

Modelling notation and analysis

- [1] Robert Colvin and Ian J. Hayes. CSP with hierarchical state. In Michael Leuschel and Heike Wehrheim, editors, *Integrated Formal Methods (IFM 2009)*, volume 5423 of *Lecture Notes in Comp. Sci.*, pages 118–135. Springer, 2009.

Complex systems

- [2] Robert Colvin, Lars Grunske, and Kirsten Winter. Timed behavior trees for failure mode and effects analysis of time-critical systems. *Journal of Systems and Software*, 81(12):2163–2182, December 2008.
- [3] Robert Colvin Kirsten Winter and R. Geoff Dromey. Dynamic relational behaviour for large-scale systems. In Colin Fidge, editor, *Australian Software Engineering Conference (ASWEC 2009)*, pages 173–182. IEEE Computer Society, 2009.
- [4] Saad Zafar, Robert Colvin, Kirsten Winter, Nisansala Yatapanage, and R. Geoff Dromey. Early validation and verification of a distributed role-based access control model. In *14th Asia-Pacific Software Engineering Conference (APSEC 2007), 5-7 December 2007, Nagoya, Japan*, pages 430–437. IEEE Computer Society, 2007.
- [5] Lars Grunske, Robert Colvin, and Kirsten Winter. Probabilistic FMEA. In Mor Harchol-Balter, Marta Kwiatkowska, and Miklos Telek, editors, *Proceedings of Quantitative Evaluation of Systems (QEST)*, pages 119–128. IEEE, 2007.
- [6] Lian Wen, Robert Colvin, Kai Lin, John Seagrott, Nisansala Yatapanage, and R. Geoff Dromey. "Integrare", a Collaborative Environment for Behavior-Oriented Design. In Yuhua Luo, editor, *Cooperative Design, Visualization, and Engineering, 4th International Conference, CDVE 2007, Shanghai, China, September 16-20, 2007, Proceedings*, volume 4674 of *Lecture Notes in Computer Science*, pages 122–131. Springer, 2007.
- [7] Robert Colvin, Lars Grunske, and Kirsten Winter. Probabilistic timed Behavior Trees. In Jim Davies and Jeremy Gibbons, editors, *Proceedings of the International Conference on Integrated Formal Methods (IFM)*, volume 4591 of *Lecture Notes in Computer Science*, pages 156–175. Springer-Verlag, 2007.
- [8] Lars Grunske, Kirsten Winter, and Robert Colvin. Timed Behavior Trees and their Application to Verifying Real-time Systems. In John Grundy and Jun Han, editors, *Proceedings of the 18th Australian Conference on Software Engineering (ASWEC 2007)*, pages 211–220. IEEE Computer Society, 2007.

Verification of nonblocking algorithms

- [9] Robert Colvin and Brijesh Dongol. A general technique for proving lock-freedom. *Science of Computer Programming*, 74(3):143–165, 2008.
- [10] Lindsay Groves and Robert Colvin. Trace-based derivation of a scalable lock-free stack algorithm. *Formal Aspects of Computing*, 21(1–2):187–223, Feb 2009.

- [11] Robert Colvin and Lindsay Groves. A scalable lock-free stack algorithm and its verification. In *Fifth IEEE International Conference on Software Engineering and Formal Methods (SEFM 2007)*, 10-14 September 2007, London, England, UK, pages 339–348. IEEE Computer Society, 2007.
- [12] Robert Colvin and Brijesh Dongol. Verifying lock-freedom using well-founded orders. In Cliff B. Jones, Zhiming Liu, and Jim Woodcock, editors, *Theoretical Aspects of Computing - ICTAC 2007, 4th International Colloquium, Macau, China, September 26-28, 2007, Proceedings*, volume 4711 of *Lecture Notes in Computer Science*, pages 124–138. Springer, 2007.
- [13] Robert Colvin and Lindsay Groves. Derivation of a scalable lock-free stack algorithm. *Proceedings of the REFINE 2006 Workshop (REFINE 2006)*, *Electronic Notes in Theoretical Computer Science*, 187:55–74, 2007.
- [14] Robert Colvin, Lindsay Groves, Victor Luchangco, and Mark Moir. Formal verification of a lazy concurrent list-based set algorithm. In Thomas Ball and Robert B. Jones, editors, *Computer Aided Verification, 18th International Conference, CAV 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, volume 4144 of *Lecture Notes in Computer Science*, pages 475–488. Springer, 2006.
- [15] Robert Colvin and Lindsay Groves. Formal verification of an array-based nonblocking queue. In *10th International Conference on Engineering of Complex Computer Systems (ICECCS 2005)*, 16-20 June 2005, Shanghai, China, pages 507–516. IEEE Computer Society, 2005.
- [16] Robert Colvin, Simon Doherty, and Lindsay Groves. Verifying concurrent data structures by simulation. *Proceedings of the REFINE 2005 Workshop (REFINE 2005)*, *Electronic Notes in Theoretical Computer Science*, 137:93–110, 2005.

