[69] Richard Hamming. *The Art of Doing SCIENCE and Engineering: Learning to Learn*. Gordon and Breach Science Publishers, 1997.

[70] Seif Haridi and Sverker Janson. Kernel Andorra Prolog and its computation model. In *7th International Conference on Logic Programming*, pages 31–48. The MIT Press, June 1990.

[71] Seif Haridi, Peter Van Roy, Per Brand, Michael Mehl, Ralf Scheidhauer, and Gert Smolka. Efficient logic variables for distributed computing. *ACM Transactions on Programming Languages and Systems*, May 1999.

[72] Seif Haridi, Peter Van Roy, Per Brand, and Christian Schulte. Programming languages for distributed applications. *New Generation Computing*, 16(3):223–261, May 1998.

[73] Seif Haridi, Peter Van Roy, and Gert Smolka. An overview of the design of Distributed Oz. In the *2nd International Symposium on Parallel Symbolic Computation (PASCO 97)*. ACM, July 1997.

[74] Martin Henz. *Objects for Concurrent Constraint Programming*. Internationale Series in Engineering and Computer Science. Kluwer Academic Publishers, Boston, MA, USA, 1997.

[75] Martin Henz. *Objects for Concurrent Constraint Programming*, volume 426 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, November 1997.

[76] Martin Henz. Objects in Oz. Doctoral dissertation, Saarland University, Saarbrücken, Germany, May 1997.

[77] Martin Henz and Leif Kornstaedt. The Oz notation. Technical report, DFKI and Saarland University, December 1999. Available at `http://www.mozart-oz.org/`.

[78] Martin Henz, Tobias Müller, and Ka Boon Ng. Figaro: Yet another constraint programming library. In *Workshop on Parallelism and Implementation Technology for Constraint Logic Programming, International Conference on Logic Programming (ICLP 99)*, 1999.

[79] Martin Henz, Gert Smolka, and Jörg Würtz. Oz – a programming language for multi-agent systems. In *13th International Joint Conference on Artificial Intelligence*, pages 404–409. Morgan Kaufmann, August 1993.

[80] Martin Henz, Gert Smolka, and Jörg Würtz. Oz—a programming language for multi-agent systems. In Ruzena Bajcsy, editor, *13th International Joint Conference on Artificial Intelligence*, volume 1, pages 404–409, Chambéry, France, 30 August–3 September 1993. Morgan Kaufmann Publishers.

[81] Martin Henz, Gert Smolka, and Jörg Würtz. Object-oriented concurrent constraint programming in Oz. In Pascal Van Hentenryck and Vijay Saraswat, editors, *Principles and Practice of Constraint Programming*, pages 29–48, Cambridge, Mass., 1995. The MIT Press.

[82] Charles Antony Richard Hoare. Monitors: An operating system structuring concept. *Communications of the ACM*, 17(10):549–557, October 1974.

[83] Charles Antony Richard Hoare. Communicating sequential processes. *Communications of the ACM*, 21(8):666–677, August 1978.

[84] Bruce K. Holmer, Barton Sano, Michael Carlton, Peter Van Roy, and Alvin M. Despain. Design and analysis of hardware for high performance Prolog. *J. Log. Prog.*, 29:107–139, November 1996.

[85] Paul Hudak. Conception, evolution, and application of functional programming languages. *Computing Surveys*, 21(3):359–411, September 1989.

[86] Paul Hudak, John Peterson, and Joseph Fasel. A gentle introduction to Haskell version 98. Available at `http://www.haskell.org/tutorial/`.

[87] John Hughes. Why Functional Programming Matters. *Computer Journal*, 32(2):98–107, 1989.

[88] Robert A. Iannucci. *Parallel Machines: Parallel Machine Languages. The Emergence of Hybrid Dataflow Computer Architectures*. Kluwer, Dordrecht, the Netherlands, 1990.

[89] Daniel H. H. Ingalls. Design principles behind Smalltalk. *Byte*, 6(8):286–298, 1981.

