Techniques,* Allen T. Goldberg, pp. 752-768, **IEEE Transactions on Software Engineering**, SE-12(7), 1986.

[GREEN81] *Research on Knowledge-Based Programming and Algorithm Design,* Cordell Green, Jorge Phillips, Stephen Westfold, Tom Pressburger, Beverly Kedzierski, Susan Angebranndt, Bernard Mont-Reynaud and Steve Tappel, KES.U.81.2, Kestrel Institute, 1801 Page Mill Road, Palo Alto, California, August 1981.

[HABEL87] *Jungle Evaluation,* Annegret Habel, Hans-Jorg Kreowski and Detlef Plump, pp. 92-112, **(Lecture Notes in Computer Science 332) Recent Trends in Data Type Specification: 5th Workshop on Specification of Abstract Data Types,** Gullane, Scotland, Springer-Verlag, New York 1987.

[HANSEN] *The ACT-System: Experiences and Future Enhancements*, Horst Hansen, pp. 113-130, **(Lecture Notes in Computer Science 332) Recent Trends in Data Type Specification: 5th Workshop on Specification of Abstract Data Types,** Gullane, Scotland, Springer-Verlag, New York 1987.

[HECKMANN88] *A Functional Language for the Specification of Complex Tree Transforms*, R. Heckmann, **Proceedings of European Symposium On Programming '88**, January 1988.

[HOARE85A] **The Mathematics of Programming**, C. A. R. Hoare, Clarendon Press, Oxford, England, 1986.

[HUET77] *Confluent Reductions: Abstract Properties and Applications to Term-rewriting Systems*, G. Huet, pp. 30-45, **18th Annual Symposium on Foundations of Computer Science**, 1977.

[HUET78] *Proving and applying program transformations expressed with second-order patterns*, G. Huet and B. Lang, pp. 31-55, **Acta Informatica**, Volume 11, 1978.

[JANSSEN82] *Some Observations on Compositional Semantics*, Theo M. V. Janssen and Peter van Emde Boas, pp. 137-149, **(Lecture Notes in Computer Science 131) Logic of Programs,** New York, 1982.

[JONES89] *Mix: A Self-Applicable Partial Evaluator for Experiments in Compiler Generation*, Neil D. Jones, Peter Sestoft and Harald Sondergaard, **Lisp and Symbolic Computation**, 2(9), 1989.

 [JONES90] **Systematic Software Development using VDM (2nd edition),** C.B. Jones, Prentice-Hall International 1990.

[KAISER89] *Rule-Based Modelling of the Software Development Process*, Gail Kaiser, pp. 84-86, **Proceedings of the 4th International Software Process Workshop: Representing and Enacting the Software Process,** Moretonhampstead, Devon, United Kingdom, May, 1988, published in **ACM Sigsoft Software Engineering Notes** 14(4), May 1989.

[KANT79] **Efficiency considerations in program synthesis: A knowledge-based approach**, Elaine Kant, Ph.D. thesis, Computer Science Department, Stanford University, 1979.

[KANT81] *The Refinement Paradigm: The Interaction of Coding and Efficiency Knowledge in Program Synthesis,* Elaine Kant and David R. Barstow, **IEEE Transactions on Software Engineering**,
 September 1981, reprinted in [AGRESTI86] pp. 257-270.

[KANT89] *Automatic Synthesis of Finite Difference Programs*, F. Daube, E. Kant, W. MacGregor and J. Wald, Schlumberger Laboratory for Computer Science, PO Box 200015, Austin, Texas 78720-0015, 1989.

[KANT90] *Automated Synthesis of Finite Difference Programs*, Elaine Kant, Francois Daube, William MacGregor and Joseph Wald, **Symbolic Computations and Their Impact on Mechanics PVP-205**,
 The American Society of Mechanical Engineers, New York, NY,1990, ISBN 0-791800598-0.

[KANT91] *Scientific Programming by Automated Synthesis*, Elaine Kant, Francois Daube, William MacGregor and Joseph Wald, **Automating Software Design**, M. Lowry and R. McCartney, eds., AAAI Press, 1991.

[KATZ91] *Constraint Propagation in Software Libraries,* Martin David Katz and Dennis J. Volper, University of California at Irvine, 1992, submitted to special issue on Domain Analysis for **Software and Knowledge Engineering**, G. Arango, ed.

[KELSEY89] **Compilation by Program Transformation**, Richard Kelsey, Ph.D. thesis, Yale University, May 1989.

[KIBLER78] **Power, Efficiency, and Correctness of Transformation Systems**, Dennis F. Kibler, Ph.D. thesis, University of California at Irvine, Irvine California, 1978.

