# Modeling Critical Systems with Timing Constraints in Event-B

Faezeh Siavashi[1], Marina Waldén[1], Leonidas Tsiopoulos[1] and Jüri Vain[2]

[1] Åbo Akademi University, Turku, Finland
fsiavash@abo.fi, mwalden@abo.fi, ltsiopou@abo.fi
[2] Tallinn University of Technology, Tallinn, Estonia
vain@ico.ee

## 1 Introduction

The complexity of safety critical systems consisting of software and hardware parts is continuously increasing. Formal methods address the issues of provably correct design offering mathematical techniques to create specifications to develop and verify safety critical systems [1]. They ensure that the implemented systems work correctly according to the defined specifications. In this paper, we study the practical aspects of applying Event-B [1] for modelling and verification of time-critical systems. Event-B has been used for developing industrial strength systems, but it lacks timing support. UPPAAL [8], on the other hand, is a model checker which has a good support for timing. In order to enrich the application areas of Event-B, we aim at extending it with timing aspects from UPPAAL. By adding timing properties to Event-B, we can guarantee provably correct timing design on the same basis as the functional correctness is ensured [3].

Event-B is based on the B-Method and is meant for refinement-based development of distributed and reactive systems where implementation details are added to design specifications in a stepwise manner. The system model is extended with new variables and assignments, and new conditions, e.g. stronger guards and invariants. Event-B comes with the Rodin tool, that provides automatic and interactive discharging of proof obligations [5]. UPPAAL is a model checker with extended timed automata called UPPAAL Timed Automata (UPTA)[2].

Our main contribution is that we exploit the patterns for modeling and refinement of timing properties within UPPAAL and transform these patterns to patterns in Event-B. Hence, we are able to verify that the refined timing specification combined with refined functionality together satisfy the more abstract specification [6]. Our work is exemplified by a case study provided by Danfoss Power Electronics, which was part of the EU-project RECOMP (2010-2013) [4]. The case study is available in detail in [7].

## 2 Model transformation from UPPAAL to Event-B

We model the timing properties of the system in UPPAAL. These models are then transformed to Event-B as follows: (1) Each UPPAAL model location is mapped to a state of Event-B. (2) Each transition between locations in the UPPAAL model is mapped to an event in Event-B. (3) The abstract clock in UPPAAL is mapped to an event in Event-B. (4) The invariants and guards in UPPAAL are modeled to guards in Event-B. (5) The declarations in the UPPAAL model is mapped to invariants and axioms in Event-B, according to the data types of the parameters in UPPAAL.

Real-time systems contain a variety of patterns for timing constraints. In this work, we focus on the two most important and common timing constraints and their refinement patterns: Delay

and Deadline.

*The delay pattern.* In the abstract level delay is modelled as an integer type counter that increases by one at each clock cycle of the system. The refined clock is modeled with the system clock that progresses a certain number of ticks within each abstract clock cycle.

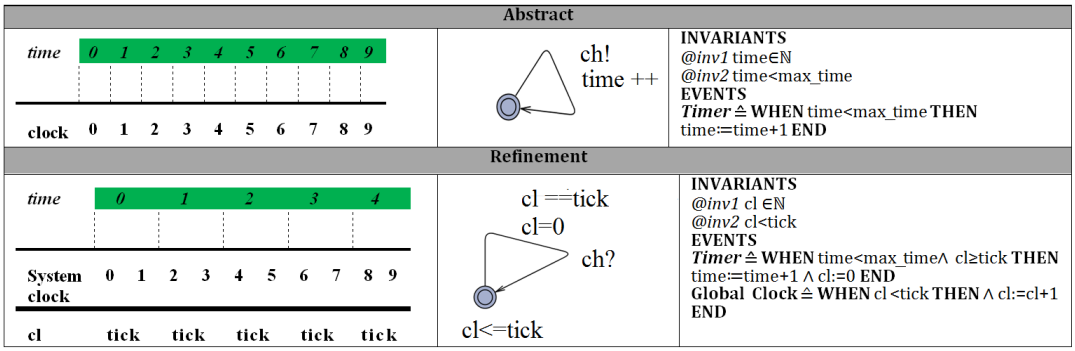


Figure 1: Delay modeling and its refinement in UPPAAL and Event-B models

Delay is modeled in UPPAAL by two UPTA models for the abstract and the refined clock that are translated to models in Event-B (see Figure 1). In the refined Event-B model we introduced a new timing variable 'cl' with a new event **Global_Clock** to model the global timer of the system. The guard of event **Timer** is refined to consider this global timer. We note that the event **Timer** is only modified by strengthening its guard and adding assignments to the new timer variable preserving the old behavior and the invariant. The new event **Global_Clock** assigns the new variable 'cl' while preserving the invariant. Moreover, the upper limit of 'cl' in the model guarantees the non-divergence of the event. Hence, the refined model is a correct refinement of the abstract model.

