The Cardiac Pacemaker Case Study and its implementation in Safety-Critical Java and Ravenscar Ada

Neeraj Kumar Singh, Andy Wellings, Ana Cavalcanti

University of York, United Kingdom

JTRES, 24-26 October 2012
Outline

1. Context and Motivation
2. Heart System
3. The Cardiac Pacemaker
4. DDDR Pacing Mode
5. The Cardiac Pacemaker Architecture in SCJ
6. Conclusion
Context and Motivation

Context

Evaluation of a concurrency model of a cardiac pacemaker using two programming languages: Safety Critical Java (SCJ) and Ravenscar Ada.

Motivation

To contribute in the area of Grand Challenges.

Implementation of cardiac pacemaker in SCJ.

A comparative study of SCJ and Ravenscar Ada.

To meet the certification standards.

hiJaC Project

Singh, Wellings and Cavalcanti (UoY)
Context and Motivation

Context
Evaluation of a concurrency model of a cardiac pacemaker using two programming languages: Safety Critical Java (SCJ) and Ravenscar Ada.

Motivation
- To contribute in the area of Grand Challenges.
Context and Motivation

Context
Evaluation of a concurrency model of a cardiac pacemaker using two programming languages: Safety Critical Java (SCJ) and Ravenscar Ada.

Motivation
- To contribute in the area of Grand Challenges.
- Implementation of cardiac pacemaker in SCJ.
Context and Motivation

**Context**
Evaluation of a concurrency model of a cardiac pacemaker using two programming languages: Safety Critical Java (SCJ) and Ravenscar Ada.

**Motivation**
- To contribute in the area of Grand Challenges.
- Implementation of cardiac pacemaker in SCJ.
- A comparative study of SCJ and Ravenscar Ada.
Context and Motivation

Context
Evaluation of a concurrency model of a cardiac pacemaker using two programming languages: Safety Critical Java (SCJ) and Ravenscar Ada.

Motivation
- To contribute in the area of Grand Challenges.
- Implementation of cardiac pacemaker in SCJ.
- A comparative study of SCJ and Ravenscar Ada.
- To meet the certification standards.
Context and Motivation

Context
Evaluation of a concurrency model of a cardiac pacemaker using two programming languages: Safety Critical Java (SCJ) and Ravenscar Ada.

Motivation
- To contribute in the area of Grand Challenges.
- Implementation of cardiac pacemaker in SCJ.
- A comparative study of SCJ and Ravenscar Ada.
- To meet the certification standards.
- hiJaC Project
## Timing Intervals in a Single Heart Beat

<table>
<thead>
<tr>
<th>Time Intervals</th>
<th>Time in milliseconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length of a P wave ($T_P$)</td>
<td>110</td>
</tr>
<tr>
<td>Duration of pulse ($T_{pulse}$)</td>
<td>1</td>
</tr>
<tr>
<td>Length of a QRS complex ($T_{QRS}$)</td>
<td>100</td>
</tr>
<tr>
<td>Atrioventricular interval (AVI)</td>
<td>150</td>
</tr>
<tr>
<td>Ventriculoatrial interval (VAI)</td>
<td>850</td>
</tr>
<tr>
<td>Postventricular atrial refractory (PVARP)</td>
<td>350</td>
</tr>
<tr>
<td>Mode Switching Interval (MSI)</td>
<td>500</td>
</tr>
</tbody>
</table>
A pacemaker is an electronic device implanted in the body to regulate the abnormal heart rhythm (bradycardia). Types: 1, 2 and 3-Electrodes.
The Cardiac Pacemaker

Pacemaker

A pacemaker is an electronic device implanted in the body to regulate the abnormal heart rhythm (bradycardia). Types: 1, 2 and 3-Electrodes.

Basic elements of Pacemaker

- Leads
- Pacemaker generator
- Device Control Monitor (DCM)
- Accelerometer
### Operating Modes: NASPE/BPEG Generic Code

<table>
<thead>
<tr>
<th>Category</th>
<th>Chambers Paced</th>
<th>Chambers Sensed</th>
<th>Response to Sensing</th>
<th>Rate Modulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Letters</td>
<td>O-None</td>
<td>O-None</td>
<td>O-None</td>
<td>R-Rate Modulation</td>
</tr>
<tr>
<td></td>
<td>A-Atrium</td>
<td>A-Atrium</td>
<td>T-Triggered</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V-Ventricle</td>
<td>V-Ventricle</td>
<td>I-Inhibited</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D-Dual(A+V)</td>
<td>D-Dual(A+V)</td>
<td>D-Dual(T+I)</td>
<td></td>
</tr>
</tbody>
</table>

i.e. AOO, VOO, AAI, AAT, VVI, DVIR, AOOR, DDDR etc.