Refinement calculus for logic programs

- [17] Robert Colvin, Ian J. Hayes, and Paul Strooper. Calculating modules in contextual logic program refinement. *Theory and Practice of Logic Programming*, 8(1):1–31, 2008.
- [18] I. J. Hayes, R. Colvin, D. Hemer, R. Nickson, and P. A. Strooper. A refinement calculus for logic programs. *Theory and Practice of Logic Programming*, 2(4–5):425–460, July–September 2002.
- [19] R. Colvin, L. Groves, I. J. Hayes, D. Hemer, R. Nickson, and P.A. Strooper. Developing logic programs from specifications using stepwise refinement. In Maurice Bruynooghe and Kung-Kiu Lau, editors, *Program Development in Computational Logic: A Decade of Research Advances in Logic-Based Program Development*, volume 3049 of *Lecture Notes in Computer Science*, pages 66–89. Springer Verlag, 2004.
- [20] R. Colvin, I. J. Hayes, D. Hemer, and P.A. Strooper. Refinement of higher-order logic programs. In M. Leuschel, editor, *Proceedings of the International Workshop on Logic-based Program Synthesis and Transformation (LOPSTR 2002)*, volume 2664 of *Lecture Notes in Computer Science*, pages 126–143. Springer, 2003.

- [21] R. Colvin, I. J. Hayes, D. Hemer, and P. Strooper. Translating refined logic programs to Mercury. In Michael Oudshoorn, editor, *Proceedings of the Twenty-Fifth Australasian Computer Science Conference*, volume 4 of *Conferences in Research and Practice in Information Technology*, pages 33–40. Australian Computer Society, January 2002.
- [22] D. Hemer, R. Colvin, I. J. Hayes, and P. Strooper. Don't care non-determinism in logic program refinement. In James Harland, editor, *Proceeding of Computing: the Australasian Theory Symposium*, volume 61 of *Electronic Notes in Computer Science (ENTCS)*. Elsevier Science, January 2002.
- [23] R. Colvin, I. J. Hayes, and P. Strooper. A technique for modular logic program refinement. In K.-K. Lau, editor, *Logic Based Program Synthesis and Transformation (LOPSTR 2000)*, *Selected Papers*, volume 2402 of *LNCS*, pages 38–56. Springer, 2001.
- [24] R. Colvin, I. J. Hayes, and P. Strooper. Refining logic programs using types. In Jenny Edwards, editor, *Australasian Computer Science Conference (ACSC 2000)*, pages 43–50. IEEE Computer Society, 2000.
- [25] R. Colvin, I. J. Hayes, and P. Strooper. Data refining logic programs. In Jim Grundy, Martin Schwenke, and Trevor Vickers, editors, *International Refinement Workshop and Formal Methods Pacific 1998*, *Discrete Mathematics and Theoretical Computer Science*, pages 100–116. Springer-Verlag, 1998.
- [26] R. Colvin, I. J. Hayes, R. Nickson, and P. Strooper. A tool for logic program refinement. In D. J. Duke and A. S. Evans, editors, *Second BCS-FACS Northern Formal Methods Workshop*, *Electronic Workshops in Computing*. Springer Verlag, 1997.
- [27] Robert Colvin. *Contextual and Data Refinement for the Refinement Calculus for Logic Programs*. PhD thesis, School of Information Technology and Electrical Engineering, University of Queensland, August 2002.