[KRIEG-BRUCKNER88] *The PROSPECTRA Methodology of Program Development*, Bernd Krieg-Bruckner, pp. 1-15, **IFIP/IFAC Working Conference on Hardware and Software for Real Time Process Control,** Warsaw, Poland, North-Holland, New York, 1988.

[KRIEG-BRUCKNER89A] *Algebraic Specification and Functionals for Transformational Program and Meta Program Development,* Bernd Krieg-Bruckner, pp. 36-59, **(Lecture Notes in Computer Science 352)TAPSOFT '89: Proceedings of the International Joint Conference on Theory and Practice of Software Development**, Barcelona, Spain, Springer-Verlag, March 1989.

 [KRIEG-BRUCKNER89B] *ESPRIT Project Report #390: Algebraic Specification with Functionals in Program Development by Transformation*, B. Krieg-Bruckner**, Esprit 89: Proceedings of the 6th Annual Esprit Conference**, Brussels, Belguim, Luwer Academics, Dordrecht, The Netherlands, November 1989. Edited by Commission of the European Communities, Directorate-General Telecommunications, Information Industries and Innovation.

[LEE88] *Formally Integrating Real-Time Specification and Implementation: A Research Proposal,* Insup Lee, Susan Davidson and Richard Gerber, GRASP Lab 160, MS-CIS-88-84, Computer and Information Science Department, University of Pennsylvania, 1988.

[LOGRIPPO92] *An introduction to LOTOS: learning by examples*, L. Logrippo, M. Faci and M. Haj-Hussein, pp. 325-342, **Computer Networks and ISDN Systems**, Volume 23, North-Holland, 1992.

[LOWRY89] *Chapter XX: Knowledge-based Software Engineering*, Michael Lowry and Raul Duran, pp. 242-322, **The Handbook of Artificial Intelligence, Volume 4**, Avron Barr, Paul R. Cohen and Edward A. Feigenbaum, eds., Addison-Wesley, Reading, Massachusetts 1989.

[LOWRY91] **Automating Software Design**, M. Lowry and R. McCartney, AAAI Press, 1991.

[MAGGIOLO-SCHETTINI88] *Web Structures: A tool for Representing and Manipulating Programs*, A. Maggiolo-Schettini, M. Napoli and G. Tortora, **IEEE Transactions on Software Engineering,** 14(11), November 1988.

[MAIBAUM84] *On What Exactly is Going On When Software is Developed Step-by-Step*, T.S.E. Maibaum and W. M. Turski, pp. 528-533, **Proceedings of the IEEE 7th International Conference on Software Engineering,** IEEE, March 1984.

[MCCARTNEY87] *Synthesizing Algorithms with Performance Constraints*, Robert D. McCartney, pp. 149-154, **Proceedings 6th National Conference on Artificial Intelligence**, AAAI, Seattle, Washington, July 1987.

[MCCARTNEY88] **Synthesizing algorithms with performance constraints**, Robert McCartney, Ph.D. thesis, Technical Report No. CS-87-28, Department of Computer Science, Brown University, 1988.

[MELTON85] *Galois Connections and Computer Science Applications*, A. Melton, D. A. Schmidt and G. E. Strecker, pp. 299-312, **(Lecture Notes in Computer Science 240) Category Theory and Computer Programming,** Springer-Verlag, New York, 1985.

[MEERTENS86] **Proceedings of the IFIP TC2/WG 2.1 Working Conference on Program Specification and Transformation**, L G. L. T. Meertens, eds., Bad Tolz, FRG, North-Holland, New York, 1986.

[MANNA92] *Fundamentals of Deductive Program Synthesis* , Zohar Manna and Richard Waldinger, CS-92-1404, Stanford University, Department of Computer Science, January 1992. Submitted to **IEEE Transactions on Software Engineering.**

[MILLI86] *Formal Models of Stepwise Refinement*, Ali Milli, Jules Desharnais and Jean Raymond Gagne, pp. 231-276, **ACM Computing Surveys** 18(3), September 1986.

[MORGAN84] *Specification of the Unix Filesystem*, Carrol Morgan and Bernard Sufrin**,** pp. 128-142, **IEEE Transactions on Software Engineering** SE-10(2), March 1984.

[MOSTOW84] *Application of a Transformational Software Development Methodology to VLSI Design,* Jack Mostow and Bob Balzer, pp. 51-61, **Systems and Software** Volume 4, 1984.

[MOSTOW85] *Towards better models of the design process*, J. Mostow, pp. 44-56, **AI Magazine** 6(1), 1985.

[NAGL78] *A Tutorial and Bibliographical Survey on Graph Grammars*, Manfred Nagl, pp. 1-69, **(Lecture Notes in Computer Science 73) Graph Grammars and Their Application to Computer Science and Biology**, V. Claus, H. Ehrig and G. Rozenberg, eds., Springer-Verlag, New York, 1978.

