

Select Z Bibliography

Jonathan P. Bowen^{*}

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD, UK
Email: Jonathan.Bowen@comlab.ox.ac.uk
URL: <http://www.comlab.ox.ac.uk/oucl/people/jonathan.bowen.html>

Abstract. This bibliography contains a list of Z references that are either available as published papers, books or technical reports from institutions, from the author, the Oxford University Computing Laboratory (OUCL), or on-line. The bibliography is in alphabetical order by author name(s).

Introduction

The list of references presented here is maintained in electronic form, in BIB_TE_X bibliography database format, which is compatible with the widely used L_AT_EX document preparation system [297]. It is intended to keep the bibliography up to date and to issue it to coincide with the regular Z User Meetings. The latest version of BIB_TE_X source file used for this bibliography [53] is available as a searchable on-line database on the World Wide Web under the following Uniform Resource Location (URL):

<http://www.comlab.ox.ac.uk/archive/z/bib.html>

The actual BIB_TE_X source is also available under via anonymous FTP on the Internet [ftp.comlab.ox.ac.uk:/pub/Zforum/z95.bib](ftp://comlab.ox.ac.uk/pub/Zforum/z95.bib) and via electronic mail by sending an electronic mail message containing the command 'send z z95.bib' to the address archive-server@comlab.ox.ac.uk. For more information on accessing the bibliography electronically from the Z archive, see [543].

To add new references concerned with Z to this list, please send details via electronic mail to the address zforum-request@comlab.ox.ac.uk or post them to Jonathan Bowen (address above). It is helpful if you can give as much information as possible so the entry could be included as a reference in future papers concerning Z.

This bibliography has been regularly maintained for Z User Meeting proceedings in the past (e.g., see [58]). For an alternative annotated Z bibliography, see [66], originally started as a result of work on the ZIP project.

Acknowledgements

Ruaridh Macdonald of RSRE, Malvern, initiated the idea of a Z bibliography and helped maintain it for several years. Joan Arnold at the OUCL has previously assisted in maintaining the bibliography as part of her work as secretary to the European ESPRIT **ProCoS-WG** Working Group (no. 8694) on 'Provably Correct Systems'. Thank you to everybody who has sent in entries over the years.

^{*} Funded by the UK Science and Engineering Research Council (SERC) on grant no. GR/J15186.

References

1. G. D. Abowd. Agents: Communicating interactive processes. In D. Diaper, D. Gilmore, Gilbert Cockton, and Brian Shackel, editors, *Human-Computer Interaction: INTERACT'90*, pages 143–148. Elsevier Science Publishers (North-Holland), 1990.
2. G. D. Abowd. *Formal Aspects of Human-Computer Interaction*. DPhil thesis, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, 1991.
3. G. D. Abowd, R. Allen, and D. Garlan. Using style to understand descriptions of software architectures. *ACM Software Engineering Notes*, 18(5):9–20, December 1993.
4. G. D. Abowd, J. P. Bowen, A. J. Dix, M. D. Harrison, and R. Took. User interface languages: A survey of existing methods. Technical Report PRG-TR-5-89, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, October 1989.
5. J.-R. Abrial. The B tool. In Bloomfield et al. [40], pages 86–87.
6. J.-R. Abrial. The B method for large software, specification, design and coding (abstract). In Prehn and Toetenel [399], pages 398–405.
7. J.-R. Abrial. *The B-Book*. Cambridge University Press, To appear.
Contents: Mathematical reasoning; Set notation; Mathematical objects; Introduction to abstract machines; Formal definition of abstract machines; Theory of abstract machines; Constructing large abstract machines; Example of abstract machines; Sequencing and loop; Programming examples; Refinement; Constructing large software systems; Example of refinement;
Appendices: Summary of the most current notations; Syntax; Definitions; Visibility rules; Rules and axioms; Proof obligations.
8. J.-R. Abrial, S. A. Schuman, and B. Meyer. Specification language. In R. M. McKeag and A. M. Macnaghten, editors, *On the Construction of Programs: An Advanced Course*, pages 343–410. Cambridge University Press, 1980.
9. J.-R. Abrial and I. H. Sørensen. KWIC-index generation. In J. Staunstrup, editor, *Program Specification: Proceedings of a Workshop*, volume 134 of *Lecture Notes in Computer Science*, pages 88–95. Springer-Verlag, 1981.
10. M. Ainsworth, A. H. Cruikchank, P. J. L. Wallis, and L. J. Groves. Viewpoint specification and Z. *Information and Software Technology*, 36(1):43–51, 1994.
11. A. J. Alencar and J. A. Goguen. OOZE: An object-oriented Z environment. In P. America, editor, *Proc. ECOOP'91 European Conference on Object-Oriented Programming*, volume 512 of *Lecture Notes in Computer Science*, pages 180–199. Springer-Verlag, 1991.
12. A. J. Alencar and J. A. Goguen. Two examples in OOZE. Technical Report PRG-TR-25-91, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, 1991.
13. P. Ammann and J. Offutt. Functional and test specifications for the Mistix file system. Technical Report ISSE-TR-93-100, Department of Information & Software Systems Engineering, George Mason University, USA, January 1993.
14. P. Ammann and J. Offutt. Using formal methods to mechanize category-partition testing. Technical Report ISSE-TR-93-105, Department of Information & Software Systems Engineering, George Mason University, USA, September 1993.
15. D. B. Arnold, D. A. Duce, and G. J. Reynolds. An approach to the formal specification of configurable models of graphics systems. In G. Maréchal, editor, *Proc. EUROGRAPHICS'87, European Computer Graphics Conference and Exhibition*, pages 439–463. Elsevier Science Publishers (North-Holland), 1987.
The paper describes a general framework for the formal specification of modular graphics systems, illustrated by an example taken from the Graphical Kernel System (GKS) standard.

16. D. B. Arnold and G. J. Reynolds. Configuring graphics systems components. *IEE/BCS Software Engineering Journal*, 3(6):248–256, November 1988.
17. R. D. Arthan. Formal specification of a proof tool. In Prehn and Toetenel [398], pages 356–370.
18. R. D. Arthan. On free type definitions in Z. In Nicholls [375], pages 40–58.
19. K. Ashoo. The Genesis Z tool – an overview. *BCS FACS FACTS*, Series II, 3(1):11–13, May 1992.
20. S. Aujla, A. Bryant, and L. Semmens. A rigorous review technique: Using formal notations within conventional development methods. In *Proc. 1993 Software Engineering Standards Symposium*, pages 247–255. IEEE Computer Society Press, 1993.
21. P. B. Austin, K. A. Murray, and A. J. Wellings. File system caching in large point-to-point networks. *IEE/BCS Software Engineering Journal*, 7(1):65–80, January 1992.
22. S. Austin and G. I. Parkin. Formal methods: A survey. Technical report, National Physical Laboratory, Queens Road, Teddington, Middlesex, TW11 0LW, UK, March 1993.
23. C. Bailes and R. Duke. The ecology of class refinement. In Morris and Shaw [359], pages 185–196.
24. M. Bailey. Formal specification using Z. In *Proc. Software Engineering anniversary meeting (SEAS)*, page 99, 1987.
25. J. Bainbridge, R. W. Whitty, and J. B. Wordsworth. Obtaining structural metrics of Z specifications for systems development. In Nicholls [373], pages 269–281.
26. J.-P. Banâtre. About programming environments. In J.-P. Banâtre, S. B. Jones, and D. de Métayer, editors, *Prospects for Functional Programming in Software Engineering*, volume 1 of *Research Reports*, chapter 1, pages 1–22. Springer-Verlag, 1991.
27. R. Barden and S. Stepney. Support for using Z. In Bowen and Nicholls [75], pages 255–280.
28. R. Barden, S. Stepney, and D. Cooper. The use of Z. In Nicholls [375], pages 99–124.
29. R. Barden, S. Stepney, and D. Cooper. *Z in Practice*. BCS Practitioner Series. Prentice Hall, 1994.
30. G. Barrett. Formal methods applied to a floating-point number system. *IEEE Transactions on Software Engineering*, 15(5):611–621, May 1989.
This paper presents a formalization of the IEEE standard for binary floating-point arithmetic in Z. The formal specification is refined into four components. The procedures presented form the basis for the floating-point unit of the Inmos IMS T800 transputer. This work resulted in a joint UK Queen’s Award for Technological Achievement for Inmos Ltd and the Oxford University Computing Laboratory in 1990. It was estimated that the approach saved a year in development time compared to traditional methods.
31. L. M. Barroca and J. A. McDermid. Formal methods: Use and relevance for the development of safety-critical systems. *The Computer Journal*, 35(6):579–599, December 1992.
32. P. Baumann. Z and natural semantics. In Bowen and Hall [67], pages 168–184.
33. M. Benjamin. A message passing system: An example of combining CSP and Z. In Nicholls [371], pages 221–228.
34. M. Benveniste. Operational semantics of a distributed object-oriented language and its Z formal specification. Rapport de recherche INRIA 1230, IRISA/INRIA-Rennes, Campus de Beaulieu, 35042 Rennes Cédex, France, May 1990.
35. M. Benveniste. Writing operational semantics in Z: A structural approach. In Prehn and Toetenel [398], pages 164–188.
36. S. Bera. Structuring for the VDM specification language. In Bloomfield et al. [40], pages 2–25.
37. J. Bicarregui and B. Ritchie. Invariants, frames and postconditions: A comparison of the VDM and B notations. *IEEE Transactions on Software Engineering*, 21(2):79–89, 1995.

38. P. G. Bishop, editor. *Fault Avoidance*, chapter 3, pages 56–140. Applied Science. Elsevier Science Publishers, 1990.
Section 3.88 (pages 94–96) provides an overview of Z. Other sections describe related techniques.
39. D. Bjørner, C. A. R. Hoare, and H. Langmaack, editors. *VDM and Z – Formal Methods in Software Development*, volume 428 of *Lecture Notes in Computer Science*. VDM-Europe, Springer-Verlag, 1990.
The 3rd VDM-Europe Symposium was held at Kiel, Germany, 17–21 April 1990. A significant number of papers concerned with Z were presented [105, 161, 191, 145, 203, 209, 290, 424, 458, 500, 530].
40. R. Bloomfield, L. Marshall, and R. Jones, editors. *VDM – The Way Ahead*, volume 328 of *Lecture Notes in Computer Science*. VDM-Europe, Springer-Verlag, 1988.
The 2nd VDM-Europe Symposium was held at Dublin, Ireland, 11–16 September 1988. See [5, 36].
41. D. Blyth. The CICS application programming interface: Temporary storage. IBM Technical Report TR12.301, IBM United Kingdom Laboratories Ltd, Hursley Park, Winchester, Hampshire SO21 2JN, UK, December 1990.
One of a number of reports on the CICS application programming interface. See also [247, 289, 360].
42. A. Boswell. Specification and validation of a security policy model. In Woodcock and Larsen [527], pages 42–51.
Revised version in [43].
43. A. Boswell. Specification and validation of a security policy model. *IEEE Transactions on Software Engineering*, 21(2):99–106, 1995.
This paper describes the development of a formal security model in Z for the NATO Air Command and Control System (ACCS): a large, distributed, multilevel-secure system. The model was subject to manual validation, and some of the issues and lessons in both writing and validating the model are discussed.
44. L. Bottaci and J. Jones. *Formal Specification Using Z: A Modelling Approach*. International Thomson Publishing, London, 1995.
45. J. P. Bowen. Formal specification and documentation of microprocessor instruction sets. *Microprocessing and Microprogramming*, 21(1–5):223–230, August 1987.
46. J. P. Bowen. The formal specification of a microprocessor instruction set. Technical Monograph PRG-60, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, January 1987.
The Z notation is used to define the Motorola M6800 8-bit microprocessor instruction set.
47. J. P. Bowen, editor. *Proc. Z Users Meeting, 1 Wellington Square, Oxford*, Wolfson Building, Parks Road, Oxford, UK, December 1987. Oxford University Computing Laboratory.
The 1987 Z Users Meeting was held on Friday 8 December at the Department of External Studies, Rewley House, 1 Wellington Square, Oxford, UK.
48. J. P. Bowen. Formal specification in Z as a design and documentation tool. In *Proc. Second IEE/BCS Conference on Software Engineering*, number 290 in Conference Publication, pages 164–168. IEE/BCS, July 1988.
49. J. P. Bowen, editor. *Proc. Third Annual Z Users Meeting*, Wolfson Building, Parks Road, Oxford, UK, December 1988. Oxford University Computing Laboratory.
The 1988 Z Users Meeting was held on Friday 16 December at the Department of External Studies, Rewley House, 1 Wellington Square, Oxford, UK. Issued with *A Miscellany of Handy Techniques* by R. Macdonald, RSRE, *Practical Experience of Formal Specification: A programming interface for communications* by J. B. Wordsworth, IBM, and a number of posters.

