An Interval-Based Approach to Modelling Time

Gintautas Sulskus [*], Michael Poppleton [†] and Abdolbaghi Rezazadeh [‡]

University of Southampton

This paper is inspired by our work on the case study of a dual chamber cardiac pacemaker. The pacemaker is a demanding real-time embedded application that interacts with a non-deterministic environment (the heart) via sensors and actuators.

Our implementation involves state based sequencing and timing aspects. We use the Rodin tool to formally model the pacemaker in Event-B. State based sequencing with concurrency elements is generated with iUML [2] – a UML-aided visual formal modelling tool. The part of pacemaker's core functionality a number of interdependent cyclic timing constraints – is implemented following Sarshogh's patterns for discrete timing [1]. We aim to improve modelling techniques, identify the applicability and constraints of used approaches, provide workarounds for identified limitations and suggest new prototype patterns.

Sarshogh denes three types of timing properties: deadline (Figure 1) – response event must occur within time t of trigger event occurring; delay (Figure 2) – response event cannot occur within time t of trigger event occurring; expiry (Figure 3) – response event cannot occur after time t of trigger event occurring.

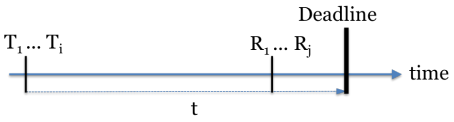

Figure 1: Deadline

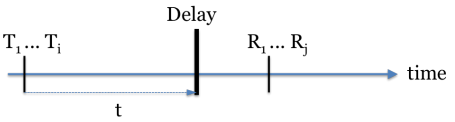

Figure 2: Delay

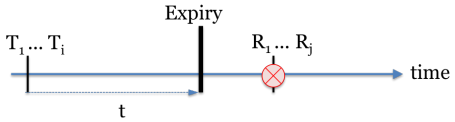

Figure 3: Expiry

In this paper we present a timing interval approach, inspired by limitations found in the pacemaker case study. We extend Sarshogh's work by introducing a redefined timing notation (1), a concept of interval and interrupt event.

We define *interval* as timing entity that has timing properties and can be manipulated by events.

$$Interval(T_1 ... T_i;\ R_1 ... R_j;\ I_1 ... I_k;\ TC_1(t_1) ... TC_n(t_n)) \tag{1}$$

The interval is manipulated by three kinds of events: one of a set of trigger events $T \in T_1 ... T_i$ initiates the interval; a response event $R \in R_1 ... R_j$ terminates the interval; an interrupt event $I \in I_1 ... I_k$ interrupts the interval.

Interval must have at least one timing property $TP(t)$, where t is the duration of the associated timing property.

[*] gs6g10@ecs.soton.ac.uk
[†] mrp@ecs.soton.ac.uk
[‡] ra3@ecs.soton.ac.uk

As a simple example, consider a Lower Rate Interval (2). LRI is the longest time interval that is allowed between consecutive heart contractions. The interval is triggered by either intrinsic or artificial electrical stimulus *sense*, *pace*. It must be responded by the pacemaker stimulus *pace* within $Deadline(t_2)$ time, but no earlier than $Delay(t_1)$. If intrinsic heart activity *sense* occurs, interval is interrupted.

$$LRI(sense, pace;\ pace;\ sense;\ Delay(t_1), Deadline(t_2)) \tag{2}$$

In contrast to the original Sarshogh's patterns, our approach is independent of event sequencing, therefore can be applied on any Event-B model. The approach pattern has a modular design: a standard template code of T, R and I events is instantiated and injected into the target Event-B model as invariants, guards and actions to implement the timing constraint requirements. The template is unaffected by target Event-B model contents. Supported event overloading enables event to serve as T and R or I for multiple intervals at the same time. Proof obligations are automatically discharged with the help of external provers.

Our future plan is to verify that our approach supports Sarshogh's refinement and decomposition patterns; consider variable timing property duration t; perform more case studies; develop a plugin for Event-B code generation (Figure 4); add visualisation support for iUML plug-in and investigate – Event-B timing to code generation.

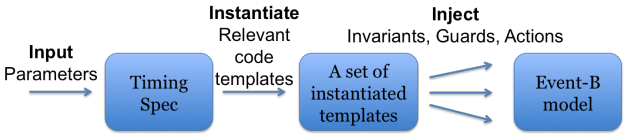

Figure 4: The workflow of the plug-in to generate Event-B timing code

Bibliography

[1] Mohammad Reza Sarshogh. *Extending Event-B with Discrete Timing Properties*. PhD thesis, 2013.

[2] Colin Snook and Michael Butler. UML-B: Formal modelling and design aided by UML. *ACM Transactions on Software Engineering and Methodology*, 15(1):92–122, January 2006.