[90] Joxan Jaffar and Michael Maher. Constraint logic programming: A survey. *J. Log. Prog.*, 19/20:503–581, May/July 1994.

[91] Raj Jain. *The Art of Computer Systems Performance Analysis*. Wiley Professional Computing, 1991.

[92] Sverker Janson. *AKL–A Multiparadigm Programming Language*. PhD thesis, Uppsala University and SICS, 1994.

[93] Sverker Janson and Seif Haridi. Programming paradigms of the Andorra Kernel Language. In *International Symposium on Logic Programming*, pages 167–183, October 1991.

[94] K. Jensen and N. Wirth. *Pascal: User Manual and Report (Second Edition)*. Springer-Verlag, 1978.

[95] Richard Jones and Rafael Lins. *Garbage Collection: Algorithms for Automatic Dynamic Memory Management*. John Wiley & Sons, 1996.

[96] Andreas Kågedal, Peter Van Roy, and Bruno Dumant. Logical State Threads 0.1, January 1997. Available at `http://www.info.ucl.ac.be/people/PVR/implementation.html`.

[97] Gilles Kahn. The semantics of a simple language for parallel programming. In *IFIP Congress*, pages 471–475, 1974.

[98] Gilles Kahn and David B. MacQueen. Coroutines and networks of parallel processes. In *IFIP Congress*, pages 993–998, 1977.

[99] B. W. Kernighan and D. M. Ritchie. *The C Programming Language (ANSI C), Second Edition*. Prentice Hall, 1988.

[100] Gregor Kiczales, Jim des Rivières, and Daniel G. Bobrow. *The Art of the Metaobject Protocol*. The MIT Press, 1991.

[101] Donald E. Knuth. *The Art of Computer Programming: Seminumerical Algorithms*, volume 2. Addison-Wesley.

[102] Donald E. Knuth. *The Art of Computer Programming: Fundamental Algorithms*, volume 1. Addison-Wesley, 1973.

[103] Donald E. Knuth. Structured programming with **go to** statements. *Computing Surveys*, 6(4), December 1974.

[104] Leif Kornstaedt. Gump – a front-end generator for Oz. Technical report, Mozart Consortium, December 2001. Available at `http://www.mozart-oz.org/`.

[105] S. Rao Kosaraju. Analysis of structured programs. *J. Computer and System Sciences*, 9(3), December 1974.

[106] Robert Kowalski. *Logic for Problem Solving*. North-Holland, 1979.

[107] James F. Kurose and Keith W. Ross. *Computer networking: a top-down approach featuring the Internet*. Addison-Wesley, 2001.

[108] Leslie Lamport. *LaTeX: A Document Preparation System, Second Edition*. Addison-Wesley, 1994.

[109] Hugh C. Lauer and Roger M. Needham. On the duality of operating system structures. In *Second International Symposium on Operating Systems, IRIA*, October 1978. Reprinted in *Operating Systems Review*, 13(2), April 1979, pp. 3–19.

[110] Doug Lea. *Concurrent Programming in Java*. Addison-Wesley, 1997.

[111] Doug Lea. *Concurrent Programming in Java, Second Edition*. Addison-Wesley, 2000.

[112] Nancy Leveson and Clark S. Turner. An investigation of the Therac-25 accidents. *IEEE Computer*, 26(7):18–41, July 1993.

[113] Henry Lieberman. Using prototypical objects to implement shared behavior in object-oriented systems. In *1st Conference on Object-Oriented Programming Languages, Systems, and Applications (OOPSLA 86)*, September 1986. Also in Object-Oriented Computing, Gerald Peterson, ed., IEEE Computer Society Press, 1987.

[114] John Lloyd. *Foundations of Logic Programming, Second Edition*. Springer-Verlag, 1987.

[115] Nancy Lynch. *Distributed Algorithms*. Morgan Kaufmann, San Francisco, Calif., 1996.