[NEIGHBORS80] **Software Construction Using Components,** James Milne Neighbors, Ph.D. thesis, Tech Report 160, Department of Information and Computer Science, University of California at Irvine, 1980.

[NEIGHBORS84A] *The Draco Approach to Constructing Software from Reusable Components,* James Neighbors, **IEEE Transactions on Software Engineering**, SE-10(5), September 1984.

[NEIGHBORS84B] **Draco 1.3 Users Manual,** James M. Neighbors, Technical Report 230, Information and Computer Sciences Department, University of California at Irvine, October 1984.

[NEIGHBORS89] *Draco: A Method for Engineering Reusable Software Systems,* James M. Neighbors, pp. 295-320, **Software Reusability**, Ted J. Biggerstaff and Alan J. Perlis, eds., ACM Press, New York,1989, ISBN 0-201-08017-6.

[PACIFIC-SIERRA90] **The MIMDizer User's Guide,** Pacific-Sierra Research Corporation, 12340 Santa Monica Blvd, Los Angeles, CA 90025, 1990.

[PAGAN81] **Formal Specification of Programming Languages: A Panoramic Primer,** Frank G. Pagan, Prentice-Hall, Englewood Cliffs, New Jersey 07632, 1981, ISBN 0-13-329052-2.

[PAIGE82] *Finite Differencing of Computable Expressions*, R. Paige and S. Koenig, pp. 402-454, **ACM Transactions on Programming Languages and Systems** 4(3), July 1982.

[PAIGE86] *Programming with Invariants*, Robert Paige, pp. 56-69, **IEEE Software** 3(1), January 1986.

[PARTSCH83] *Program Transformation Systems,* H. Partsch and R. Steinbruggen, pp. 199-236, **Computing Surveys** 15(3), March 1983. Reprinted in [AGRESTI86].

[PARTSCH90] **Specification and Transformation of Programs,** Helmut A. Partsch, Springer-Verlag, 1990, ISBN 3-540-52589-0.

[PEPPER80] *On the Feedback between Specifications and Implementations: an Example*, P. Pepper and H. Partsch, Technical Report TUM-I80111, Institut fur Informatik, Technische Universitat Munchen, 1980.

[POTTER91] **An introduction to formal specification and Z,** B. Potter, D. Till and J. Seymour, Prentice-Hall, 1991.

[PRYWES88] *Reverse Software Engineering,* N. Prywes, X. Ge, I. Lee and M. Song, MS-CIS-88-99, Computer and Information Science Department, University of Pennsylvania, 1988.

[REDDY88] *Transformational derivation of programs using the Focus System,* Uday S. Reddy, **Proceedings of the Symposium on Practical Software Engineering Environments**, Illinois Urbana-Champain, **ACM SIGSOFT/SIGPLAN**, 1988.

[REASONINGSYSTEMS86] REFINE User's Guide, Reasoning Systems Incorporated, Palo Alto, 1986.

[RICH86] **Readings in Artificial Intelligence and Software Engineering,** Charles Rich and Richard C. Waters, eds., Morgan-Kauffmann, Los Altos, California, 1986.

[SANELLA88B] *A Survey of Formal Software Development Methods,* Donald Sanella, ECS-LFCS-88-56, Laboratory for Foundations of Computer Science, University of Edinburgh, July 1988.

[SCACCHI86] *Gist: An Operational Knowledge Specification Language,* Walt Scacchi, University of Southern California, Information Sciences Institute Institute, April 1986.

[SCHERLIS81] *Program Improvement by Internal Specialization,* William L. Scherlis, **Conference Record of the Eighth Annual ACM Symposium on Principles of Programming Languages**, ACM, Williamsburg, Virginia, January 1981.

[SCHERLIS83] *First Steps Towards Inferential Programming,* W. Scherlis and D. Scott, pp. 199-212, **Proceedings IFIP 9th World Computer Congress,** 1983.

[SEPPANEN87] *Background and Relevant Work: Navigation in a Transformational Software Construction Process*, Veikko Seppanen, Advanced Software Engineering Project, Department of Information and Computer Science, University of California at Irvine, August 1987.

[SIMONYI95] *The Death of Computer Languages, The Birth of Intentional Programming*, Charles Simonyi, 28th Annual International Seminar on the Teaching of Computing Science at University Level, (sub-Seminar) The Future of Software, University of Newcastle upon Tyne, September 1995. Microsoft Tech report MSR-TR-95-52.

[SIMONYI96] *Intentional Programming - Innovation in the Legacy Age*, Charles Simonyi, Microsoft Research, presented at IFIP WG 2.1 meeting, June 4, 1996.