50. J. P. Bowen. Formal specification of window systems. Technical Monograph PRG-74, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, June 1989.
Three existing window systems, X from MIT, WM from Carnegie-Mellon University and the Blit from AT&T Bell Laboratories are covered.
51. J. P. Bowen. POS: Formal specification of a UNIX tool. *IEE/BCS Software Engineering Journal*, 4(1):67–72, January 1989.
52. J. P. Bowen. Formal specification of the ProCoS/safemos instruction set. *Microprocessors and Microsystems*, 14(10):631–643, December 1990.
This article is part of a special issue on *Formal aspects of microprocessor design*, edited by H. S. M. Zedan. See also [437].
53. J. P. Bowen. Z bibliography. Oxford University Computing Laboratory, 1990–1993.
This bibliography is maintained in BIB_TE_X database format at the Oxford University Computing Laboratory. To add entries, please send as complete information as possible to `zforum-request@comlab.ox.ac.uk`.
54. J. P. Bowen. X: Why Z? *Computer Graphics Forum*, 11(4):221–234, October 1992.
This paper asks whether window management systems would not be better specified through a formal methodology and gives examples in Z of X11.
55. J. P. Bowen. Formal methods in safety-critical standards. In *Proc. 1993 Software Engineering Standards Symposium*, pages 168–177. IEEE Computer Society Press, 1993.
56. J. P. Bowen. Report on Z User Meeting, London 1992. *BCS FACS FACTS*, Series III, 1(3):7–8, Summer 1993.
Other versions of this report have appeared as follows:
 - Z User Meetings, *Safety Systems: The Safety-Critical Systems Club Newsletter*, 3(1):13, September 1993.
 - Z User Group activities, *JFIT News*, 46:5, September 1993.
 - Report on Z User Meeting, *Information and Software Technology*, 35(10):613, October 1993.
 - Z User Meeting Activities, *High Integrity Systems*, 1(1):93–94, 1994.
57. J. P. Bowen. Comp.specification.z and Z FORUM frequently asked questions. In Bowen and Hall [67], pages 397–404.
58. J. P. Bowen. Select Z bibliography. In Bowen and Hall [67], pages 359–396.
59. J. P. Bowen. *Formal Specification and Documentation using Z: A Case Study Approach*. International Thomson Press, London, 1995.
60. J. P. Bowen. Z glossary. *Information and Software Technology*, 37(5):333–334, May 1995.
61. J. P. Bowen, P. T. Breuer, and K. C. Lano. A compendium of formal techniques for software maintenance. *IEE/BCS Software Engineering Journal*, 8(5):253–262, September 1993.
62. J. P. Bowen, P. T. Breuer, and K. C. Lano. Formal specifications in software maintenance: From code to Z⁺⁺ and back again. *Information and Software Technology*, 35(11/12):679–690, November/December 1993.
63. J. P. Bowen, R. B. Gimson, and S. Topp-Jørgensen. The specification of network services. Technical Monograph PRG-61, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, August 1987.
64. J. P. Bowen, R. B. Gimson, and S. Topp-Jørgensen. Specifying system implementations in Z. Technical Monograph PRG-63, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, February 1988.
65. J. P. Bowen and M. J. C. Gordon. Z and HOL. In Bowen and Hall [67], pages 141–167.
66. J. P. Bowen and M. J. C. Gordon. A shallow embedding of Z in HOL. *Information and Software Technology*, 37(5):269–276, May 1995.
Revised version of [65].

67. J. P. Bowen and J. A. Hall, editors. *Z User Workshop, Cambridge 1994*, Workshops in Computing. Springer-Verlag, 1994.
Proceedings of the Eighth Annual Z User Meeting, St. John's College, Cambridge, UK. Published in collaboration with the British Computer Society. For individual papers, see [32, 65, 58, 57, 81, 103, 106, 151, 172, 174, 189, 210, 211, 215, 222, 298, 329, 393, 446, 504, 526, 531]. The proceedings also includes an *Introduction and Opening Remarks*, a *Select Z Bibliography* [58] and a section answering *Frequently Asked Questions* [57].
68. J. P. Bowen and M. G. Hinchey. Formal methods and safety-critical standards. *IEEE Computer*, 27(8):68–71, August 1994.
69. J. P. Bowen and M. G. Hinchey. Seven more myths of formal methods: Dispelling industrial prejudices. In Naftalin et al. [361], pages 105–117.
70. J. P. Bowen and M. G. Hinchey. Editorial. *Information and Software Technology*, 37(5):258–259, May 1995.
A special issue on Z. See [60, 71, 66, 78, 190, 310, 326, 331, 497].
71. J. P. Bowen and M. G. Hinchey. Report on Z User Meeting (ZUM'94). *Information and Software Technology*, 37(5):335–336, May 1995.
72. J. P. Bowen and M. G. Hinchey. Seven more myths of formal methods. *IEEE Software*, 12(4), July 1995.
This article deals with further myths in addition to those presented in [208]. Previous versions issued as:
 - Technical Report PRG-TR-7-94, Oxford University Computing Laboratory, June 1994.
 - Technical Report 357, University of Cambridge, Computer Laboratory, January 1995.
73. J. P. Bowen and M. G. Hinchey. Ten commandments of formal methods. *IEEE Computer*, 28(4):56–63, April 1995.
Previously issued as: Technical Report 350, University of Cambridge, Computer Laboratory, September 1994.
74. J. P. Bowen and M. G. Hinchey, editors. *ZUM'95 – 9th International Conference of Z Users, Limerick 1995*, Lecture Notes in Computer Science. Springer-Verlag, 1995.
75. J. P. Bowen and J. E. Nicholls, editors. *Z User Workshop, London 1992*, Workshops in Computing. Springer-Verlag, 1993.
Proceedings of the Seventh Annual Z User Meeting, DTI Offices, London, UK. Published in collaboration with the British Computer Society. For individual papers, see [27, 79, 118, 124, 134, 132, 155, 238, 261, 293, 307, 319, 332, 376, 382, 400, 420, 482, 499]. The proceedings also includes an *Introduction and Opening Remarks*, a *Select Z Bibliography* and a section answering *Frequently Asked Questions*.
76. J. P. Bowen and V. Stavridou. The industrial take-up of formal methods in safety-critical and other areas: A perspective. In Woodcock and Larsen [527], pages 183–195.
77. J. P. Bowen and V. Stavridou. Safety-critical systems, formal methods and standards. *IEE/BCS Software Engineering Journal*, 8(4):189–209, July 1993.
A survey on the use of formal methods, including B and Z, for safety-critical systems. Winner of the 1994 IEE Charles Babbage Premium award. A previous version is also available as Oxford University Computing Laboratory Technical Report PRG-TR-5-92.
78. J. P. Bowen, S. Stepney, and R. Barden. Annotated Z bibliography. *Information and Software Technology*, 37(5):317–332, May 1995.
Revised version of [461].
79. A. Bradley. Requirements for Defence Standard 00-55. In Bowen and Nicholls [75], pages 93–94.
80. P. T. Breuer. Z! in progress: Maintaining Z specifications. In Nicholls [373], pages 295–318.
81. P. T. Breuer and J. P. Bowen. Towards correct executable semantics for Z. In Bowen and Hall [67], pages 185–209.

82. S. M. Brien. The development of Z. In D. J. Andrews, J. F. Groote, and C. A. Middelburg, editors, *Semantics of Specification Languages (SoSL)*, Workshops in Computing, pages 1–14. Springer-Verlag, 1994.
83. S. M. Brien and J. E. Nicholls. Z base standard. Technical Monograph PRG-107, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, November 1992. Accepted for standardization under ISO/IEC JTC1/SC22.
This is the first publicly available version of the proposed ISO Z Standard. See also [457] for the current most widely available Z reference manual.
84. C. Britton, M. Loomes, and R. Mitchell. Formal specification as constructive diagrams. *Microprocessing and Microprogramming*, 37(1–5):175–178, January 1993.
85. D. J. Brown and J. P. Bowen. The Event Queue: An extensible input system for UNIX workstations. In *Proc. European Unix Users Group Conference*, pages 29–52. EUUG, May 1987.
Available from EUUG Secretariat, Owles Hall, Buntingford, Hertfordshire SG9 9PL, UK.
86. D. Brownbridge. Using Z to develop a CASE toolset. In Nicholls [371], pages 142–149.
87. A. Bryant. Structured methodologies and formal notations: Developing a framework for synthesis and investigation. In Nicholls [371], pages 229–241.
88. G. R. Buckberry. ZED: A Z notation editor and syntax analyser. *BCS FACS FACTS*, Series II, 2(3):13–23, November 1991.
89. A. Burns and I. W. Morrison. A formal description of the structure attribute model for tool interfacing. *IEE/BCS Software Engineering Journal*, 4(2):74–78, March 1989.
90. A. Burns and A. J. Wellings. Occam’s priority model and deadline scheduling. In *Proc. 7th Occam User Group Meeting, Grenoble*, 1987.
91. A. Burns and A. J. Wellings. A formal description of Ada tasking in Z. Computer Science Report YCS122, University of York, Heslington, York YO1 5DD, UK, 1989.
92. A. Burns and A. J. Wellings. Priority inheritance and message passing communication: A formal treatment. Computer Science Report YCS116, University of York, Heslington, York YO1 5DD, UK, 1989.
93. J. S. Busby and D. Hutchison. The practical integration of manufacturing applications. *Software Practice and Experience*, 22(2):183–207, 1992.
94. P. Butcher. A behavioural semantics for Linda-2. *IEE/BCS Software Engineering Journal*, 6(4):196–204, July 1991.
95. M. J. Butler. Service extension at the specification level. In Nicholls [373], pages 319–336.
96. M. J. Butler. Feature interaction analysis using Z. Åbo Akademi University, Finland, 1994.
97. M. J. Butler. Z specification of the X400 reliable transfer service. Åbo Akademi University, Finland, 1994.
98. D. Carrington. ZOOM workshop report. In Nicholls [375], pages 352–364.
This paper records the activities of a workshop on Z and object-oriented methods held in August 1992 at Oxford. A comprehensive bibliography is included.
99. D. Carrington, D. J. Duke, R. Duke, P. King, G. A. Rose, and G. Smith. Object-Z: An object-oriented extension to Z. In S. Vuong, editor, *Formal Description Techniques, II (FORTE’89)*, pages 281–296. Elsevier Science Publishers (North-Holland), 1990.
100. D. Carrington, D. J. Duke, I. J. Hayes, and J. Welsh. Deriving modular designs from formal specifications. *ACM Software Engineering Notes*, 18(5):89–98, December 1993.
101. D. Carrington, D. J. Duke, I. J. Hayes, and J. Welsh. Deriving modular designs from formal specifications: The analysis phase. Technical Report 93-13, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, October 1993.
102. D. Carrington and G. Smith. Extending Z for object-oriented specifications. In *Proc. 5th Australian Software Engineering Conference (ASWEC’90)*, pages 9–14, 1990.

103. D. Carrington and P. Stocks. A tale of two paradigms: Formal methods and software testing. In Bowen and Hall [67], pages 51–68.
Also available as Technical Report 94-4, Department of Computer Science, University of Queensland, 1994.
104. D. Carrington and P. Stocks. A tale of two paradigms: Formal methods and software testing. Technical Report 94-4, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, February 1994.
105. P. Chalin and P. Grogono. Z specification of an object manager. In Bjørner et al. [39], pages 41–71.
106. D. K. C. Chan and P. W. Trinder. An object-oriented data model supporting multi-methods, multiple inheritance, and static type checking: A specification in Z. In Bowen and Hall [67], pages 297–315.
107. W. Chantatub and M. Holcombe. Software testing strategies for software requirements and design. In *Proc. EuroSTAR'94*, pages 40/1–40/29, 3000-2 Hartley Road, Jacksonville, Florida 32257, USA, 1994. Software Quality Engineering.
The paper describes how to construct a detailed Z specification using traditional software engineering techniques (ERDs, DFDs, etc.) in a top down manner. It introduces a number of notational devices to help with the management of large Z specifications. Some issues about proving consistency between levels are also addressed.
108. A. S. K. Cheng, J. Han, J. Welsh, and A. Wood. Providing user-oriented support for software development by formal methods. Technical Report 92-8, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, December 1992.
109. S. J. Clarke, A. C. Combes, and J. A. McDermid. The analysis of safety arguments in the specification of a motor speed control loop. Computer Science Report YCS136, University of York, Heslington, York YO1 5DD, UK, 1990.
This report describes some timing extensions to Z.
110. B. Cohen. Justification of formal methods for system specifications & A rejustification of formal notations. *IEE/BCS Software Engineering Journal*, 4(1):26–38, January 1989.
111. B. Cohen and D. Mannering. The rigorous specification and verification of the safety aspects of a real-time system. In *COMPASS '90*, 1990.
112. B. P. Collins, J. E. Nicholls, and I. H. Sørensen. Introducing formal methods: The CICS experience with Z. In B. Neumann et al., editors, *Mathematical Structures for Software Engineering*. Oxford University Press, 1991.
113. J. Cooke. Editorial – formal methods: What? why? and when? *The Computer Journal*, 35(5):417–418, October 1992.
An editorial introduction to two special issues on *Formal Methods*. See also [31, 114, 344, 431, 524] for papers relevant to Z.
114. J. Cooke. Formal methods – mathematics, theory, recipes or what? *The Computer Journal*, 35(5):419–423, October 1992.
115. A. C. Coombes, L. Barroca, J. S. Fitzgerald, J. A. McDermid, L. Spencer, and A. Saeed. Formal specification of an aerospace system: The attitude monitor. In Hinchey and Bowen [245], pages 307–332.
116. A. C. Coombes and J. A. McDermid. A tool for defining the architecture of Z specifications. In Nicholls [373], pages 77–92.
117. A. C. Coombes and J. A. McDermid. Specifying temporal requirements for distributed real-time systems in Z. Computer Science Report YCS176, University of York, Heslington, York YO1 5DD, UK, 1992.
118. A. C. Coombes and J. A. McDermid. Using diagrams to give a formal specification of timing constraints in Z. In Bowen and Nicholls [75], pages 119–130.
119. D. Cooper. Educating management in Z. In Nicholls [371], pages 192–194.