[116] Bruce J. MacLennan. *Principles of Programming Languages, Second Edition*. Saunders, Harcourt Brace Jovanovich, 1987.

[117] Michael Maher. Logic semantics for a class of committed-choice programs. In *International Conference on Logic Programming (ICLP 87)*, pages 858–876, May 1987.

[118] Zohar Manna. *The Mathematical Theory of Computation*. McGraw-Hill, 1974.

[119] Sun Microsystems. *The Remote Method Invocation Specification*, 1997. Available at `http://www.javasoft.com`.

[120] John McCarthy. *LISP 1.5 Programmer's Manual*. The MIT Press, 1962.

[121] Michael Mehl, Christian Schulte, and Gert Smolka. Futures and by-need synchronization for Oz. DRAFT. Available at `http://www.mozart-oz.org/papers/`, May 1998.

[122] Bertrand Meyer. *Object-Oriented Software Construction, Second Edition*. Prentice Hall, 2000.

[123] Mark Miller, Marc Stiegler, Tyler Close, Bill Frantz, Ka-Ping Yee, Chip Morningstar, Jonathan Shapiro, and Norm Hardy. E: Open source distributed capabilities, 2001. Available at `http://www.erights.org`.

[124] Mark Miller, Ka-Ping Yee, and Jonathan Shapiro. Capability myths demolished. Draft available at `http://zesty.ca/capmyths`, 2003.

[125] Mark S. Miller, Chip Morningstar, and Bill Frantz. Capability-based financial instruments. In *Financial Cryptography 2000*, Anguilla, British West Indies, February 2000.

[126] Robin Milner, Mads Tofte, and Robert Harper. *Definition of Standard ML*. MIT Press, Cambridge, MA, USA, 1990.

[127] J. Paul Morrison. *Flow-Based Programming: A New Approach to Application Development*. Van Nostrand Reinhold, 1994.

[128] Almetwally Mostafa, Iliès Alouini, and Peter Van Roy. Fault tolerant global store module, 2001. Available at `http://www.mozart-oz.org/mogul/info/mostafa/globalstore.html`.

[129] Mozart Consortium. The Mozart Programming System version 1.2.3, December 2001. Available at `http://www.mozart-oz.org/`.

[130] Peter Naur. Revised report on the algorithmic language ALGOL 60. *Communications of the ACM*, 1963.

[131] Rishiyur S. Nikhil. ID language reference manual version 90.1. Technical Report Memo 284-2, MIT, Computation Structures Group, July 1994.

[132] Rishiyur S. Nikhil. An overview of the parallel language Id – a foundation for pH, a parallel dialect of Haskell. Technical report, Digital Equipment Corporation, Cambridge Research Laboratory, 1994.

[133] Rishiyur S. Nikhil and Arvind. *Implicit Parallel Programming in pH*. Morgan Kaufmann, 2001.

[134] Donald A. Norman. *The Design of Everyday Things*. Basic Books, Inc., 1988.

[135] Theodore Norvell. Monads for the working Haskell programmer – a short tutorial. Available at `http://www.haskell.org/`.

[136] Peter Norvig. *Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp*. Morgan Kaufmann, 1992.

[137] K. Nygaard and O. J. Dahl. *The Development of the SIMULA Languages*, pages 439–493. Academic Press, 1981.

[138] Chris Okasaki. *Purely Functional Data Structures*. Cambridge University Press, 1998.

[139] Richard A. O'Keefe. *The Craft of Prolog*. The MIT Press, 1990.

[140] Andreas Paepcke, editor. *Object-Oriented Programming: The CLOS Perspective*. The MIT Press, 1993.

[141] Seymour Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. The Harvester Press, 1980.

[142] David Lorge Parnas. On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15(12):1053–1058, December 1972.

[143] David Lorge Parnas. Teaching programming as engineering. In *9th International Conference of Z Users*, volume 967 of *Lecture Notes in Computer Science*. Springer-Verlag, 1995. Reprinted in *Software Fundamentals*, Addison-Wesley, 2001.

