Introduction of timing aspects into Event-B

Nowadays, an incremental development process is increasingly adopted to cope with the complexity of systems. Designers start with a very abstract model that mainly considers functional requirements, and then non-functional requirements are gradually introduced by refinement. Such requirements include, among others, timing aspects. However, such a development paradigm turns out to be inadequate for safety-critical applications in which timing requirements are a cornerstone and need to be considered from the initial development phases to avoid disastrous situations. On the other hand, Event-B [1] is a formal development method with refinement as a central concept that permits building the correct system through an incremental process. Nevertheless, it lacks native support for specifying and verifying timing aspects. To overcome this limit, many research works have studied how to express and verify timing properties in an Event-B specification. These approaches can be classified into two categories: the first one suggests encoding the timing properties directly in Event-B by introducing new variables constrained by invariants and events to represent the time and its progression along with those updating the variables according to the timing constraints [2, 4, 6, 9]. The second category is based on using an existing timing language (like UPPAAL) into which an Event-B specification is translated and augmented by timing properties [3,7,8,10]. A general remark regarding these approaches is that each one is defined to answer a particular problem of timing property verification and thus may be inadequate for a more general problem. In other words, these approaches do not really extend Event-B with time, but it is more an ad-hoc encoding of the discrete-time. Moreover, these approaches are not supported by tools. Our objective is to endow the Event-B method with timing features that allow expressing timing requirements from very early development stages along with their refinement until the implementation of the system. The Event-B Method is refinement-based, so augmenting it with a new notion will require revising the refinement process to preserve clock constraints. The approach introduced in [5] to define a refinement relationship for preserving temporal properties over reconfigurations can be followed to this aim.

The objectives of this partnership are as follows:

1. defining a set of relevant timing constraints and its formal semantics
2. modeling the selected timing constraints in Event-B: clauses + invariants + proof obligations
3. defining a set of refinement rules for these timing constraints.
4. extending the Rodin platform for Event-B development with timing constraints

Profile and skills required
The profile sought is a candidate having obtained a Master in computer science with a solid background in logic.

Supervisors:
Amel Mammar (amel.mammar@telecom-sudparis.eu), IP Paris, Télécom SudParis, SAMOVAR/METHODES
Idir Ait Sadoune (idir.aitsadoune@centralesupelec.fr), Paris-Saclay University, CentraleSupelec, LMF

Duration/ Gratification: 6 months, 600€/month

References

1. J.R. Abrial : Modeling in Event-B - System and Software Engineering. Cambridge University Press, 2010
2. R. Banach, M. J. Butler, S. Qin, N. Verma, and H. Zhu, “Core hybrid event-b I: single hybrid event-b machines,” Science Computer Programming, vol. 105, pp. 92–123, 2015.

3. J. Berthing, P. Boström, K. Sere, L. Tsiopoulos, and J. Vain, “Refinement-based development of timed systems,” in Integrated Formal Methods - 9th International Conference, IFM 2012, Pisa, Italy, June 18-21, 2012. Proceedings (J. Derrick, S. Gnesi, D. Latella, and H. Treharne, eds.), vol. 7321 of Lecture Notes in Computer Science, pp. 69–83, Springer, 2012.
4. D. Cansell, D. Méry, and J. Rehm, “Time constraint patterns for event B development,” in B 2007: Formal Specification and Development in B, 7th International Conference of B Users, Besançon, France, January 17-19, 2007, Proceedings (J. Julliand and O. Kouchnarenko, eds.), vol. 4355 of Lecture Notes in Computer Science, pp. 140–154, Springer, 2007.
5. J. Dormoy, O. Kouchnarenko, and A. Lanoix, “When structural refinement of components keeps temporal properties over reconfigurations,” in FM 2012: Formal Methods - 18th International Symposium, Paris, France, August 27-31, 2012. Proceedings (D. Giannakopoulou and D. Méry, eds.), vol. 7436 of Lecture Notes in Computer Science, pp. 171–186, Springer, 2012.
6. C. Fu and K. Zheng, “Patterns for modeling task-level timing constraints with Event-B,” in 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), pp. 260–266, 2018.
7. A. Iliasov, A. B. Romanovsky, L. Laibinis, E. Troubitsyna, and T. Latvala, “Augmenting Event-B modeling with real-time verification,” in Proceedings of the First International Workshop on Formal Methods in Software Engineering - Rigorous and Agile Approaches, FormSERA 2012, Zurich, Switzerland, June 2, 2012 (S. Gnesi, S. Gruner, N. Plat, and B. Rumpe, eds.), pp. 51–57, IEEE, 2012.
8. H. Peng, X. Zhang, G. Cao, Z. Liu, Y. Jing, and L. Rao, “A time refinement framework based on iUML-B state machine”, Sci. Program., vol. 2021, pp. 6672717:1–6672717:21, 2021
9. M. R. Sarshogh and M. J. Butler, “Specification and refinement of discrete timing properties in event-b,” Electron. Commun. Eur. Assoc. Softw. Sci. Technol., vol. 46, 2011.
10. F. Shokri-Manninen, L. Tsiopoulos, J. Vain, and M. Waldén, “Integration of iuml-b and UPPAAL timed automata for development of real-time systems with concurrent processes,” in Rigorous State-Based Methods - 7th International Conference, ABZ 2020, Ulm, Germany, May 27-29, 2020, Proceedings (A. Raschke, D. Méry, and F. Houdek, eds.), vol. 12071 of Lecture Notes in Computer Science, pp. 186–202, Springer, 2020.