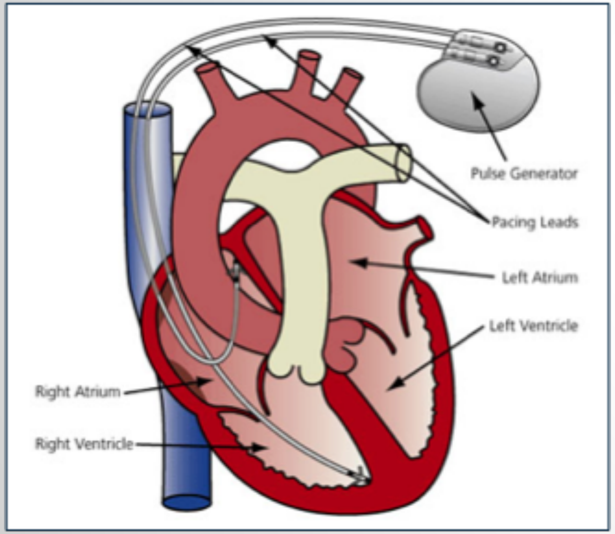
Towards Patterns for Modelling Timing Constraints

Gintautas Sulskus, Michael Poppleton and Abdolbaghi Rezazadeh
University of Southampton

5th Rodin Workshop
June 2014, Toulouse

Pacemaker Case Study



Why Pacemaker:

- Real time safety-critical system
- Dual chamber - concurrency
- PM’s core - ideal heart’s model
  - a set of complex cyclic timing constraints

The process:

- iUML & Sarshogh’s timing pattern combination
- Provide workarounds
- Interval approach

Interval Approach

$\mathbf{Interval}(\{T_1, \ldots, T_i\}, \{R_1 , \ldots, R_j\}, \{I_1, \ldots, I_k\}, \{TP_1(t_1), TP_1(t_2)\})$

- **Interval** is a modelling abstraction that denotes a period of time
  - Characterised by a set of timing properties and can be manipulated by a set of events
  - Multiple instances of the interval supported

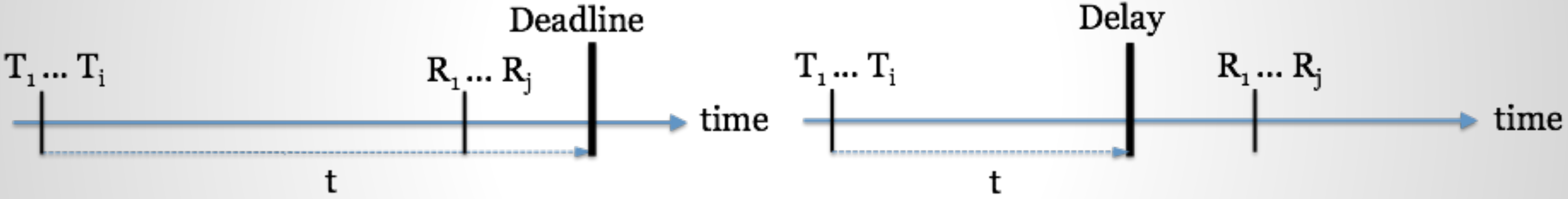
Interval Events

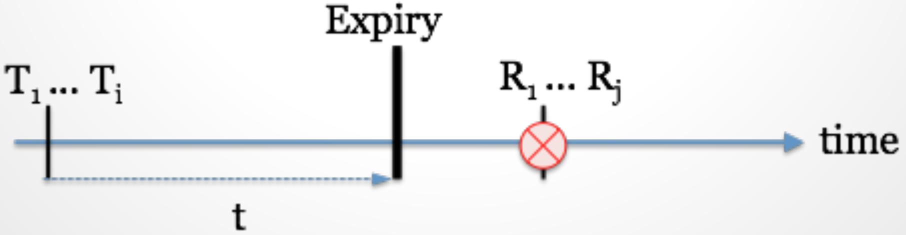
Interval($\{T_1, \ldots, T_i\}, \{R_1, \ldots, R_j\}, \{I_1, \ldots, I_k\}, \{TP(t_1), TP(t_2)\}$)

- Trigger (T) – creates an instance of the interval
  - Required at least 1 event
- Response (R) – always terminates 1 active interval instance
  - Required at least 1 event
  - Constrained by timing properties
- Interrupt (I) – if exists, must interrupt active interval inst.
  - Optional
  - Unconstrained by interval instance existence

Interval Timing Properties

$\text{Interval}(\{T_1, \ldots, T_i\}, \{R_1, \ldots, R_j\}, \{I_1, \ldots, I_k\}, \{TP(t_1), TP(t_2)\})$





Example

LRI({pace, sense}, {pace}, {sense}, {Deadline(t_1), Delay(t_2)})

- We have defined a **L**ower **R**ate **I**nterval
- Interval is triggered by intrinsic or artificial heart stimulus
- The interval can be responded by the pacemaker stimulus no later than t_1 time, but no sooner than t_2
- In case of sensed intrinsic heart activity, interval is interrupted

Improvements

- Changes to model do not affect interval invariants
  - Presents overhead
- Modular, template-based design
- Interrupt event
  - No event replication to handle different cases
- Multiple T, R and I events in the same ref. level
- Supports event overloading
- Automatically discharged POs (abstract level)
  - Can be tough to prove the refinement though

Future Work

- Compare against Sarshogh’s refinement patterns and decomposition support
- Variable timing constraint duration *t*
- More case studies
- Event-B timing to code generation
- Tooling
    - Plugin for Specification to Event-B code generation
    - Visualisation support for iUML plug-in

UNIVERSITY OF Southampton

Thank You!

?