[144] David Lorge Parnas. *Software Fundamentals*. Addison-Wesley, 2001.

[145] F. Paternò. *Model-based Design and Evaluation of Interactive Applications*. Springer-Verlag, Berlin, 1999.

[146] David A. Patterson and John L. Hennessy. *Computer Architecture: A Quantitative Approach, Second Edition*. Morgan Kaufmann, 1996.

[147] Simon Peyton Jones. Tackling the awkward squad: monadic input/output, concurrency, exceptions, and foreign-language calls in Haskell. In Tony Hoare, Manfred Broy, and Ralf Steinbruggen, editors, *Engineering theories of software construction*, pages 47–96. IOS Press, 2001. Presented at the 2000 Marktoberdorf Summer School.

[148] Simon Peyton Jones, editor. *Haskell 98 language and libraries: The revised report*. Cambridge University Press, 2003. Also published as the January 2003 Special Issue of the Journal of Functional Programming.

[149] Simon Peyton Jones, Andrew Gordon, and Sigbjorn Finne. Concurrent Haskell. In *Principles of Programming Languages (POPL)*, 1996.

[150] Shari Lawrence Pfleeger. *Software Engineering: The Production of Quality Software, Second Edition*. Macmillan, 1991.

[151] David Plainfossé and Marc Shapiro. A survey of distributed garbage collection techniques. In the *International Workshop on Memory Management*, volume 986 of *Lecture Notes in Computer Science*, pages 211–249, Berlin, September 1995. Springer-Verlag.

[152] R. J. Pooley. *An Introduction to Programming in SIMULA*. Blackwell Scientific Publishers, 1987.

[153] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 1986.

[154] Roger S. Pressman. *Software Engineering, Sixth Edition*. Addison-Wesley, 2000.

[155] Mahmoud Rafea, Fredrik Holmgren, Konstantin Popov, Seif Haridi, Stelios Lelis, Petros Kavassalis, and Jakka Sairamesh. Application architecture of the Internet simulation model: Web Word of Mouth (WoM). In *IASTED International Conference on Modelling and Simulation MS2002*, May 2002.

[156] Eric Raymond. The cathedral and the bazaar, May 1997.

[157] Juris Reinfelds. Teaching of programming with a programmer's theory of programming. In *Informatics Curricula, Teaching Methods, and Best Practice (ICTEM 2002, IFIP Working Group 3.2 Working Conference)*. Kluwer Academic Publishers, 2002.

[158] John H. Reppy. *Concurrent Programming in ML*. Cambridge University Press, 1999.

[159] James Rumbaugh, Ivar Jacobson, and Grady Booch. *The Unified Modeling Language reference manual*. Addison-Wesley, 1999.

[160] Stuart Russell and Peter Norvig. *Artificial Intelligence: A modern approach*. Prentice Hall, 1995.

[161] Oliver Sacks. *The Man Who Mistook His Wife for a Hat: And Other Clinical Tales*. Touchstone Books, 1998.

[162] Jakka Sairamesh, Petros Kavassalis, Manolis Marazakis, Christos Nikolaos, and Seif Haridi. Information cities over the Internet: Taxonomy, principles and architecture. In *Digital Communities 2002*, November 2001.

[163] Vijay A. Saraswat. *Concurrent Constraint Programming*. The MIT Press, 1993.

[164] Vijay A. Saraswat, Martin C. Rinard, and Prakash Panangaden. Semantic foundations of concurrent constraint programming. In *Principles of Programming Languages (POPL)*, pages 333–352, 1991.

[165] Steve Schneider. *Concurrent and Real-time Systems: The CSP Approach*. John Wiley & Sons, 2000.

[166] Bruce Schneier. *Applied Cryptography*. John Wiley & Sons, 1996.