120. V. A. O. Cordeiro, A. C. A. Sampaio, and S. L. Meira. From MooZ to Eiffel – a rigorous approach to system development. In Naftalin et al. [361], pages 306–325.
121. S. Craggs and J. B. Wordsworth. Hursley Lab wins another Queen’s Award & Hursley and Oxford – a marriage of minds & Z stands for quality. *Developments, IBM Hursley Park*, 8:1–2, 21 April 1992.
122. I. Craig. *The Formal Specification of Advanced AI Architectures*. AI Series. Ellis Horwood, September 1991.
This book contains two rather large (and relatively complete) specifications of Artificial Intelligence (AI) systems using Z. The architectures are the blackboard and Cassandra architectures. As well as showing that formal specification *can* be used in AI at the architecture level, the book is intended as a case-studies book, and also contains introductory material on Z (for AI people). The book assumes a knowledge of Z, so for non-AI people its primary use is for the presentation of the large specifications. The blackboard specification, with explanatory text, is around 100 pages.
123. D. Craigen, S. Gerhart, and T. Ralston. Formal methods technology transfer: Impediments and innovation. In Hinchey and Bowen [245], pages 399–419.
124. D. Craigen, S. L. Gerhart, and T. Ralston. An international survey of industrial applications of formal methods. In Bowen and Nicholls [75], pages 1–5.
125. D. Craigen, S. L. Gerhart, and T. J. Ralston. Formal methods reality check: Industrial usage. In Woodcock and Larsen [527], pages 250–267.
Revised version in [127].
126. D. Craigen, S. L. Gerhart, and T. J. Ralston. An international survey of industrial applications of formal methods. Technical Report NIST GCR 93/626-V1 & 2, Atomic Energy Control Board of Canada, US National Institute of Standards and Technology, and US Naval Research Laboratories, 1993.
Volume 1: Purpose, Approach, Analysis and Conclusions; Volume 2: Case Studies. Order numbers: PB93-178556/AS & PB93-178564/AS; National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161, USA.
127. D. Craigen, S. L. Gerhart, and T. J. Ralston. Formal methods reality check: Industrial usage. *IEEE Transactions on Software Engineering*, 21(2):90–98, 1995.
Revised version of [262].
128. D. Craigen, S. Kromodimoeljo, I. Meisels, W. Pase, and M. Saaltink. EVES: An overview. In Prehn and Toetenel [398], pages 389–405.
129. S. Croxall, P. Lupton, and J. B. Wordsworth. A formal specification of the CPI communications. IBM Technical Report TR12.277, IBM United Kingdom Laboratories Ltd, Hursley Park, Winchester, Hampshire SO21 2JN, UK, 1990.
130. E. Cusack. Inheritance in object oriented Z. In P. America, editor, *Proc. ECOOP’91 European Conference on Object-Oriented Programming*, volume 512 of *Lecture Notes in Computer Science*, pages 167–179. Springer-Verlag, 1991.
131. E. Cusack. Object oriented modelling in Z for open distributed systems. In J. de Meer, editor, *Proc. International Workshop on ODP*. Elsevier Science Publishers (North-Holland), 1992.
132. E. Cusack. Using Z in communications engineering. In Bowen and Nicholls [75], pages 196–202.
133. E. Cusack and M. Lai. Object oriented specification in LOTOS and Z (or my cat really is object oriented!). In J. W. de Bakker, W. P. de Roever, and G. Rozenberg, editors, *REX/FOOL School/Workshop on Foundations of Object-Oriented Languages*, volume 489 of *Lecture Notes in Computer Science*, pages 179–202. Springer-Verlag, 1990.
134. E. Cusack and C. Wezeman. Deriving tests for objects specified in Z. In Bowen and Nicholls [75], pages 180–195.

135. R. S. M. de Barros. Formal specification of relational database applications: A method and an example. Research report GE-93-02, Department of Computing Science, University of Glasgow, UK, September 1993.
136. R. S. M. de Barros. Deriving relational database programs from formal specifications. In Naftalin et al. [361], pages 703–723.
137. R. S. M. de Barros. On the derivation of relational database programs from formal specifications. Research report GE-94-01, Department of Computing Science, University of Glasgow, UK, July 1994.
An extended version of [136].
138. R. S. M. de Barros and D. J. Harper. Formal development of relational database applications. In D. J. Harper and M. C. Norrie, editors, *Specifications of Database Systems, Glasgow 1991*, Workshops in Computing, pages 21–43. Springer-Verlag, 1992.
Zc, a Z-like formalism, is used.
139. R. S. M. de Barros and D. J. Harper. A method for the specification of relational database applications. In Nicholls [375], pages 261–286.
140. A. M. L. de Vasconcelos and J. A. McDermid. Incremental type-checking in Z. Computer Science Report YCS185, University of York, Heslington, York YO1 5DD, UK, 1992.
141. C. N. Dean and M. G. Hinchey. Introducing formal methods through rôle-playing. *ACM SIGCSE Bulletin*, 27(1):302–306, March 1995.
142. B. Dehbonei and F. Mejia. Formal methods in the railways signalling industry. In Naftalin et al. [361], pages 26–34.
143. N. Delisle and D. Garlan. Formally specifying electronic instruments. In *Proc. Fifth International Workshop on Software Specification and Design*. IEEE Computer Society, May 1989. Also published in ACM SIGSOFT Software Engineering Notes 14(3).
144. N. Delisle and D. Garlan. A formal specification of an oscilloscope. *IEEE Software*, 7(5):29–36, September 1990.
Unlike most work on the application of formal methods, this research uses formal methods to gain insight into system architecture. The context for this case study is electronic instrument design.
145. R. Di Giovanni and P. L. Iachini. HOOD and Z for the development of complex systems. In Bjørner et al. [39], pages 262–289.
146. A. J. J. Dick, P. J. Krause, and J. Cozens. Computer aided transformation of Z into Prolog. In Nicholls [371], pages 71–85.
147. A. Diller. Specifying interactive programs in Z. Research Report CSR-90-13, School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK, August 1990.
148. A. Diller. *Z: An Introduction to Formal Methods*. John Wiley & Sons, 1990.
This book offers a comprehensive tutorial to Z from the practical viewpoint. Many natural deduction style proofs are presented and exercises are included. A second edition is now available [150].
149. A. Diller. Z and Hoare logics. In Nicholls [375], pages 59–76.
150. A. Diller. *Z: An Introduction to Formal Methods*. John Wiley & Sons, Chichester, UK, 2nd edition, 1994.
This book offers a comprehensive tutorial to Z from the practical viewpoint. Many natural deduction style proofs are presented and exercises are included. Z as defined in the 2nd edition of *The Z Notation* [457] is used throughout.
Contents: Tutorial introduction; Methods of reasoning; Case studies; Specification animation; Reference manual; Answers to exercises; Glossaries of terms and symbols; Bibliography.
151. A. Diller and R. Docherty. Z and abstract machine notation: A comparison. In Bowen and Hall [67], pages 250–263.

152. A. J. Dix. *Formal Methods for Interactive Systems*. Computers and People Series. Academic Press, 1991.
153. A. J. Dix, J. Finlay, G. D. Abowd, and R. Beale. *Human-Computer Interaction*. Prentice Hall International, 1993.
154. V. Doma and R. Nicholl. EZ: A system for automatic prototyping of Z specifications. In Prehn and Toetenel [398], pages 189–203.
155. C. Draper. Practical experiences of Z and SSADM. In Bowen and Nicholls [75], pages 240–251.
156. D. A. Duce, D. J. Duke, P. J. W. ten Hagen, and G. J. Reynolds. PREMO - an initial approach to a formal definition. *Computer Graphics Forum*, 13(3):C-393–C-406, 1994. PREMO (Presentation Environments for Multimedia Objects) is a work item proposal by the ISO/IEC JTC11/SC24 committee, which is responsible for international standardization in the area of computer graphics and image processing.
157. D. J. Duke. Structuring Z specifications. In *Proc. 14th Australian Computer Science Conference*, 1991.
158. D. J. Duke. Enhancing the structures of Z specifications. In Nicholls [375], pages 329–351.
159. D. J. Duke. *Object-Oriented Formal Specification*. PhD thesis, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, 1992.
160. D. J. Duke and R. Duke. A history model for classes in Object-Z. Technical Report 120, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, 1989.
161. D. J. Duke and R. Duke. Towards a semantics for Object-Z. In Bjørner et al. [39], pages 244–261.
162. D. J. Duke and M. D. Harrison. Event model of human-system interaction. *IEEE/BCS Software Engineering Journal*, 10(1):3–12, January 1995.
163. D. J. Duke and M. D. Harrison. Mapping user requirements to implementations. *IEEE/BCS Software Engineering Journal*, 10(1):13–20, January 1995.
164. R. Duke and D. J. Duke. Aspects of object-oriented formal specification. In *Proc. 5th Australian Software Engineering Conference (ASWEC'90)*, pages 21–26, 1990.
165. R. Duke, I. J. Hayes, P. King, and G. A. Rose. Protocol specification and verification using Z. In S. Aggarwal and K. Sabnani, editors, *Protocol Specification, Testing, and Verification VIII*, pages 33–46. Elsevier Science Publishers (North-Holland), 1988.
166. R. Duke, P. King, G. A. Rose, and G. Smith. The Object-Z specification language. In T. Korson, V. Vaishnavi, and B. Meyer, editors, *Technology of Object-Oriented Languages and Systems: TOOLS 5*, pages 465–483. Prentice Hall, 1991.
167. R. Duke, P. King, G. A. Rose, and G. Smith. The Object-Z specification language: Version 1. Technical Report 91-1, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, April 1991. The most complete (and currently the standard) reference on Object-Z. It has been reprinted by ISO JTC1 WG7 as document number 372. A condensed version of this report was published as [166].
168. R. Duke and G. A. Rose. A complete Z specification of an interactive program editor. Technical Report 71, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, 1986.
169. R. Duke, G. A. Rose, and A. Lee. Object-oriented protocol specification. In L. Logrippo, R. L. Probert, and H. Ural, editors, *Protocol Specification, Testing, and Verification X*, pages 325–338. Elsevier Science Publishers (North-Holland), 1990.
170. R. Duke and G. Smith. Temporal logic and Z specifications. *Australian Computer Journal*, 21(2):62–69, May 1989.
171. D. Edmond. *Information Modeling: Specification and Implementation*. Prentice Hall, 1992.

172. M. Engel. Specifying real-time systems with Z and the Duration Calculus. In Bowen and Hall [67], pages 282–294.
173. A. S. Evans. Specifying & verifying concurrent systems using Z. In Naftalin et al. [361], pages 366–400.
174. A. S. Evans. Visualising concurrent Z specifications. In Bowen and Hall [67], pages 269–281.
175. P. C. Fencott, A. J. Galloway, M. A. Lockyer, S. J. O’Brien, and S. Pearson. Formalising the semantics of Ward/Mellor SA/RT essential models using a process algebra. In Naftalin et al. [361], pages 681–702.
176. N. E. Fenton and D. Mole. A note on the use of Z for flowgraph transformation. *Information and Software Technology*, 30(7):432–437, 1988.
177. E. Fergus and D. C. Ince. Z specifications and modal logic. In P. A. V. Hall, editor, *Proc. Software Engineering 90*, volume 1 of *British Computer Society Conference Series*. Cambridge University Press, 1990.
178. C. J. Fidge. Specification and verification of real-time behaviour using Z and RTL. In J. Vytupil, editor, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, Lecture Notes in Computer Science, pages 393–410. Springer-Verlag, 1992.
179. C. J. Fidge. Real-time refinement. In Woodcock and Larsen [527], pages 314–331.
180. C. J. Fidge. Adding real time to formal program development. In Naftalin et al. [361], pages 618–638.
181. C. J. Fidge. Proof obligations for real-time refinement. In Till [490], pages 279–305.
182. C. J. Fidge, P. Kearney, and J. Staples. Formally verified real-time software: an integrated development strategy (extended version). Technical Report 93-10, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, June 1993.
183. M. Flynn, T. Hoverd, and D. Brazier. Formaliser – an interactive support tool for Z. In Nicholls [371], pages 128–141.
184. I. Fogg, B. Hicks, A. Lister, T. Mansfield, and K. Raymond. A comparison of LOTOS and Z for specifying distributed systems. *Australian Computer Science Communications*, 12(1):88–96, February 1990.
185. D. C. Fowler, P. A. Swatman, and P. M. C. Swatman. Implementing EDI in the public sector: Including formality for enhanced control. In *Proc. 7th International Conference on Electronic Data Interchange*, June 1993.
186. N. E. Fuchs. Specifications are (preferably) executable. *IEE/BCS Software Engineering Journal*, 7(5):323–334, September 1992.
187. P. H. B. Gardiner, P. J. Lupton, and J. C. P. Woodcock. A simpler semantics for Z. In Nicholls [373], pages 3–11.
188. D. Garlan. The role of reusable frameworks. *ACM SIGSOFT Software Engineering Notes*, 15(4):42–44, September 1990.
189. D. Garlan. Integrating formal methods into a professional master of software engineering program. In Bowen and Hall [67], pages 71–85.
190. D. Garlan. Making formal methods effective for professional software engineers. *Information and Software Technology*, 37(5):261–268, May 1995.
Revised version of [189].
191. D. Garlan and N. Delisle. Formal specifications as reusable frameworks. In Bjørner et al. [39], pages 150–163.
192. D. Garlan and N. Delisle. Formal specification of an architecture for a family of instrumentation systems. In Hinchey and Bowen [245], pages 55–72.
193. D. Garlan and D. Notkin. Formalizing design spaces: Implicit invocation mechanisms. In Prehn and Toetenel [398], pages 31–45.