[SIMONYI98] *Transformation in Intentional Programming* , W. Aitken, B. Dickens, P. Kwiatkowski, O. de Moor, D. Richter, and C. Simonyi, **5th International Conference on Software Reusability**, 1998, IEEE.

[SINTZOFF86] *Replay Project Synopsis,* M. Sintzoff, REPLAY-1651, Cisi Ingenierie, December 1986.

(This report actually has no designated author. However, the contents are very similar to that of M. Sintzoff's work, and so it has been attributed to him. An addendum to this report appeared in August 1987).

[SMITH85] *Top-Down Synthesis of Divide-and-Conquer Algorithms,* Douglas R. Smith, pp. 43-96,

Artificial Intelligence (27), Elsevier Sciences B. V. (North Holland), 1985.

[SMITH86] *Research on Knowledge-Based Software Environments at Kestrel Institute,* D. R. Smith, G. B. Kotik and S. J. Westfold, pp. 1278-1295, **IEEE Transactions on Software Engineering** SE-11, November 1985.

[SMITH89A] *KIDS: A Semi-Automatic Program Development System*, Douglas R. Smith, Kestrel Institute, Palo Alto, California 94304, October 1989. To appear in *Special Issue on Formal Methods***, IEEE Transactions on Software Engineering.**

[SMITH89B] *Algorithm Theories and Design Tactics,* Douglas R. Smith and Micheal R. Lowry, KES.U.89.3, Kestrel Institute, Palo Alto, California 94304, April 1989.

[SNEED88] *Inverse Transformation of Software from Code to Specification*, Harry M. Sneed and Gabor Jandrasics, pp. 102-109, **Proceedings- Conference on Software Maintenance**, Phoenix, Arizona, October 1988, ISBN 0-8186-0879X, IEEE Catalog 88CH2615-3.

[SRINIVAS89] *Algebraic Specification: Syntax, Semantics and Structure,* Y. V. Srinivas, Technical Report 90-15, Advanced Software Engineering Project, Department of Information and Computer Science, University of California at Irvine, April 1990.

[SRINIVAS96] *The Architecture of Specware, a Formal Software Development System*, Yellamraju V. Srinivas and James L. McDonald,Kestrel Institute Technical Report KES.U.96.7, August 1996

[STEIER89] **Algorithm Synthesis: A Comparative Study**, D. M. Steier and A. P. Anderson, Springer-Verlag, New York, 1989, ISBN 0-387-96960-8.

[STEINBERG87] *Design as Refinement Plus Constraint Propagation: The VEXED Experience*, L. Steinberg, pp. 830-835, **Proceedings of Sixth National Conference on Artificial Intelligence**, Seattle, AAAI, 1987. Also available as AI/VLSI Project Working Paper No. 51, Rutgers University Computer Science Department, February, 1987.

[SWARTOUT82] *On the Inevitable Intertwining of Specification and Implementation,* W. Swartout, pp. 438-440, **Communications of the ACM** 25(7), July 1982.

[TURSKI87] **The Specification of Computer Programs,** Wladyslaw M. Turski and Thomas S. E. Maibaum, Addison-Wesley, New York, 1987.

[WARD89] *The Maintainer's Assistant,* M. Ward, F. W. Calliss and M. Munro, pp. 307-315, **Proceedings of Conference on Software Maintenance,** Miami, Florida, October 1989, IEEE Computer Society Press, 1989, ISBN 0-8186-1965-1, IEEE Catalog 89CH27441-1.

[WATERS88] *Program Translation via Abstraction and Reimplementation,* R. C. Waters, pp. 1207-1228, **IEEE Transactions on Software Engineering** 14(8), August 1988.

[WEISE90] *Graphs as an Intermediate Representation for Partial Evaluation,* Daniel Weise, CSL-TR-90-421, Computer Systems Laboratory, Stanford University, March 1990.

[WERTHEIMER89] *Derivation of an Efficient Rule System Pattern Matcher,* Jeremy Wertheimer, **Proceedings IJCAI-89 Workshop on Automating Software Design,** 1989.

[WILE83] *Program Developments: Formal Explanations of Implementations,* D. Wile, pp. 902-911, **Communications of the ACM** 26(11), November 1983.

[WILLS90] *Recognizing a Program's Design: A Graph-Parsing Approach,* Charles Rich and Linda M. Wills, **IEEE Software** 7(1), January 1990.

[WIRSING89] *MENU- An Example of Systematic Reuse of Specifications*, M. Wirsing, R. Hennicker and R. Stabl, MIP-8930, Universitat Passau, Fakultat fur Mathematik und Informatik, West Germany, October 1989.

[WOLFRAM88] **Mathematica: A System for Doing Mathematics by Computer,** Stephan Wolfram, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1988, ISBN 0-201-19334-5, QA76.95.W651988.