[167] Christian Schulte. Programming constraint inference engines. In Gert Smolka, editor, *Proceedings of the Third International Conference on Principles and Practice of Constraint Programming*, volume 1330 of *Lecture Notes in Computer Science*, pages 519–533, Schloß Hagenberg, Austria, October 1997. Springer-Verlag.

[168] Christian Schulte. Comparing trailing and copying for constraint programming. In *International Conference on Logic Programming (ICLP 99)*, pages 275–289. The MIT Press, November 1999.

[169] Christian Schulte. *Programming Constraint Inference Services*. PhD thesis, Saarland University, Fachbereich Informatik, Saarbrücken, Germany, 2000.

[170] Christian Schulte. Programming deep concurrent constraint combinators. In Enrico Pontelli and Vítor Santos Costa, editors, *Practical Aspects of Declarative Languages, Second International Workshop, PADL 2000*, volume 1753 of *Lecture Notes in Computer Science*, pages 215–229, Boston, MA, USA, January 2000. Springer-Verlag.

[171] Christian Schulte. Oz Explorer – visual constraint programming support. Technical report, Mozart Consortium, December 2001. Available at `http://www.mozart-oz.org/`.

[172] Christian Schulte. *Programming Constraint Services: High-Level Programming of Standard and New Constraint Services*, volume 2302 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.

[173] Christian Schulte and Gert Smolka. Encapsulated search for higher-order concurrent constraint programming. In *1994 International Symposium on Logic Programming*, pages 505–520. The MIT Press, November 1994.

[174] Christian Schulte and Gert Smolka. Finite domain constraint programming in Oz. A tutorial. Technical report, DFKI and Saarland University, December 1999. Available at `http://www.mozart-oz.org/`.

[175] Ehud Shapiro. A subset of Concurrent Prolog and its interpreter. Technical Report TR-003, Institute for New Generation Computer Technology (ICOT), Cambridge, Mass., January 1983.

[176] Ehud Shapiro, editor. *Concurrent Prolog: Collected Papers*, volume 1-2. The MIT Press, Cambridge, Mass., 1987.

[177] Ehud Shapiro. The family of concurrent logic programming languages. *ACM Computing Surveys*, 21(3):413–510, September 1989.

[178] Daniel P. Siewiorek, C. Gordon Bell, and Allen Newell. *Computer Structures: Principles and Examples*. McGraw-Hill Book Company, 1982.

[179] Gert Smolka. The definition of Kernel Oz. In Andreas Podelski, editor, *Constraints: Basics and Trends*, volume 910 of *Lecture Notes in Computer Science*, pages 251–292. Springer-Verlag, Berlin, 1995.

[180] Gert Smolka. The Oz programming model. In *Computer Science Today*, volume 1000 of *Lecture Notes in Computer Science*, pages 324–343. Springer-Verlag, Berlin, 1995.

[181] Guy L. Steele, Jr. *Common LISP: The Language*. Digital Press, 1984.

[182] Leon Sterling and Ehud Shapiro. *The Art of Prolog–Advanced Programming Techniques*. Series in Logic Programming. The MIT Press, 1986.

[183] Marc Stiegler. *The E Language in a Walnut*. 2000. Draft, available at `http://www.erights.org`.

[184] Bjarne Stroustrup. *The C++ Programming Language, Third Edition*. Addison-Wesley, 1997.

[185] Giancarlo Succi and Michele Marchesi. *Extreme Programming Examined*. Addison-Wesley, 2001.

[186] Sun Microsystems. *The Java Series*. Sun Microsystems, Mountain View, Calif., 1996. Available at `http://www.javasoft.com`.

[187] Clemens Szyperski. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley and ACM Press, 1999.

[188] Andrew Taylor. *High-Performance Prolog Implementation*. PhD thesis, Basser Department of Computer Science, University of Sydney, June 1991.

[189] Gerard Tel. *An Introduction to Distributed Algorithms*. Cambridge University Press, Cambridge, United Kingdom, 1994.

[190] Evan Tick. The deevolution of concurrent logic programming. *J. Log. Prog.*, 23(2):89–123, May 1995.