194. S. L. Gerhart. Applications of formal methods: Developing virtuoso software. *IEEE Software*, 7(5):6–10, September 1990.
This is an introduction to a special issue on Formal Methods with an emphasis on Z in particular. It was published in conjunction with special Formal Methods issues of *IEEE Transactions on Software Engineering* and *IEEE Computer*. See also [144, 208, 362, 455, 508].
195. S. L. Gerhart, D. Craigen, and T. Ralston. Observations on industrial practice using formal methods. In *Proc. 15th International Conference on Software Engineering (ICSE), Baltimore, Maryland, USA*, May 1993.
196. S. L. Gerhart, D. Craigen, and T. Ralston. Experience with formal methods in critical systems. *IEEE Software*, 11(1):21–28, January 1994.
Several commercial and exploratory cases in which Z features heavily are briefly presented on page 24. See also [294].
197. S. Gilmore. Correctness-oriented approaches to software development. Technical Report ECS-LFCS-91-147 (also CST-76-91), Department of Computer Science, University of Edinburgh, Edinburgh EH9 3JZ, UK, 1991.
This PhD thesis provides a critical evaluation of Z, VDM and algebraic specifications.
198. R. B. Gimson. The formal documentation of a Block Storage Service. Technical Monograph PRG-62, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, August 1987.
199. R. B. Gimson and C. C. Morgan. Ease of use through proper specification. In D. A. Duce, editor, *Distributed Computing Systems Programme*. Peter Peregrinus, London, 1984.
200. R. B. Gimson and C. C. Morgan. The Distributed Computing Software project. Technical Monograph PRG-50, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, July 1985.
201. J. Ginbayashi. Analysis of business processes specified in Z against an E-R data model. Technical Monograph PRG-103, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, December 1992.
202. H. S. Goodman. Animating Z specifications in Haskell using a monad. Technical Report CSR-93-10, School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK, November 1993.
203. R. Gotzhein. Specifying open distributed systems with Z. In Bjørner et al. [39], pages 319–339.
204. A. M. Gravell. Minimisation in formal specification and design. In Nicholls [371], pages 32–45.
205. A. M. Gravell. What is a good formal specification? In Nicholls [373], pages 137–150.
206. H. Habrias, S. Dunne, and B. Stoddart. NIAM and Z specifications. Rapport de recherche IRIN 32, Institut de Recherche en Informatique de Nantes, IUT de Nantes, Département Informatique, 3 rue du Maréchal Joffre, 44041 Nantes Cedex 01, France, May 1993.
207. F. Halasz and M. Schwartz. The Dexter hypertext reference model. In *NIST Hypertext Standardization Workshop*, January 1990.
208. J. A. Hall. Seven myths of formal methods. *IEEE Software*, 7(5):11–19, September 1990.
Formal methods are difficult, expensive, and not widely useful, detractors say. Using a case study and other real-world examples, this article challenges such common myths.
209. J. A. Hall. Using Z as a specification calculus for object-oriented systems. In Bjørner et al. [39], pages 290–318.
210. J. A. Hall. Specifying and interpreting class hierarchies in Z. In Bowen and Hall [67], pages 120–138.
211. J. G. Hall and J. A. McDermid. Towards a Z method: Axiomatic specification in Z. In Bowen and Hall [67], pages 213–229.

212. P. A. V. Hall. Towards testing with respect to formal specification. In *Proc. Second IEE/BCS Conference on Software Engineering*, number 290 in Conference Publication, pages 159–163. IEE/BCS, July 1988.
213. U. Hamer and J. Peleska. Z applied to the A330/340 CICS cabin communication system. In Hinchey and Bowen [245], pages 253–284.
214. V. Hamilton. The use of Z within a safety-critical software system. In Hinchey and Bowen [245], pages 357–374.
215. J. A. R. Hammond. Producing Z specifications from object-oriented analysis. In Bowen and Hall [67], pages 316–336.
216. J. A. R. Hammond. Z. In J. J. Marciniak, editor, *Encyclopedia of Software Engineering*, volume 2, pages 1452–1453. John Wiley & Sons, 1994.
217. M. D. Harrison. Engineering human-error tolerant software. In Nicholls [375], pages 191–204.
218. C. L. Harrold. Formal specification of a secure document control system for SMITE. Report no. 88002, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, February 1988.
219. K. Harwood, P. A. Lindsay, and R. Matthews. An approach to constructing verified software. Technical Report 93-8, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, September 1993.
220. W. T. Harwood. Proof rules for Balzac. Technical Report WTH/P7/001, Imperial Software Technology, Cambridge, UK, 1991.
221. W. Hasselbring. A formal Z specification of ProSet-Linda. Technical report, University of Essen, Fachbereich Mathematik und Informatik – Software Engineering, Schuetzenbahn 70, 4300 Essen 1, Germany, 1992.
222. W. Hasselbring. Animation of Object-Z specifications with a set-oriented prototyping language. In Bowen and Hall [67], pages 337–356.
223. W. Hasselbring. *Prototyping Parallel Algorithms in a Set-Oriented Language*. Dissertation (University of Dortmund, Dept. Computer Science). Verlag Dr. Kovac, Hamburg, Germany, 1994.
This book presents the design and implementation of an approach to prototyping parallel algorithms with ProSet-Linda. The presented approach to designing and implementing ProSet-Linda relies on the use of the formal specification language Object-Z and the prototyping language ProSet itself.
224. H. P. Haughton. Using Z to model and analyse safety and liveness properties of communication protocols. *Information and Software Technology*, 33(8):575–580, October 1991.
225. H. P. Haughton and K. C. Lano. Three dimensional maintenance. In M. Munro and P. Carroll, editors, *Fourth Software Maintenance Workshop Notes*. Centre for Software Maintenance, Durham, UK, 18–20 September 1990.
This paper presents an object-oriented extension to Z with the aim to aid software maintenance.
226. I. J. Hayes. Applying formal specification to software development in industry. *IEEE Transactions on Software Engineering*, 11(2):169–178, February 1985.
227. I. J. Hayes. Specification directed module testing. *IEEE Transactions on Software Engineering*, 12(1):124–133, January 1986.
228. I. J. Hayes. Using mathematics to specify software. In *Proc. First Australian Software Engineering Conference*. Institution of Engineers, Australia, May 1986.
229. I. J. Hayes. A generalisation of bags in Z. In Nicholls [371], pages 113–127.
230. I. J. Hayes. Multi-sets and multi-relations in Z with an application to a bill-of-materials system. Technical Report UQCS-176, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, July 1990.

The first chapter is an updated version of [229] and the second and third chapters are updated versions of [233]. The changes are mainly for consistency with [235, 457].

231. I. J. Hayes. Specifying physical limitations: A case study of an oscilloscope. Technical Report 167, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, July 1990.
232. I. J. Hayes. Interpretations of Z schema operators. In Nicholls [373], pages 12–26.
233. I. J. Hayes. Multi-relations in Z: A cross between multi-sets and binary relations. *Acta Informatica*, 29(1):33–62, February 1992.
234. I. J. Hayes. VDM and Z: A comparative case study. *Formal Aspects of Computing*, 4(1):76–99, 1992.
235. I. J. Hayes, editor. *Specification Case Studies*. Prentice Hall International Series in Computer Science, 2nd edition, 1993.
This is a revised edition of the first ever book on Z, originally published in 1987; it contains substantial changes to every chapter. The notation has been revised to be consistent with *The Z Notation: A Reference Manual* by Mike Spivey [457]. The CAVIAR chapter has been extensively changed to make use of a form of modularization.
Divided into four sections, the first provides tutorial examples of specifications, the second is devoted to the area of software engineering, the third covers distributed computing, analyzing the role of mathematical specification, and the fourth part covers the IBM CICS transaction processing system. Appendices include comprehensive glossaries of the Z mathematical and schema notation. The book will be of interest to the professional software engineer involved in designing and specifying large software projects.
The other contributors are W. Flinn, R. B. Gimson, S. King, C. C. Morgan, I. H. Sørensen and B. A. Sufrin.
236. I. J. Hayes and C. B. Jones. Specifications are not (necessarily) executable. *IEE/BCS Software Engineering Journal*, 4(6):330–338, November 1989.
237. I. J. Hayes, C. B. Jones, and J. E. Nicholls. Understanding the differences between VDM and Z. *FACS Europe*, Series I, 1(1):7–30, Autumn 1993.
Also available as Technical Report UMCS-93-8-1, Department of Computer Science, University of Manchester, UK, 1993.
238. I. J. Hayes and L. Wildman. Towards libraries for Z. In Bowen and Nicholls [75], pages 9–36.
239. He Jifeng, C. A. R. Hoare, M. Fränzle, Markus Müller-Ulm, E.-R. Olderog, M. Schenke, A. P. Ravn, and H. Rischel. Provably correct systems. In H. Langmaack, W.-P. de Roever, and J. Vytupil, editors, *Formal Techniques in Real Time and Fault Tolerant Systems*, volume 863 of *Lecture Notes in Computer Science*, pages 288–335. Springer-Verlag, 1994.
240. He Jifeng, C. A. R. Hoare, and J. W. Sanders. Data refinement refined. In B. Robinet and R. Wilhelm, editors, *Proc. ESOP 86*, volume 213 of *Lecture Notes in Computer Science*, pages 187–196. Springer-Verlag, 1986.
241. D. Heath, D. Allum, and L. Dunckley. *Introductory Logic and Formal Methods*. A. Waller, Henley-on-Thames, UK, 1994.
242. B. Hepworth. ZIP: A unification initiative for Z standards, methods and tools. In Nicholls [371], pages 253–259.
243. B. Hepworth and D. Simpson. The ZIP project. In Nicholls [373], pages 129–133.
244. M. G. Hinchey. Formal methods for system specification: An ounce of prevention is worth a pound of cure. *IEEE Potentials Magazine*, 12(3):50–52, October 1993.
245. M. G. Hinchey and J. P. Bowen, editors. *Applications of Formal Methods*. Prentice Hall International Series in Computer Science, 1995.
A collection on industrial examples of the use of formal methods. Chapters relevant to Z include [115, 123, 192, 213, 214, 246, 330].

246. M. G. Hinchey and J. P. Bowen. Applications of formal methods FAQ. In *Applications of Formal Methods* [245], pages 1–15.
247. I. S. C. Houston. The CICS application programming interface: Automatic transaction initiation. IBM Technical Report TR12.300, IBM United Kingdom Laboratories Ltd, Hursley Park, Winchester, Hampshire SO21 2JN, UK, December 1990.
One of a number of reports on the CICS application programming interface. See also [41, 289, 360].
248. I. S. C. Houston and M. Josephs. Specifying distributed CICS in Z: Accessing local and remote resources (short communication). *Formal Aspects of Computing*, 6(6):569–579, 1994.
249. I. S. C. Houston and S. King. CICS project report: Experiences and results from the use of Z in IBM. In Prehn and Toetenel [398], pages 588–596.
250. I. S. C. Houston and J. B. Wordsworth. A Z specification of part of the CICS file control API. IBM Technical Report TR12.272, IBM United Kingdom Laboratories Ltd, Hursley Park, Winchester, Hampshire SO21 2JN, UK, 1990.
251. A. D. Hutcheon and A. J. Wellings. Specifying restrictions on imperative programming languages for use in a distributed embedded environment. *IEE/BCS Software Engineering Journal*, 5(2):93–104, March 1990.
252. P. L. Iachini. Operation schema iterations. In Nicholls [373], pages 50–57.
253. M. Imperato. *An Introduction to Z*. Chartwell-Bratt, 1991.
Contents: Introduction; Set theory; Logic; Building Z specifications; Relations; Functions; Sequences; Bags; Advanced Z; Case study: a simple banking system.
254. D. C. Ince. Z and system specification. In D. C. Ince and D. Andrews, editors, *The Software Life Cycle*, chapter 12, pages 260–277. Butterworths, 1990.
255. D. C. Ince. *An Introduction to Discrete Mathematics, Formal System Specification and Z*. Oxford Applied Mathematics and Computing Science Series. Oxford University Press, 2nd edition, 1993.
256. INMOS Limited. Specification of instruction set & Specification of floating point unit instructions. In *Transputer Instruction Set – A compiler writer’s guide*, pages 127–161. Prentice Hall, 1988.
Appendices F and G use a Z-like notation to give a specification of the instruction set of the IMS T212 and T414 transputers, and the T800 floating-point transputer.
257. A. Jack. It’s hard to explain, but Z is much clearer than English. *Financial Times*, page 22, 21 April 1992.
258. D. Jackson. Abstract model checking of infinite specifications. In Naftalin et al. [361], pages 519–531.
259. D. Jackson. Structuring Z specifications with views. Technical Report CMU-CS-94-126, Carnegie-Mellon University, USA, March 1994.
260. J. Jacky. Formal specifications for a clinical cyclotron control system. *ACM SIGSOFT Software Engineering Notes*, 15(4):45–54, September 1990.
261. J. Jacky. Formal specification and development of control system input/output. In Bowen and Nicholls [75], pages 95–108.
262. J. Jacky. Specifying a safety-critical control system in Z. In Woodcock and Larsen [527], pages 388–402.
Revised version in [263].
263. J. Jacky. Specifying a safety-critical control system in Z. *IEEE Transactions on Software Engineering*, 21(2):99–106, 1995.
Revised version of [262].
264. J. Jacob. The varieties of refinements. In Morris and Shaw [359], pages 441–455.

