

forks events correspond to subtasks, join event corresponds to joining the results of subtasks. For practical application, Our approach provides a pattern for the refinement and illustrate it by using an extended atomicity decomposition diagram. We will choose a model decomposition method according to communication style among forks events to decompose the Fork/Join model into multiple sub-models. Then refine the sub-models until the desired level of abstraction is gained. Our approach provides a good framework for modeling Fork/Join parallel programs and showing proof obligations of correctness for such programs.

Acknowledgement

We thank anonymous referees for their suggestions and comments on an earlier version of this paper. The work is supported by the Science and Technology Research Project of Jianxi Province Educational Department (Grant No.209411) and the Scientific Research Foundation for Young Doctoral Teachers of College of Applied Science, Jiangxi University of Science and Technology.

References

- [1] J. R. Abrial, *Modeling in Event B: System and Software Engineering*, Cambridge University Press, London, 2010
- [2] J. R. Abrial, S. Hallerstede, Refinement Decomposition and Instantiation of Discrete Models: Application to Event-B, *Fundamenta Informaticae* 2 (2007) 1-28
- [3] A. T. Khan, X. Cao, Z. Li, and S. Li, "Evolutionary Computation Based Real-time Robot Arm Path-planning Using Beetle Antennae Search", *EAI Endorsed Trans AI Robotics*, vol. 1, pp. 1-10, Jan. 2022.
- [4] Z. Chen, J. Walters, G. Xiao, and S. Li, "An Enhanced GRU Model With Application to Manipulator Trajectory Tracking", *EAI Endorsed Trans AI Robotics*, vol. 1, pp. 1-11, Jan. 2022.
- [5] Y. Yang, L. Liao, H. Yang and S. Li, "An Optimal Control Strategy for Multi-UAVs Target Tracking and Cooperative Competition," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 12, pp. 1931-1947, 2020.
- [6] L. Liao, Y. Yang, "Adaptive radial basis function neural network bi-quadratic functional optimal control for manipulators," *Control Theory & Applications*, vol. 37, no. 1, pp. 47-58, 2020.
- [7] S. Akkara and J. T, "PI Controller Based Switching Reluctance Motor Drives using Smart Bacterial Foraging Algorithm", *EAI Endorsed Trans AI Robotics*, vol. 1, pp. 1-8, Jan. 2022.
- [8] D. Lea, A Java fork/join framework, in: *Proc. ACM 2000 conference on Java Grande, 2000*, pp. 36-43
- [9] M. D. Wael, How to achieve scalable fork/join on many-core architectures, in: *Proc. Annual conference on Systems, programming, and applications: software for humanity, 2012*, pp. 85-86
- [10] A. S. Fathabadi, M. Butler, A systematic approach to atomicity decomposition in event-b, in: *Proc. Software Engineering and Formal Methods, 2012*, pp. 78-93
- [11] A. S. Fathabadi, M. Butler, Applying Event-B atomicity decomposition to a multi media protocol, in: *Proc. Formal Methods for Components and Objects, 2010*, pp. 89-104
- [12] S. Hallerstede, M. Leuschel, D. Plagge, Refinement-animation for Event-B-towards a method of validation, in: *Proc. Abstract State Machines, Alloy, B and Z, 2010*, pp. 287-301
- [13] T. S. Hoang, A. Iliarov, R. A. Silva, et al, A survey on Event-B decomposition, *Electronic Communications of the EASST*, 46 (2011) 1-15
- [14] J. R. Abrial, M. Butler, Hallerstede S, et al, Rodin: an open toolset for modelling and reasoning in Event-B, *International journal on software tools for technology transfer*, 16(2010) 447-466
- [15] A. Edmunds, M. Butler, Linking Event-B and concurrent object-oriented programs, *Electronic Notes in Theoretical Computer Science*, 214 (2008) 159-182.
- [16] P. Bostrom, F. Degerlund, K. Sere, et al, Derivation of concurrent programs by stepwise scheduling of Event-B models, *Formal Aspects of Computing*, 8 (2012) 1-23
- [17] D. Mery, N. K. Singh, Automatic code generation from event-B models, in: *Proc. the Second Symposium on Information and Communication Technology, 2011*, pp. 179-188.
- [18] L. Butler, Decomposition structures for Event-B, in: *Proc. Integrated Formal Methods, 2009*, pp. 20-38
- [19] R. Silva, C. Pascal, M. Butler, et al, Decomposition tool for Event-B, *Software: Practice and Experience*, 41 (2011) 199-208.
- [20] T. S. Hoang, J. R. Abrial, *Event-B decomposition for parallel programs*, Springer Berlin, Heidelberg, 2010
- [21] G. Dupont, Y. Ameur, M. Pantel, et al. Handling refinement of continuous behaviors: a proof based approach with event-B, 2019 International Symposium on Theoretical Aspects of Software Engineering (TASE). IEEE, 2019: 9-16.
- [22] S. Sritharan, T. S. Hoang, Towards generating SPARK from Event-B models, *International Conference on Integrated Formal Methods*. Springer, Cham, 2020: 103-120.
- [23] M. Leuschel, M. Mutz, M. Werth, Modelling and validating an automotive system in classical B and event-B, *International Conference on Rigorous State-Based Methods*. Springer, Cham, 2020: 335-350.
- [24] Sritharan, Sanjeevan, and T. S. Hoang. Towards generating SPARK from Event-B models, *International Conference on Integrated Formal Methods*. Springer, Cham, 2020.
- [25] D. Mohammadsadegh, B. Michael, R. Abdolbaghi, Verifiable code generation from scheduled Event-B models. In *6th International Conference, ABZ 2018*, Southampton, UK, June 5-8, 2018.
- [26] K. Rustan M. Leino. Developing verified programs with Dafny. In Rajeiv Joshi, Peter Muller, and Andreas Podolski, editors, *Verified Software: Theories, Tools, Experiments-4th International Conference, VSTTE 2012*, Philadelphia, PA, USA, January 28-29, 2012. Proceedings, volume 7152 of *Lecture Notes in Computer Science*, page 82. Springer, 2012.