[191] Kasunori Ueda. Guarded Horn Clauses. In Eiti Wada, editor, *Proceedings of the 4th Conference on Logic Programming*, volume 221 of *Lecture Notes in Computer Science*, pages 168–179, Tokyo, Japan, July 1985. Springer.

[192] Jeffrey D. Ullman. *Elements of ML Programming*. Prentice Hall, 1998.

[193] Peter Van Roy. VLSI-BAM Diagnostic Generator, 1989. Prolog program to generate assembly language diagnostics. Aquarius Project, University of California, Berkeley.

[194] Peter Van Roy. *Can Logic Programming Execute as Fast as Imperative Programming?* PhD thesis, Computer Science Division, University of California at Berkeley, December 1990. Technical Report UCB/CSD 90/600.

[195] Peter Van Roy. 1983–1993: The wonder years of sequential Prolog implementation. *J. Log. Prog.*, 19/20:385–441, May/July 1994.

[196] Peter Van Roy, Per Brand, Denys Duchier, Seif Haridi, Martin Henz, and Christian Schulte. Logic programming in the context of multiparadigm programming: the Oz experience. *Theory and Practice of Logic Programming*, 2003. To appear.

[197] Peter Van Roy, Per Brand, Seif Haridi, and Raphaël Collet. A lightweight reliable object migration protocol. In Henri E. Bal, Boumediene Belkhouche, and Luca Cardelli, editors, *Internet Programming Languages*, volume 1686 of *Lecture Notes in Computer Science*. Springer Verlag, October 1999.

[198] Peter Van Roy and Alvin Despain. High-performance logic programming with the Aquarius Prolog compiler. *IEEE Computer*, pages 54–68, January 1992.

[199] Peter Van Roy and Seif Haridi. Teaching programming broadly and deeply: the kernel language approach. In *Informatics Curricula, Teaching Methods, and Best Practice (ICTEM 2002, IFIP Working Group 3.2 Working Conference)*. Kluwer Academic Publishers, 2002.

[200] Peter Van Roy and Seif Haridi. Teaching programming with the kernel language approach. In *Workshop on Functional and Declarative Programming in Education (FDPE02), at Principles, Logics, and Implementations of High-Level Programming Languages (PLI2002)*. University of Kiel, Germany, October 2002.

[201] Peter Van Roy, Seif Haridi, Per Brand, Gert Smolka, Michael Mehl, and Ralf Scheidhauer. Mobile objects in Distributed Oz. *ACM Transactions on Programming Languages and Systems*, 19(5):804–851, September 1997.

[202] Arthur H. Veen. Dataflow machine architecture. *ACM Computing Surveys*, 18(4):365–396, December 1986.

[203] Duncan J. Watts. *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton University Press, 1999.

[204] Gerhard Weikum and Gottfried Vossen. *Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control and Recovery*. Morgan Kaufmann, 2002.

[205] Gerhard Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, 1999.

[206] Claes Wikström. Distributed programming in Erlang. In the *1st International Symposium on Parallel Symbolic Computation* (*PASCO 94*), pages 412–421, Singapore, September 1994. World Scientific.

[207] Herbert S. Wilf. *generatingfunctionology*. Academic Press, 1994.

[208] Glynn Winskel. *The Formal Semantics of Programming Languages*. Foundations of Computing Series. The MIT Press, Cambridge, Massachusetts, 1993.

[209] Noel Winstanley. What the hell are Monads?, 1999. Available at `http://www.haskell.org`.

[210] David Wood. Use of objects and agents at Symbian, September 2000. Talk given at the Newcastle Seminar on the Teaching of Computing Science.

[211] Matthias Zenger and Martin Odersky. Implementing extensible compilers. In *1st International Workshop on Multiparadigm Programming with Object-Oriented Languages*, pages 61–80. John von Neumann Institute for Computing (NIC), June 2001. Workshop held as part of ECOOP 2001.

Index