265. Jin Song Dong and R. Duke. An object-oriented approach to the formal specification of ODP trader. In *Proc. IFIP TC6/WG6.1 International Conference on Open Distributed Processing*, pages 341–352, September 1993.
266. Jin Song Dong, R. Duke, and G. A. Rose. An object-oriented approach to the semantics of programming languages. In *Proc. 17th Australian Computer Science Conference (ACSC-17)*, pages 767–775, January 1994.
267. C. W. Johnson. Using Z to support the design of interactive safety-critical systems. *IEE/BCS Software Engineering Journal*, 10(2):49–60, March 1995.
268. M. Johnson and P. Sanders. From Z specifications to functional implementations. In Nicholls [371], pages 86–112.
269. P. Johnson. Using Z to specify CICS. In *Proc. Software Engineering anniversary meeting (SEAS)*, page 303, 1987.
270. W. Johnston and G. A. Rose. Guidelines for the manual conversion of Object-Z to C++. Technical Report 93-14, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, September 1993.
271. C. B. Jones. Interference revisited. In Nicholls [373], pages 58–73.
272. C. B. Jones, R. C. Shaw, and T. Denvir, editors. *5th Refinement Workshop*, Workshop in Computing. Springer-Verlag, 1992.
The workshop was held at Lloyd’s Register, London, UK, 8–10 January 1992. See [435].
273. R. B. Jones. ICL ProofPower. *BCS FACS FACTS*, Series III, 1(1):10–13, Winter 1992.
274. D. Jordan, J. A. McDermid, and I. Toyn. CADiZ – computer aided design in Z. In Nicholls [373], pages 93–104.
275. L. E. Jordan. The kernel Z type checking rules. Technical Report LEJ/TC3/001, Imperial Software Technology, Cambridge, UK, 1991.
276. L. E. Jordan. The Z syntax supported by Balzac-II/1. Technical Report LEJ/S1/001, Imperial Software Technology, Cambridge, UK, 1991.
277. M. B. Josepfs. The data refinement calculator for Z specifications. *Information Processing Letters*, 27(1):29–33, 1988.
278. M. B. Josepfs. A state-based approach to communicating processes. *Distributed Computing*, 3:9–18, 1988.
A theoretical paper on combining features of CSP and Z.
279. M. B. Josepfs. Specifying reactive systems in Z. Technical Report PRG-TR-19-91, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, July 1991.
280. M. B. Josepfs and D. Redmond-Pyle. Entity-relationship models expressed in Z: A synthesis of structured and formal methods. Technical Report PRG-TR-20-91, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, July 1991.
281. M. B. Josepfs and D. Redmond-Pyle. A library of Z schemas for use in entity-relationship modelling. Technical Report PRG-TR-21-91, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, August 1991.
282. D. H. Kemp. Specification of Viper1 in Z. Memorandum no. 4195, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, October 1988.
283. D. H. Kemp. Specification of Viper2 in Z. Memorandum no. 4217, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, October 1988.
284. H. Kilov. Information modeling and Object Z: Specifying generic reusable associations. In O. Etzion and A. Segev, editors, *Proc. NGITS-93 (Next Generation Information Technology and Systems)*, pages 182–191, June 1993.
285. H. Kilov and J. Ross. Declarative specifications of collective behavior: Generic reusable frameworks. In H. Kilov and W. Harvey, editors, *Proc. Workshop on Specification of Behavioral Semantics in Object-Oriented Information Modeling*, pages 71–75, Institute for

- Information Management and Department of Computer and Information Systems, Robert Morris College, Coraopolos and Pittsburgh, Pennsylvania, USA, 1993. OOPSLA.
286. H. Kilov and J. Ross. Appendix A: A more formal approach. In *Information Modeling: An Object-Oriented Approach*, Object-Oriented Series, pages 199–207. Prentice Hall, 1994.
 287. H. Kilov and J. Ross. *Information Modeling: An Object-Oriented Approach*. Object-Oriented Series. Prentice Hall, 1994.
 288. P. King. Printing Z and Object-Z \LaTeX documents. Department of Computer Science, University of Queensland, May 1990.
A description of a Z style option ‘oz.sty’, an extended version of Mike Spivey’s ‘zed.sty’ [454], for use with the \LaTeX document preparation system [297]. It is particularly useful for printing Object-Z documents [99, 161].
 289. S. King. The CICS application programming interface: Program control. IBM Technical Report TR12.302, IBM United Kingdom Laboratories Ltd, Hursley Park, Winchester, Hampshire SO21 2JN, UK, December 1990.
One of a number of reports on the CICS application programming interface. See also [41, 247, 360].
 290. S. King. Z and the refinement calculus. In Bjørner et al. [39], pages 164–188.
Also published as Technical Monograph PRG-79, Oxford University Computing Laboratory, February 1990.
 291. S. King and I. H. Sørensen. Specification and design of a library system. In McDermid [337].
 292. S. King, I. H. Sørensen, and J. C. P. Woodcock. Z: Grammar and concrete and abstract syntaxes. Technical Monograph PRG-68, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, 1988.
 293. J. C. Knight and D. M. Kienzle. Preliminary experience using Z to specify a safety-critical system. In Bowen and Nicholls [75], pages 109–118.
 294. J. C. Knight and B. Littlewood. Critical task of writing dependable software. *IEEE Software*, 11(1):16–20, January 1994.
Guest editors’ introduction to a special issue of *IEEE Software on Safety-Critical Systems*. A short section on formal methods mentions several Z books on page 18. See also [196].
 295. R. D. Knott and P. J. Krause. The implementation of Z specifications using program transformation systems: The SuZan project. In C. Rattray and R. G. Clark, editors, *The Unified Computation Laboratory*, volume 35 of *IMA Conference Series*, pages 207–220, Oxford, UK, 1992. Clarendon Press.
 296. M. K. F. Lai. A formal interpretation of the MAA standard in Z. Technical Report DITC 184/91, National Physical Laboratory, Teddington, Middlesex, UK, June 1991.
 297. L. Lamport. *\LaTeX User’s Guide & Reference Manual*. Addison-Wesley Publishing Company, Reading, Massachusetts, USA, 1986.
Z specifications may be produced using the document preparation system \LaTeX together with a special \LaTeX style option. The most widely used style files are fuzz.sty [456], zed.sty [454] and oz.sty [288].
 298. L. Lamport. TLZ. In Bowen and Hall [67], pages 267–268. Abstract.
 299. K. C. Lano. Z^{++} , an object-orientated extension to Z. In Nicholls [373], pages 151–172.
 300. K. C. Lano. Refinement in object-oriented specification languages. In Till [490], pages 236–259.
 301. K. C. Lano and P. T. Breuer. From programs to Z specifications. In Nicholls [371], pages 46–70.
 302. K. C. Lano, P. T. Breuer, and H. P. Haughton. Reverse engineering COBOL via formal methods. *Software Maintenance: Research and Practice*, 5:13–35, 1993.
Also published in a shortened form as Chapter 16 in [502].

303. K. C. Lano and H. P. Haughton. An algebraic semantics for the specification language Z^{++} . In *Proc. Algebraic Methodology and Software Technology Conference (AMAST '91)*. Springer-Verlag, 1992.
304. K. C. Lano and H. P. Haughton. Reasoning and refinement in object-oriented specification languages. In O. L. Madsen, editor, *ECOOP '92: European Conference on Object-Oriented Programming*, volume 615 of *Lecture Notes in Computer Science*, pages 78–97. Springer-Verlag, 1992.
305. K. C. Lano and H. P. Haughton. *The Z^{++} Manual*. Lloyd's Register of Shipping, 29 Wellesley Road, Croydon CRO 2AJ, UK, 1992.
306. K. C. Lano and H. P. Haughton, editors. *Object Oriented Specification Case Studies*. Object Oriented Series. Prentice Hall International, 1993.
 Contents: Chapters introducing object oriented methods, object oriented formal specification and the links between formal and structured object-oriented techniques; seven case studies in particular object oriented formal methods, including:
 The Unix Filing System: A MooZ Specification; An Object-Z Specification of a Mobile Phone System; Object-oriented Specification in VDM^{++} ; Specifying a Concept-recognition System in Z^{++} ; Specification in OOZE; Refinement in Fresco; SmallVDM: An Environment for Formal Specification and Prototyping in Smalltalk.
 A glossary, index and bibliography are also included. The contributors are some of the leading figures in the area, including the developers of the above methods and languages: Silvio Meira, Gordon Rose, Roger Duke, Antonio Alencar, Joseph Goguen, Alan Wills, Cassio Souza dos Santos, Ana Cavalcanti.
307. K. C. Lano and H. P. Haughton. Reuse and adaptation of Z specifications. In Bowen and Nicholls [75], pages 62–90.
308. K. C. Lano and H. P. Haughton. *Reverse Engineering and Software Maintenance: A Practical Approach*. International Series in Software Engineering. McGraw Hill, 1993.
309. K. C. Lano and H. P. Haughton. Standards and techniques for object-oriented formal specification. In *Proc. 1993 Software Engineering Standards Symposium*, pages 237–246. IEEE Computer Society Press, 1993.
310. K. C. Lano and H. P. Haughton. Formal development in B Abstract Machine Notation. *Information and Software Technology*, 37(5):303–316, May 1995.
311. K. C. Lano, H. P. Haughton, and P. T. Breuer. Using object-oriented extensions of Z for maintenance and reverse-engineering. Technical Report PRG-TR-22-91, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, 1991.
312. G. Laycock. Formal specification and testing: A case study. *Software Testing, Verification and Reliability*, 2(1):7–23, May 1992.
313. D. Lightfoot. *Formal Specification using Z*. Macmillan, 1991.
 Contents: Introduction; Sets in Z; Using sets to describe a system – a simple example; Logic: propositional calculus; Example of a Z specification document; Logic: predicate calculus; Relations; Functions; A seat allocation system; Sequences; An example of sequences – the aircraft example again; Extending a specification; Collected notation; Books on formal specification; Hints on creating specifications; Solutions to exercises. Also available in French.
314. P. A. Lindsay. Reasoning about Z specifications: A VDM perspective. Technical Report 93-20, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, October 1993.
315. P. A. Lindsay. On transferring VDM verification techniques to Z. In Naftalin et al. [361], pages 190–213.

Also available as Technical Report 94-10, Department of Computer Science, University of Queensland, 1994.

316. P. A. Lindsay and E. van Keulen. Case studies in the verification of specifications in VDM and Z. Technical Report 94-3, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, March 1994.
317. R. L. London. Specifying reusable components using Z: Sets implemented by bit vectors. Technical Report CR-88-14, Tektronix Laboratories, P. O. Box 500, MS 50-662, Beaverton, Oregon 97077, USA, November 1988.
318. R. L. London and K. R. Milsted. Specifying reusable components using Z: Realistic sets and dictionaries. *ACM SIGSOFT Software Engineering Notes*, 14(3):120–127, May 1989.
319. M. Love. Animating Z specifications in SQL*Forms3.0. In Bowen and Nicholls [75], pages 294–306.
320. P. J. Lupton. Promoting forward simulation. In Nicholls [373], pages 27–49.
321. A. MacDonald and D. Carrington. Synthesising designs from formal specifications: A case study. Technical Report 93-21, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, November 1993.
322. R. Macdonald. Z usage and abuse. Report no. 91003, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, February 1991.
This paper presents a miscellany of observations drawn from experience of using Z, shows a variety of techniques for expressing certain class of idea concisely and clearly, and alerts the reader to certain pitfalls which may trap the unwary.
323. B. P. Mahony and I. J. Hayes. A case-study in timed refinement: A central heater. In Morris and Shaw [359], pages 138–149.
324. B. P. Mahony and I. J. Hayes. A case-study in timed refinement: A mine pump. *IEEE Transactions on Software Engineering*, 18(9):817–826, September 1992.
325. B. P. Mahony, C. Millerchip, and I. J. Hayes. A boiler control system: A case-study in timed refinement. Technical report, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, 23 June 1993.
A specification and top-level design of a steam generating boiler system is presented as an example of the formal development of a real-time system.
326. K. C. Mander and F. Polack. Rigorous specification using structured systems analysis and Z. *Information and Software Technology*, 37(5):285–291, May 1995.
Revised version of [393].
327. A. Martin. Encoding W: A logic for Z in 2OBJ. In Woodcock and Larsen [527], pages 462–481.
328. P. Martin. The formal specification in Z of task migration on the testbed multicomputer. Technical Report ECS-CSG-2-94, Department of Computer Science, University of Edinburgh, Edinburgh EH9 3JZ, UK., February 1994. To appear in *IEEE Software*.
329. P. Mataga and P. Zave. Formal specification of telephone features. In Bowen and Hall [67], pages 29–50.
330. P. Mataga and P. Zave. Multiparadigm specification of an AT&T switching system. In Hinchey and Bowen [245], pages 375–398.
331. P. Mataga and P. Zave. Using Z to specify telephone features. *Information and Software Technology*, 37(5):277–283, May 1995.
Revised version of [329].
332. I. Maung and J. R. Howse. Introducing Hyper-Z – a new approach to object orientation in Z. In Bowen and Nicholls [75], pages 149–165.
333. M. D. May. Use of formal methods by a silicon manufacturer. In C. A. R. Hoare, editor, *Developments in Concurrency and Communication*, University of Texas at Austin Year

- of Programming Series, chapter 4, pages 107–129. Addison-Wesley Publishing Company, 1990.
334. M. D. May, G. Barrett, and D. E. Shepherd. Designing chips that work. In C. A. R. Hoare and M. J. C. Gordon, editors, *Mechanized Reasoning and Hardware Design*, pages 3–19. Prentice Hall International Series in Computer Science, 1992.
 335. M. D. May and D. E. Shepherd. Verification of the IMS T800 microprocessor. In *Proc. Electronic Design Automation*, pages 605–615, London, UK, September 1987.
 336. J. A. McDermid. Special section on Z. *IEE/BCS Software Engineering Journal*, 4(1):25–72, January 1989.
A special issue on Z, introduced and edited by Prof. J. A. McDermid. See also [51, 110, 453, 519].
 337. J. A. McDermid, editor. *The Theory and Practice of Refinement: Approaches to the Formal Development of Large-Scale Software Systems*. Butterworth Scientific, 1989.
This book contains papers from the 1st Refinement Workshop held at the University of York, UK, 7–8 January 1988. Z-related papers include [366, 291].
 338. J. A. McDermid. Formal methods: Use and relevance for the development of safety critical systems. In P. A. Bennett, editor, *Safety Aspects of Computer Control*. Butterworth-Heinemann, Oxford, UK, 1993.
This paper discusses a number of formal methods and summarizes strengths and weaknesses in safety critical applications; a major safety-related example is presented in Z.
 339. M. A. McMorran and J. E. Nicholls. Z user manual. Technical Report TR12.274, IBM United Kingdom Laboratories Ltd, Hursley Park, Winchester, Hampshire SO21 2JN, UK, July 1989.
 340. M. A. McMorran and S. Powell. *Z Guide for Beginners*. Blackwell Scientific, 1993.
 341. S. L. Meira and A. L. C. Cavalcanti. Modular object-oriented Z specifications. In Nicholls [373], pages 173–192.
 342. S. L. Meira and A. L. C. Cavalcanti. The MooZ specification language. Technical report, Universidade Federal de Pernambuco, Departamento de Informática, Recife – PE, Brasil, 1992.
 343. B. Meyer. On formalism in specifications. *IEEE Software*, 2(1):6–26, January 1985.
 344. V. Mišić, D. Velašević, and B. Lazarević. Formal specification of a data dictionary for an extended ER data model. *The Computer Journal*, 35(6):611–622, December 1992.
 345. J. D. Moffett and M. S. Sloman. A case study representing a model: To Z or not to Z? In Nicholls [373], pages 254–268.
 346. B. Q. Monahan. Book review. *Formal Aspects of Computing*, 1(1):137–142, January–March 1989.
A review of *Understanding Z: A Specification Language and Its Formal Semantics* by Mike Spivey [452].
 347. B. Q. Monahan and R. C. Shaw. Model-based specifications. In J. A. McDermid, editor, *Software Engineer's Reference Book*, chapter 21. Butterworth-Heinemann, Oxford, UK, 1991.
This chapter contains a case study in Z, followed by a discussion of the respective trade-offs in specification between Z and VDM.
 348. C. C. Morgan. Data refinement using miracles. *Information Processing Letters*, 26(5):243–246, January 1988.
 349. C. C. Morgan. Procedures, parameters, and abstraction: Separate concerns. *Science of Computer Programming*, 11(1), October 1988.
 350. C. C. Morgan. The specification statement. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 10(3), July 1988.