*Nested Time Interval (Deadline pattern).* We have two transitions $e1$ and $e2$. They occur within a specific time span ($t1 \leq e1 < e2 \leq t2$) that is refined to ($ref\_t1 \leq e1 < e2 < ref\_t2$). In the abstract UPTA model, the deadline of $e1$ is modeled by an invariant ($cl < t2$) and a guard ($cl >= t1$) and in the refinement the deadlines are shorter (see Figure 2). In Event-B, this is modeled by three events **e1_Change**, **e2_Change** and **Timer**. Due to the decreased time intervals of the events, the guards are strengthened. The correctness of the refined deadline interval is ensured by the events **e1_Change** and **e2_Change** preserving the invariant. Since the action part of the events are not changed, it trivially guarantees that the behavior of the model is not changed.

*Case Study.* The Delay and Deadline patterns have been applied on an industrial case study, where a frequency converter with two reset buttons are connected to a pair of redundant processors via a safety module. The reset buttons shutdown the converter whenever there is a difference between power cycles of the motor. There are two different safety functions called SafeStop1 (SS1) and SafeTorqueOff (STO) that can be activated. Whenever Emergency Shutdown (ES) button is pushed for some amount of time (delay), the system will be reset. The reaction is based on two-step reset: first the SS1 signal will be activated and then within a certain time (deadline) the STO signal will be activated to shut down the system. The delay and deadline patterns introduced above were used to refine the ES delay and the deadline time for the STO [7].

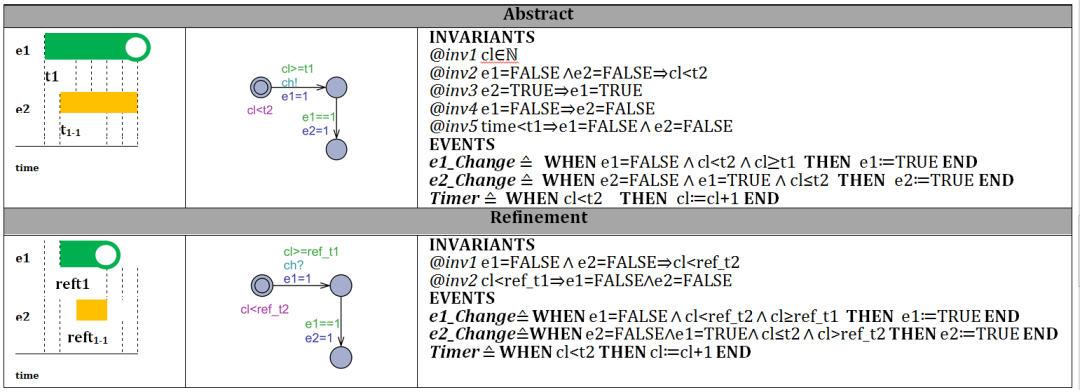


Figure 2: Nested time intervals (deadline) modeling and refinement in UPPAAL and Event-B

# 3 Conclusion

The main problem is defining a clock in Event-B, since it is not similar to a real clock, which models continuous time with identical time slots for each clock cycle. In Event-B, we have defined a clock as a discrete time element which does not necessarily increase continuously. In addition, the clock cycles in a discrete timer do not have the same duration. It is possible that some of the clock cycles take shorter time while others take longer time. This is because of the nature of the Event-B language. The progress of time is dependent on the events in the model rather than on a reference clock that is running with its own rate. In case more than one event are enabled at a time, Event-B can give priority to an event suspending the clock event. Moreover, an event which is enabled will not necessarily be executed.

Modeling time in Event-B mostly covers properties and temporal relations of events. It ensures that if the deadline for the execution of an event is passed and the event missed the deadline, it cannot be executed. If any of the events misses its deadline, then the reliability of the system is not either assured, since the reliability of real-time systems often depends on the response time. Guards and invariants of the model guarantee that if the timer does not prohibit the executions of the other events before the deadline passes, then the events will occur in correct time and order.

# References

[1] J.-R. Abrial. *Modeling in Event-B: System and Software Engineering.* Cambridge University Press, New York, USA, 1st edition, 2010.

[2] J. Bengtsson and W. Yi. Timed automata: Semantics, algorithms and tools. Springer, 2004.

[3] D. Cansell, D. Méry, and J. Rehm. Time constraint patterns for event b development. In the Proc. of *Formal Specification and Development in B*, LNCS 4355. Springer, 2006.

[4] RECOMP. http://atcproyectos.ugr.es/recomp/, last access 10/11/13.

[5] RODIN. Event-B and the Rodin platform. http://www.event-b.org/, last access 10/11/13.

[6] M. R. Sarshogh and M. Butler. Specification and refinement of discrete timing properties in Event-B. In the Proc. of AVoCS 2011. September 2011.

[7] F. Siavashi. Modelling critical systems with time constraints in Event-B. Master's thesis, Åbo Akademi University, 2012.

[8] UPPAAL. http://www.uppaal.org, last access 10/11/12.