351. C. C. Morgan. Types and invariants in the refinement calculus. In *Proc. Mathematics of Program Construction Conference*, Twente, June 1989.
352. C. C. Morgan. *Programming from Specifications*. Prentice Hall International Series in Computer Science, 2nd edition, 1994.
This book presents a rigorous treatment of most elementary program development techniques, including iteration, recursion, procedures, parameters, modules and data refinement.
353. C. C. Morgan and K. A. Robinson. Specification statements and refinement. *IBM Journal of Research and Development*, 31(5), September 1987.
354. C. C. Morgan and J. W. Sanders. Laws of the logical calculi. Technical Monograph PRG-78, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, September 1989.
This document records some important laws of classical predicate logic. It is designed as a reservoir to be tapped by *users* of logic, in system development.
355. C. C. Morgan and B. A. Sufrin. Specification of the Unix filing system. *IEEE Transactions on Software Engineering*, 10(2):128–142, March 1984.
356. C. C. Morgan and T. Vickers, editors. *On the Refinement Calculus*. Formal Approaches to Computing and Information Technology series (FACIT). Springer-Verlag, 1994.
This book collects together the work accomplished at Oxford on the refinement calculus: the rigorous development, from state-based assertional specification, of executable imperative code.
357. C. C. Morgan and J. C. P. Woodcock. What is a specification? In D. Craigen and K. Summerskill, editors, *Formal Methods for Trustworthy Computer Systems (FM89)*, Workshops in Computing, pages 38–43. Springer-Verlag, 1990.
358. C. C. Morgan and J. C. P. Woodcock, editors. *3rd Refinement Workshop*, Workshops in Computing. Springer-Verlag, 1991.
The workshop was held at the IBM Laboratories, Hursley Park, UK, 9–11 January 1990. See [434].
359. J. M. Morris and R. C. Shaw, editors. *4th Refinement Workshop*, Workshops in Computing. Springer-Verlag, 1991.
The workshop was held at Cambridge, UK, 9–11 January 1991. For Z related papers, see [23, 264, 323, 512, 522, 507].
360. P. Mundy and J. B. Wordsworth. The CICS application programming interface: Transient data and storage control. IBM Technical Report TR12.299, IBM United Kingdom Laboratories Ltd, Hursley Park, Winchester, Hampshire SO21 2JN, UK, October 1990.
One of a number of reports on the CICS application programming interface. See also [41, 247, 289].
361. M. Naftalin, T. Denvir, and M. Bertran, editors. *FME'94: Industrial Benefit of Formal Methods*, volume 873 of *Lecture Notes in Computer Science*. Formal Methods Europe, Springer-Verlag, 1994.
The 2nd FME Symposium was held at Barcelona, Spain, 24–28 October 1994. Z-related papers include [69, 120, 136, 173, 175, 180, 258, 315]. B-related papers include [142, 415, 472].
362. K. T. Narayana and S. Dharap. Formal specification of a look manager. *IEEE Transactions on Software Engineering*, 16(9):1089–1103, September 1990.
A formal specification of the look manager of a dialog system is presented in Z. This deals with the presentation of visual aspects of objects and the editing of those visual aspects.
363. K. T. Narayana and S. Dharap. Invariant properties in a dialog system. *ACM SIGSOFT Software Engineering Notes*, 15(4):67–79, September 1990.
364. T. C. Nash. Using Z to describe large systems. In Nicholls [371], pages 150–178.

365. Ph. W. Nehlig and D. A. Duce. GKS-9x: The design output primitive, an approach to specification. *Computer Graphics Forum*, 13(3):C-381-C-392, 1994.
366. D. S. Neilson. Hierarchical refinement of a Z specification. In McDermid [337].
367. D. S. Neilson. From Z to C: Illustration of a rigorous development method. Technical Monograph PRG-101, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, 1990.
368. D. S. Neilson. Machine support for Z: The zedB tool. In Nicholls [373], pages 105–128.
369. D. S. Neilson and D. Prasad. zedB: A proof tool for Z built on B. In Nicholls [375], pages 243–258.
370. J. E. Nicholls. Working with formal methods. *Journal of Information Technology*, 2(2):67–71, June 1987.
371. J. E. Nicholls, editor. *Z User Workshop, Oxford 1989*, Workshops in Computing. Springer-Verlag, 1990.
 Proceedings of the Fourth Annual Z User Meeting, Wolfson College & Rewley House, Oxford, UK, 14–15 December 1989. Published in collaboration with the British Computer Society. For the opening address see [384]. For individual papers, see [33, 86, 87, 119, 146, 183, 204, 229, 242, 268, 301, 364, 388, 443, 459, 505].
372. J. E. Nicholls. A survey of Z courses in the UK. In *Z User Workshop, Oxford 1990* [373], pages 343–350.
373. J. E. Nicholls, editor. *Z User Workshop, Oxford 1990*, Workshops in Computing. Springer-Verlag, 1991.
 Proceedings of the Fifth Annual Z User Meeting, Lady Margaret Hall, Oxford, UK, 17–18 December 1990. Published in collaboration with the British Computer Society. For individual papers, see [25, 80, 95, 116, 187, 205, 232, 243, 252, 271, 274, 299, 341, 345, 368, 372, 381, 403, 430, 506, 536]. The proceedings also includes an *Introduction and Opening Remarks*, a *Selected Z Bibliography*, a selection of posters and information on Z tools.
374. J. E. Nicholls. Domains of application for formal methods. In *Z User Workshop, York 1991* [375], pages 145–156.
375. J. E. Nicholls, editor. *Z User Workshop, York 1991*, Workshops in Computing. Springer-Verlag, 1992.
 Proceedings of the Sixth Annual Z User Meeting, York, UK. Published in collaboration with the British Computer Society. For individual papers, see [18, 28, 139, 98, 149, 158, 217, 369, 374, 394, 422, 444, 483, 498, 525, 541].
376. J. E. Nicholls. Plain guide to the Z base standard. In Bowen and Nicholls [75], pages 52–61.
377. J. E. Nicholls et al. Z in the development process. Technical Report PRG-TR-1-89, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, June 1989. Proceedings of a discussion workshop held on 15 December 1988 in Oxford, UK, with contributions by Peter Collins, David Cooper, Anthony Hall, Patrick Hall, Brian Hepworth, Ben Potter and Andrew Ricketts.
378. C. J. Nix and B. P. Collins. The use of software engineering, including the Z notation, in the development of CICS. *Quality Assurance*, 14(3):103–110, September 1988.
379. A. Norcliffe and G. Slater. *Mathematics for Software Construction*. Series in Mathematics and its Applications. Ellis Horwood, 1991.
 Contents: Why mathematics; Getting started: sets and logic; Developing ideas: schemas; Functions; Functions in action; A real problem from start to finish: a drinks machine; Sequences; Relations; Generating programs from specifications: refinement; The role of proof; More examples of specifications; Concluding remarks; Answers to exercises.

380. A. Norcliffe and S. Valentine. Z readers video course. PAVIC Publications, 1992. Sheffield Hallam University, 33 Collegiate Crescent, Sheffield S10 2BP, UK.
Video-based Training Course on the Z Specification Language. The course consists of 5 videos, each of approximately one hour duration, together with supporting texts and case studies.
381. A. Norcliffe and S. H. Valentine. A video-based training course in reading Z specifications. In Nicholls [373], pages 337–342.
382. G. Normington. Cleanroom and Z. In Bowen and Nicholls [75], pages 281–293.
383. C. O’Halloran. Evaluation semantics in Z. In Naftalin et al. [361], pages 502–518.
384. B. Oakley. The state of use of formal methods. In Nicholls [371], pages 1–5.
A record of the opening address at ZUM’89.
385. C. O’Halloran. The software repeater (an exercise in Z specification). Report no. 4090, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, 1987.
386. E. A. Oxborrow and H. M. Ismail. KBZ - an object-oriented approach to the specification and management of knowledge bases. Technical Report 51, Computing Laboratory, University of Kent at Canterbury, UK, 1988.
The report describes the important features of KBZ, an extension of Z which should better support the specification of the semantics of conceptual data models. The report then clarifies the use of KBZ by discussing its application in a distributed database environment.
387. C. E. Parker. Z tools catalogue. ZIP project report ZIP/BAe/90/020, British Aerospace, Software Technology Department, Warton PR4 1AX, UK, May 1991.
388. M. Phillips. CICS/ESA 3.1 experiences. In Nicholls [371], pages 179–185.
Z was used to specify 37,000 lines out of 268,000 lines of code in the IBM CICS/ESA 3.1 release. The initial development benefit from using Z was assessed as being a 9% improvement in the *total development cost* of the release, based on the reduction of programmer days fixing problems.
389. M. Pilling, A. Burns, and K. Raymond. Formal specifications and proofs of inheritance protocols for real-time scheduling. *IEE/BCS Software Engineering Journal*, 5(5):263–279, September 1990.
390. P. R. H. Place and K. C. Kang. Safety-critical software: Status report and annotated bibliography. Technical Report CMU/SEI-92-TR-5 & ESC-TR-93-182, Software Engineering Institute, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213, USA, June 1993.
391. P. R. H. Place and W. Wood. Survey of formal specification techniques for reactive systems. CMU Technical Report CMU/SEI-90-TR-5, ADA22374, Software Engineering Institute, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213, USA, 1990.
392. P. R. H. Place and W. Wood. Formal development of Ada programs using Z and Anna: A case study. CMU Technical Report CMU/SEI-91-TR-1, ADA235698, Software Engineering Institute, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213, USA, February 1991.
Copies available from: Research Access Inc., 3400 Forbes Avenue, Suite 302, Pittsburgh, PA 15213, USA.
393. F. Polack and K. C. Mander. Software quality assurance using the SAZ method. In Bowen and Hall [67], pages 230–249.
394. F. Polack, M. Whiston, and P. Hitchcock. Structured analysis – a draft method for writing Z specifications. In Nicholls [375], pages 261–286.
395. F. Polack, M. Whiston, and K. C. Mander. The SAZ project: Integrating SSADM and Z. In Woodcock and Larsen [527], pages 541–557.
396. B. F. Potter, J. E. Sinclair, and D. Till. *An Introduction to Formal Specification and Z*. Prentice Hall International Series in Computer Science, 1991.

Contents: Formal specification in the context of software engineering; An informal introduction to logic and set theory; A first specification; The Z notation: the mathematical language, relations and functions, schemas and specification structure; A first specification revisited; Formal reasoning; From specification to program: data and operation refinement, operation decomposition; From theory to practice.

397. B. F. Potter and D. Till. The specification in Z of gateway functions within a communications network. In *Proc. IFIP WG10.3 Conference on Distributed Processing*. Elsevier Science Publishers (North-Holland), October 1987.
398. S. Prehn and W. J. Toetenel, editors. *VDM'91: Formal Software Development Methods*, volume 551 of *Lecture Notes in Computer Science*. Springer-Verlag, 1991. Volume 1: Conference Contributions.
The 4th VDM-Europe Symposium was held at Noordwijkerhout, The Netherlands, 21–25 October 1991. Papers with relevance to Z include [17, 35, 128, 154, 193, 249, 501, 509, 540]. See also [399].
399. S. Prehn and W. J. Toetenel, editors. *VDM'91: Formal Software Development Methods*, volume 552 of *Lecture Notes in Computer Science*. Springer-Verlag, 1991. Volume 2: Tutorials.
Papers with relevance to Z include [6, 523]. See also [398].
400. G-H. B. Rafsanjani and S. J. Colwill. From Object-Z to C^{++} : A structural mapping. In Bowen and Nicholls [75], pages 166–179.
401. RAISE Language Group. *The RAISE Specification Language*. BCS Practitioner Series. Prentice Hall International, 1992.
402. G. P. Randell. Translating data flow diagrams into Z (and vice versa). Report no. 90019, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, October 1990.
403. G. P. Randell. Data flow diagrams and Z. In Nicholls [373], pages 216–227.
404. G. P. Randell. Improving the translation from data flow diagrams into Z by incorporating the data dictionary. Report no. 92004, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, January 1992.
405. D. Rann, J. Turner, and J. Whitworth. *Z: A Beginner's Guide*. Chapman & Hall, London, 1994.
406. B. Ratcliff. *Introducing Specification Using Z: A Practical Case Study Approach*. International Series in Software Engineering. McGraw-Hill, 1994.
407. A. P. Ravn, H. Rischel, and V. Stavridou. Provably correct safety critical software. In *Proc. IFAC Safety of Computer Controlled Systems 1990 (SAFECOMP'90)*. Pergamon Press, 1990.
Also available as Technical Report CSD-TR-625 from Department of Computer Science, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK.
408. M. Rawson. OOPSLA'93: Workshop on formal specification of object-oriented systems – position paper. In H. Kilov and W. Harvey, editors, *Proc. Workshop on Specification of Behavioral Semantics in Object-Oriented Information Modeling*, pages 125–135, Institute for Information Management and Department of Computer and Information Systems, Robert Morris College, Coraopolos and Pittsburgh, Pennsylvania, USA, 1993. OOPSLA.
409. K. Raymond, P. Stocks, and D. Carrington. Using Z to specify distributed systems. Technical Report 181, Key Centre for Software Technology, University of Queensland, St. Lucia 4072, Australia, 1990.
410. T. J. Read. Formal specification of reusable Ada software packages. In A. Burns, editor, *Towards Ada 9X Conference Proceedings*, pages 98–117, 1991.
411. J. N. Reed. Semantics-based tools for a specification support environment. In *Mathematical Foundations of Programming Language Semantics*, volume 298 of *Lecture Notes in Computer Science*. Springer-Verlag, 1988.

412. J. N. Reed and J. E. Sinclair. An algorithm for type-checking Z: A Z specification. Technical Monograph PRG-81, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, March 1990.
413. N. R. Reizer, G. D. Abowd, B. C. Meyers, and P. R. H. Place. Using formal methods for requirements specification of a proposed POSIX standard. In *IEEE International Conference on Requirements Engineering (ICRE'94)*, April 1994.
414. G. J. Reynolds. Yet another approach to the formal specification of a configurable graphics system. In *Proc. Eurographics Association Formal Methods in Computer Graphics*, June 1991.
415. B. Ritchie, J. Bicarregui, and H. P. Haughton. Experiences in using the abstract machine notation in a GKS case study. In Naftalin et al. [361], pages 93–104.
416. K. A. Robinson. Refining Z specifications to programs. In *Proc. Australian Software Engineering Conference*, pages 87–97, 1987.
417. G. A. Rose. Object-Z. In Stepney et al. [462], pages 59–77.
418. G. A. Rose and P. Robinson. A case study in formal specifications. In *Proc. First Australian Software Engineering Conference*, May 1986.
419. K. J. Ross and P. A. Lindsay. Maintaining consistency under changes to formal specifications. Technical Report 93-3, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, March 1993.
420. A. R. Ruddle. Formal methods in the specification of real-time, safety-critical control systems. In Bowen and Nicholls [75], pages 131–146.
421. P. Rudkin. Modelling information objects in Z. In J. de Meer, editor, *Proc. International Workshop on ODP*. Elsevier Science Publishers (North-Holland), 1992.
422. M. Saaltink. Z and Eves. In Nicholls [375], pages 223–242.
423. H. Saiedian. The mathematics of computing. *Journal of Computer Science Education*, 3(3):203–221, 1992.
424. A. C. A. Sampaio and S. L. Meira. Modular extensions to Z. In Bjørner et al. [39], pages 211–232.
425. P. Sanders, M. Johnson, and R. Tinker. From Z specifications to functional implementations. *British Telecom Technology Journal*, 7(4), October 1989.
426. S. A. Schuman and D. H. Pitt. Object-oriented subsystem specification. In L. G. L. T. Meertens, editor, *Program Specification and Transformation*, pages 313–341. Elsevier Science Publishers (North-Holland), 1987.
427. S. A. Schuman, D. H. Pitt, and P. J. Byers. Object-oriented process specification. In C. Rattray, editor, *Specification and Verification of Concurrent Systems*, Workshops in Computing, pages 21–70. Springer-Verlag, 1990.
428. L. T. Semmens and P. M. Allen. Using entity relationship models as a basis for Z specifications. Technical Report IES1/90, Leeds Polytechnic, Faculty of Information and Engineering Systems, Leeds, UK, 1990.
429. L. T. Semmens and P. M. Allen. Using Yourdon and Z to specify computer security: A case study. Technical Report IES4/90, Leeds Polytechnic, Faculty of Information and Engineering Systems, Leeds, UK, 1990.
430. L. T. Semmens and P. M. Allen. Using Yourdon and Z: An approach to formal specification. In Nicholls [373], pages 228–253.
431. L. T. Semmens, R. B. France, and T. W. G. Docker. Integrated structured analysis and formal specification techniques. *The Computer Journal*, 35(6):600–610, December 1992.
432. C. T. Sennett. Review of type checking and scope rules of the specification language Z. Report no. 87017, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, November 1987.

433. C. T. Sennett. Formal specification and implementation. In C. T. Sennett, editor, *High-Integrity Software*, Computer Systems Series. Pitman, 1989.
434. C. T. Sennett. Using refinement to convince: Lessons learned from a case study. In Morgan and Woodcock [358], pages 172–197.
435. C. T. Sennett. Demonstrating the compliance of Ada programs with Z specifications. In Jones et al. [272].
436. C. T. Sennett and R. Macdonald. Separability and security models. Report no. 87020, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, November 1987.
437. D. E. Shepherd. Verified microcode design. *Microprocessors and Microsystems*, 14(10):623–630, December 1990.
This article is part of a special issue on *Formal aspects of microprocessor design*, edited by H. S. M. Zedan. See also [52].
438. D. E. Shepherd and G. Wilson. Making chips that work. *New Scientist*, 1664:61–64, May 1989.
A general article containing information on the formal development of the T800 floating-point unit for the transputer including the use of Z.
439. D. Sheppard. *An Introduction to Formal Specification with Z and VDM*. International Series in Software Engineering. McGraw Hill, 1995.
440. L. B. Sherrell and D. L. Carver. Z meets Haskell: A case study. In *COMPSAC '93: 17th Annual International Computer Software and Applications Conference*, pages 320–326. IEEE Computer Society Press, November 1993.
The paper traces the development of a simple system, the class manager's assistant, from an existing Z specification, through design in Z, to a Haskell implementation.
441. L. N. Simcox. The application of Z to the specification of air traffic control systems: 1. Memorandum no. 4280, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, April 1989.
442. R. Sinnott and K. J. Turner. Modeling ODP viewpoints. In H. Kilov, W. Harvey, and H. Mili, editors, *Proc. Workshop on Precise Behavioral Specifications in Object-Oriented Information Modeling, OOPSLA 1994*, pages 121–128, Robert Morris College, Coraopolis and Pittsburgh, Pennsylvania 15108-1189, USA, 1994. OOPSLA.
443. A. Smith. The Knuth-Bendix completion algorithm and its specification in Z. In Nicholls [371], pages 195–220.
444. A. Smith. On recursive free types in Z. In Nicholls [375], pages 3–39.
445. G. Smith. *An Object-Oriented Approach to Formal Specification*. PhD thesis, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, October 1992.
A detailed description of a version of Object-Z similar to (but not identical to) that in [167]. The thesis also includes a formalization of temporal logic history invariants and a fully-abstract model of classes in Object-Z.
446. G. Smith. A object-oriented development framework for Z. In Bowen and Hall [67], pages 89–107.
447. G. Smith and R. Duke. Specification and verification of a cache coherence protocol. Technical Report 126, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, 1989.
448. G. Smith and R. Duke. Modelling a cache coherence protocol using Object-Z. In *Proc. 13th Australian Computer Science Conference (ACSC-13)*, pages 352–361, 1990.
449. P. Smith and R. Keighley. The formal development of a secure transaction mechanism. In Prehn and Toetenel [398], pages 457–476.
450. I. Sommerville. *Software Engineering*, chapter 9, pages 153–168. Addison-Wesley, 4th edition, 1992.
A chapter entitled *Model-Based Specification* including examples using Z.

451. I. H. Sørensen. A specification language. In J. Staunstrup, editor, *Program Specification: Proceedings of a Workshop*, volume 134 of *Lecture Notes in Computer Science*, pages 381–401. Springer-Verlag, 1981.
452. J. M. Spivey. *Understanding Z: A Specification Language and its Formal Semantics*, volume 3 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, January 1988.
Published version of 1985 DPhil thesis.
453. J. M. Spivey. An introduction to Z and formal specifications. *IEE/BCS Software Engineering Journal*, 4(1):40–50, January 1989.
454. J. M. Spivey. A guide to the zed style option. Oxford University Computing Laboratory, December 1990.
A description of the Z style option ‘zed.sty’ for use with the L^AT_EX document preparation system [297].
455. J. M. Spivey. Specifying a real-time kernel. *IEEE Software*, 7(5):21–28, September 1990.
This case study of an embedded real-time kernel shows that mathematical techniques have an important role to play in documenting systems and avoiding design flaws.
456. J. M. Spivey. *The fUZZ Manual*. Computing Science Consultancy, 34 Westlands Grove, Stockton Lane, York YO3 0EF, UK, 2nd edition, July 1992.
The manual describes a Z type-checker and ‘fuzz.sty’ style option for L^AT_EX documents [297]. The package is compatible with the book, *The Z Notation: A Reference Manual* by the same author [457].
457. J. M. Spivey. *The Z Notation: A Reference Manual*. Prentice Hall International Series in Computer Science, 2nd edition, 1992.
This is a revised edition of the first widely available reference manual on Z originally published in 1989. The book provides a complete and definitive guide to the use of Z in specifying information systems, writing specifications and designing implementations. See also the draft Z standard [83].
Contents: Tutorial introduction; Background; The Z language; The mathematical tool-kit; Sequential systems; Syntax summary; Changes from the first edition; Glossary.
458. J. M. Spivey and B. A. Suftrin. Type inference in Z. In Bjørner et al. [39], pages 426–438.
459. J. M. Spivey and B. A. Suftrin. Type inference in Z. In Nicholls [371], pages 6–31.
Also published as [458].
460. S. Stepney. *High Integrity Compilation: A Case Study*. Prentice Hall, 1993.
461. S. Stepney and R. Barden. Annotated Z bibliography. *Bulletin of the European Association of Theoretical Computer Science*, 50:280–313, June 1993.
462. S. Stepney, R. Barden, and D. Cooper, editors. *Object Orientation in Z*. Workshops in Computing. Springer-Verlag, 1992.
This is a collection of papers describing various OOZ approaches – Hall, ZERO, MooZ, Object-Z, OOZE, Schuman & Pitt, Z⁺⁺, ZEST and Fresco (an object-oriented VDM method) – in the main written by the methods’ inventors, and all specifying the same two examples. The collection is a revised and expanded version of a ZIP report distributed at the 1991 Z User Meeting at York.
463. S. Stepney, R. Barden, and D. Cooper. A survey of object orientation in Z. *IEE/BCS Software Engineering Journal*, 7(2):150–160, March 1992.
464. S. Stepney and S. P. Lord. Formal specification of an access control system. *Software – Practice and Experience*, 17(9):575–593, September 1987.
465. P. Stocks. *Applying formal methods to software testing*. PhD thesis, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, 1993.
466. P. Stocks and D. A. Carrington. Deriving software test cases from formal specifications. In *6th Australian Software Engineering Conference*, pages 327–340, July 1991.

467. P. Stocks and D. A. Carrington. Test template framework: A specification-based testing case study. In *Proc. International Symposium on Software Testing and Analysis (ISSTA'93)*, pages 11–18, June 1993.
Also available in a longer form as Technical Report UQCS-255, Department of Computer Science, University of Queensland.
468. P. Stocks and D. A. Carrington. Test templates: A specification-based testing framework. In *Proc. 15th International Conference on Software Engineering*, pages 405–414, May 1993.
Also available in a longer form as Technical Report UQCS-243, Department of Computer Science, University of Queensland.
469. P. Stocks, K. Raymond, and D. Carrington. Representing distributed system concepts in Z. Technical Report 180, Key Centre for Software Technology, University of Queensland, St. Lucia 4072, Australia, 1990.
470. P. Stocks, K. Raymond, D. Carrington, and A. Lister. Modelling open distributed systems in Z. *Computer Communications*, 15(2):103–113, March 1992.
In a special issue on the practical use of FDTs (Formal Description Techniques) in communications and distributed systems, edited by Dr. Gordon S. Blair.
471. B. Stoddart and P. Knaggs. The event calculus (formal specification of real time systems by means of diagrams and Z schemas). In *5th International Conference on putting into practice methods and tools for information system design*, University of Nantes, Institute Universitaire de Technologie, 3 Rue du Maréchal Joffre, 44041 Nantes Cedex 01, France, September 1992.
472. A. C. Storey and H. P. Houghton. A strategy for the production of verifiable code using the B method. In Naftalin et al. [361], pages 346–365.
473. B. A. Sufrin. Formal system specification: Notation and examples. In D. Neel, editor, *Tools and Notations for Program Construction*. Cambridge University Press, 1982.
An example of a filing system specification, this was the first published use of the schema notation to put together states.
474. B. A. Sufrin. Towards formal specification of the ICL data dictionary. *ICL Technical Journal*, August 1984.
475. B. A. Sufrin. Formal methods and the design of effective user interfaces. In M. D. Harrison and A. F. Monk, editors, *People and Computers: Designing for Usability*. Cambridge University Press, 1986.
476. B. A. Sufrin. Formal specification of a display-oriented editor. In N. Gehani and A. D. McGettrick, editors, *Software Specification Techniques*, International Computer Science Series, pages 223–267. Addison-Wesley Publishing Company, 1986.
Originally published in *Science of Computer Programming*, 1:157–202, 1982.
477. B. A. Sufrin. A formal framework for classifying interactive information systems. In *IEE Colloquium on Formal Methods and Human-Computer Interaction*, number 09 in IEE Digest, pages 4/1–14, London, UK, 1987. The Institution of Electrical Engineers.
478. B. A. Sufrin. Effective industrial application of formal methods. In G. X. Ritter, editor, *Information Processing 89, Proc. 11th IFIP Computer Congress*, pages 61–69. Elsevier Science Publishers (North-Holland), 1989.
This paper presents a Z model of the Unix *make* utility.
479. B. A. Sufrin and He Jifeng. Specification, analysis and refinement of interactive processes. In M. D. Harrison and H. Thimbleby, editors, *Formal Methods in Human-Computer Interaction*, volume 2 of *Cambridge Series on Human-Computer Interaction*, chapter 6, pages 153–200. Cambridge University Press, 1990.
A case study on using Z for process modelling.

480. B. A. Sufrin and J. C. P. Woodcock. Towards the formal specification of a simple programming support environment. *IEE/BCS Software Engineering Journal*, 2(4):86–94, July 1987.
481. P. A. Swatman. *Increasing Formality in the Specification of High-Quality Information Systems in a Commercial Context*. Phd thesis, Curtin University of Technology, School of Computing, Perth, Western Australia, July 1992.
482. P. A. Swatman. Using formal specification in the acquisition of information systems: Educating information systems professionals. In Bowen and Nicholls [75], pages 205–239.
483. P. A. Swatman, D. Fowler, and C. Y. M. Gan. Extending the useful application domain for formal methods. In Nicholls [375], pages 125–144.
484. P. A. Swatman and P. M. C. Swatman. Formal specification: An analytic tool for (management) information systems. *Journal of Information Systems*, 2(2):121–160, April 1992.
485. P. A. Swatman and P. M. C. Swatman. Is the information systems community wrong to ignore formal specification methods? In R. Clarke and J. Cameron, editors, *Managing Information Technology's Organisational Impact*. Elsevier Science Publishers (North-Holland), October 1992.
486. P. A. Swatman and P. M. C. Swatman. Managing the formal specification of information systems. In *Proc. International Conference on Organization and Information Systems*, September 1992.
487. P. A. Swatman, P. M. C. Swatman, and R. Duke. Electronic data interchange: A high-level formal specification in Object-Z. In *Proc. 6th Australian Software Engineering Conference (ASWEC'91)*, 1991.
488. P. F. Terry and S. R. Wiseman. On the design and implementation of a secure computer system. Memorandum no. 4188, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, June 1988.
489. S. Thompson. Specification techniques [9004-0316]. *ACM Computing Reviews*, 31(4):213, April 1990.
A review of *Formal methods applied to a floating-point number system* by Geoff Barrett [30].
490. D. Till, editor. *6th Refinement Workshop*, Workshop in Computing. Springer-Verlag, 1994. The workshop was held at City University, London, UK, 5–7 January 1994. See [181, 300].
491. B. S. Todd. A model-based diagnostic program. *IEE/BCS Software Engineering Journal*, 2(3):54–63, May 1987.
492. R. Took. The presenter – a formal design for an autonomous display manager. In I. Sommerville, editor, *Software Engineering Environments*, pages 151–169. Peter Peregrinus, London, 1986.
493. I. Toyn. *CADiZ Quick Reference Guide*. York Software Engineering Ltd, University of York, York YO1 5DD, UK, 1990.
A guide to the CADiZ (Computer Aided Design in Z) toolkit. This makes use of the Unix *troff* family of text formatting tools. Contact David Jordan at the address above or on `yse@minster.york.ac.uk` via e-mail for further information on CADiZ. See also [274] for a paper introducing CADiZ. Support for L^AT_EX [297] is now available.
494. I. Toyn and A. J. Dix. Efficient binary transfer of pointer structures. Technical document, Computer Science Department, University of York, York YO1 5DD, UK, November 1993.
495. I. Toyn and J. A. McDermid. CADiZ: An architecture for Z tools and its implementation. Technical document, Computer Science Department, University of York, York YO1 5DD, UK, November 1993.
496. O. Traynor, P. Kearney, E. Kazmierczak, Li Wang, and E. Karlsen. Extending Z with modules. *Australian Computer Science Communications*, 17(1), 1995. Proc. ACSC'95.

497. S. Valentine. The programming language Z^{-} . *Information and Software Technology*, 37(5):293–301, May 1995.
498. S. H. Valentine. Z^{-} , an executable subset of Z . In Nicholls [375], pages 157–187.
499. S. H. Valentine. Putting numbers into the mathematical toolkit. In Bowen and Nicholls [75], pages 9–36.
500. M. J. van Diepen and K. M. van Hee. A formal semantics for Z and the link between Z and the relational algebra. In Bjørner et al. [39], pages 526–551.
501. K. M. van Hee, L. J. Somers, and M. Voorhoeve. Z and high level Petri nets. In Prehn and Toetenel [398], pages 204–219.
502. H. J. van Zuylen, editor. *The REDO Compendium: Reverse Engineering for Software Maintenance*. John Wiley & Sons, 1993.
An overview of the results of the ESPRIT REDO project, including the use of Z and Z^{++} . See in particular Chapter 16, also published in a longer form as [302].
503. M. M. West and B. M. Eaglestone. Software development: Two approaches to animation of Z specifications using Prolog. *IEE/BCS Software Engineering Journal*, 7(4):264–276, July 1992.
504. C. Wezeman and A. Judge. Z for managed objects. In Bowen and Hall [67], pages 108–119.
505. R. W. Whitty. Structural metrics for Z specifications. In Nicholls [371], pages 186–191.
506. P. J. Whysall and J. A. McDermid. An approach to object-oriented specification using Z . In Nicholls [373], pages 193–215.
507. P. J. Whysall and J. A. McDermid. Object-oriented specification and refinement. In Morris and Shaw [359], pages 151–184.
508. J. M. Wing. A specifier’s introduction to formal methods. *IEEE Computer*, 23(9):8–24, September 1990.
509. J. M. Wing and A. M. Zaremski. Unintrusive ways to integrate formal specifications in practice. In Prehn and Toetenel [398], pages 545–570.
510. S. R. Wiseman and C. L. Harrold. A security model and its implementation. Memorandum no. 4222, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, September 1988.
511. A. W. Wood. A Z specification of the MaCHO interface editor. Memorandum no. 4247, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, November 1988.
512. K. R. Wood. The elusive software refinery: a case study in program development. In Morris and Shaw [359], pages 281–325.
513. K. R. Wood. A practical approach to software engineering using Z and the refinement calculus. *ACM Software Engineering Notes*, 18(5):79–88, December 1993.
514. W. G. Wood. Application of formal methods to system and software specification. *ACM SIGSOFT Software Engineering Notes*, 15(4):144–146, September 1990.
515. J. C. P. Woodcock. Teaching how to use mathematics for large-scale software development. *Bulletin of BCS-FACS*, July 1988.
516. J. C. P. Woodcock. Calculating properties of Z specifications. *ACM SIGSOFT Software Engineering Notes*, 14(4):43–54, 1989.
517. J. C. P. Woodcock. Mathematics as a management tool: Proof rules for promotion. In *Proc. 6th Annual CSR Conference on Large Software Systems*, Bristol, UK, September 1989.
518. J. C. P. Woodcock. Parallel refinement in Z . In *Proc. Workshop on Refinement*, January 1989.
519. J. C. P. Woodcock. Structuring specifications in Z . *IEE/BCS Software Engineering Journal*, 4(1):51–66, January 1989.
520. J. C. P. Woodcock. Transaction refinement in Z . In *Proc. Workshop on Refinement*, January 1989.

521. J. C. P. Woodcock. Z. In D. Craigen and K. Summerskill, editors, *Formal Methods for Trustworthy Computer Systems (FM89)*, Workshops in Computing, pages 57–62. Springer-Verlag, 1990.
522. J. C. P. Woodcock. Implementing promoted operations in Z. In Morris and Shaw [359], pages 366–378.
523. J. C. P. Woodcock. A tutorial on the refinement calculus. In Prehn and Toetenel [399], pages 79–140.
524. J. C. P. Woodcock. The rudiments of algorithm design. *The Computer Journal*, 35(5):441–450, October 1992.
525. J. C. P. Woodcock and S. M. Brien. W: A logic for Z. In Nicholls [375], pages 77–96.
526. J. C. P. Woodcock, P. H. B. Gardiner, and J. R. Hulance. The formal specification in Z of Defence Standard 00-56. In Bowen and Hall [67], pages 9–28.
527. J. C. P. Woodcock and P. G. Larsen, editors. *FME'93: Industrial-Strength Formal Methods*, volume 670 of *Lecture Notes in Computer Science*. Formal Methods Europe, Springer-Verlag, 1993.
The 1st FME Symposium was held at Odense, Denmark, 19–23 April 1993. Z-related papers include [76, 125, 179, 262, 327, 395].
528. J. C. P. Woodcock and P. G. Larsen. Guest editorial. *IEEE Transactions on Software Engineering*, 21(2):61–62, 1995.
Best papers of FME'93 [527]. See [43, 37, 127, 263].
529. J. C. P. Woodcock and M. Loomes. *Software Engineering Mathematics: Formal Methods Demystified*. Pitman, 1988.
Also published as: *Software Engineering Mathematics*, Addison-Wesley, 1989.
530. J. C. P. Woodcock and C. C. Morgan. Refinement of state-based concurrent systems. In Bjørner et al. [39], pages 340–351.
Work on combining Z and CSP.
531. R. Worden. Fermenting and distilling. In Bowen and Hall [67], pages 1–6.
532. J. B. Wordsworth. Teaching formal specification methods in an industrial environment. In *Proc. Software Engineering '86*, London, 1986. IEE/BCS, Peter Peregrinus.
533. J. B. Wordsworth. Specifying and refining programs with Z. In *Proc. Second IEE/BCS Conference on Software Engineering*, number 290 in Conference Publication, pages 8–16. IEE/BCS, July 1988.
A tutorial summary.
534. J. B. Wordsworth. Refinement tutorial: A storage manager. In *Proc. Workshop on Refinement*, January 1989.
535. J. B. Wordsworth. A Z development method. In *Proc. Workshop on Refinement*, January 1989.
536. J. B. Wordsworth. The CICS application programming interface definition. In Nicholls [373], pages 285–294.
537. J. B. Wordsworth. *Software Development with Z: A Practical Approach to Formal Methods in Software Engineering*. Addison-Wesley, 1993.
This book provides a guide to developing software from specification to code, and is based in part on work done at IBM's UK Laboratory that won the UK Queen's Award for Technological Achievement in 1992.
Contents: Introduction; A simple Z specification; Sets and predicates; Relations and functions; Schemas and specifications; Data design; Algorithm design; Specification of an oil terminal control system.
538. Xiaoping Jia. *ZTC: A Type Checker for Z—User's Guide*. Institute for Software Engineering, Department of Computer Science and Information Systems, DePaul University, Chicago, IL 60604, USA, 1994.

ZTC is a type checker for the Z specification language. ZTC accepts two forms of input: \LaTeX with `zed` style option and ZSL, an ASCII version of Z. ZTC can also perform translations between the two input forms. This document is intended to serve as both a user's guide and a reference manual for ZTC.

539. W. D. Young. Comparing specifications paradigms: Gypsy and Z. Technical Report 45, Computational Logic Inc., 1717 W. 6th St., Suite 290, Austin, Texas 78703, USA, 1989.
540. P. Zave and M. Jackson. Techniques for partial specification and specification of switching systems. In Prehn and Toetenel [398], pages 511–525.
Also published as [541].
541. P. Zave and M. Jackson. Techniques for partial specification and specification of switching systems. In Nicholls [375], pages 205–219.
542. Y. Zhang and P. Hitchcock. EMS: Case study in methodology for designing knowledge-based systems and information systems. *Information and Software Technology*, 33(7):518–526, September 1991.
543. Z archive. Oxford University Computing Laboratory, 1993.
A computer-based archive server at the Programming Research Group in Oxford is available for use by anyone with World-Wide Web (WWW) access, anonymous FTP access or an electronic mail address. This allows people interested in Z (and other things) to access various archived files. In particular, messages from the Z FORUM electronic mailing list [544] and a Z bibliography [53] are available.
The preferred method of access to the on-line Z archive is via the World-Wide Web (WWW) under the following 'URL' (Uniform Resource Locator):

`http://www.comlab.ox.ac.uk/archive/z.html`

Simply follow the hyperlinks of interest.

Much of the Z archive is also available via anonymous FTP on the Internet. Type the command `'ftp ftp.comlab.ox.ac.uk'` (or alternatively if this does not work, `'ftp 163.1.27.2'`) and use 'anonymous' as the login id and your e-mail address as the password when prompted. The FTP command `'cd pub/zforum'` will get you into the Z archive directory. The file `README` gives some general information and `00index` gives a list of the available files. The Z bibliography may be retrieved using the FTP command `'get z.bib'`, for example. If you wish to access any of the compressed POSTSCRIPT files in the archive, please issue the `'binary'` command first.

For users without on-line access on the Internet, it is possible to access parts of the Z archive using electronic mail, send a message to `archive-server@comlab.ox.ac.uk` with the 'Subject:' line and/or the body of the message containing commands such as the following:

<code>help</code>	help on using the PRG archive server
<code>index</code>	general index of categories (e.g., 'z')
<code>index z</code>	index of Z-related files
<code>send z z.bib</code>	send Z bibliography in \BibTeX format
<code>send z file1 file2 ...</code>	send multiple files
<code>path name@site</code>	optionally specify return e-mail address

If you have serious problems accessing the Z archive using WWW, anonymous FTP access or the electronic mail server and thus need human help, or if you wish to submit an item for the archive, please send electronic mail to `archive-management@comlab.ox.ac.uk`.

544. Z FORUM. Oxford University Computing Laboratory, 1986 onwards. Electronic mailing list: vol. 1.1–9 (1986), vol. 2.1–4 (1987), vol. 3.1–7 (1988), vol. 4.1–4 (1989), vol. 5.1–3 (1990).

Z FORUM is an electronic mailing list. It was initiated as an edited newsletter by R. Macdonald of DRA (formerly RSRE), Malvern, Worcestershire, UK, and is now maintained by J. P. Bowen at the Oxford University Computing Laboratory. Contributions should be sent to zforum@comlab.ox.ac.uk. Requests to join or leave the list should be sent to zforum-request@comlab.ox.ac.uk. Messages are now forwarded to the list directly to ensure timeliness. The list is also gatewayed to the USENET newsgroup `comp.specification.z` at Oxford and messages posted on either will automatically appear on both. A message answering some frequently asked questions is maintained and sent to the list once a month. A list of back issues of newsletters and other Z-related material is available electronically via anonymous FTP from [ftp.comlab.ox.ac.uk:/pub/Zforum](ftp://comlab.ox.ac.uk/pub/Zforum) in the file `00index` or via e-mail from the OUCL archive server [543]. For messages from a particular month (e.g., May 1995), access a file such as `zforum95-05`; for the most recent messages, see the file `zforum`.

This article was processed using the L^AT_EX macro package with LLNCS style