**Periodic stimuli:** (AOO, VOO and DOO)

**Aperiodic stimuli:** (AAI, VVI, DDD, DDI, etc.)
Basic Architecture of a Cardiac Pacemaker

MICROCHIP PIC Processor

- Atrial Pacing Pulse Generator
- Atrial Heartbeat Sensor
- Rate Modulation Sensor
- Ventrical Heartbeat Sensor
- Ventrical Pacing Pulse Generator
The DDDR Pacing Scenarios

(A)

(B)

(C)

(D)
Rate Modulation and Mode Switching

- Rate Modulation

[Diagram showing daily activities: wake up, get up, go to bathroom, have a shower, have breakfast, listen to the radio, go to work, come home, make dinner, phone (or call) a friend, watch TV, go to bed]
Rate Modulation and Mode Switching

- Rate Modulation
- Mode Switching \((\text{DDDR} \leftrightarrow \text{DDIR})\)
The Required DDDR Pacing Cycle

- **Set VAI Countdown Timer**
- **Wait PVARP**
- **Atrium Sensed**
- **VAI Expired**
- **Pace Atrium**
- **Set AVI Countdown Timer**
- **Ventricle Sensed**
- **AVI Expired**
- **Pace Ventricle**

**Update VAI**

- **Change Rate**
  - **yes**
  - **no**

**Singh, Wellings and Cavalcanti (UoY) Cardiac Pacemaker in SCJ & Ravenscar Ada JTRES, 24-26 October 2012**
Safety Critical Java

Start → Select Mission → Mission Initialization → Mission Execution → Mission Cleanup → Halt
The Cardiac Pacemaker Architecture in SCJ

MICROCHIP
PIC Processor

Atrial Pacing Pulse Generator
Atrial Heartbeat Sensor
Rate Modulation Sensor
Ventrical Heartbeat Sensor
Ventrical Pacing Pulse Generator

Aperiodic Handler
Periodic Handler
Periodic Handler
Periodic Handler
Aperiodic Handler
Algorithm for the Atrium Sensor Periodic Handler

1. **AtriumActivity Occurred**
   - No
   - **Time elapsed since lastVentricularActivity >= PVARP**
     - Yes
     - Read sensor
     - Intrinsics Atrium activity sensed
       - Yes
       - Set lastAtriumActivity time
       - Set atriumActivityOccurred
       - Clear ventricularActivityOccurred
       - No
       - Wait for next release event
     - No
   - Yes
   - Release atrium pacer
     - Set atriumActivityOccurred

2. **Time elapsed since lastVentricularActivity >= VAI**
   - Yes
   - Release atrium pacer
     - Set atriumActivityOccurred
   - No
   - **AtriumActivity Occurred**
     - No
     - **Time elapsed since lastVentricularActivity >= PVARP**
       - Yes
       - Read sensor
       - Intrinsics Atrium activity sensed
         - Yes
         - Set lastAtriumActivity time
         - Set atriumActivityOccurred
         - Clear ventricularActivityOccurred
       - No
       - Wait for next release event
     - No
   - Yes
   - Release atrium pacer
     - Set atriumActivityOccurred

3. **Wait for next release event**

Singh, Wellings and Cavalcanti (UoY) Cardiac Pacemaker in SCJ & Ravenscar Ada JTRES, 24-26 October 2012 16 / 23
Algorithm for the Atrium Pacer Aperiodic Handler

Time elapse since atrium activity >= VAI

Yes
Set lastAtriumActivityTime
Pace current on
Sleep 1 ms
Pace current off
Clear ventricularActivityOccurred

No
Wait next release

Set lastAtriumActivityTime
Pace current on
Sleep 1 ms
Pace current off
Clear ventricularActivityOccurred
Algorithm for the Ventricle Sensor Periodic handler

1. **Read sensor**
2. **Intrinsic ventricular activity sensed**
   - **Yes**: Set lastVentricularActivity
   - **No**: Clear atriumActivityOccurred

3. **Wait for next release event**
   - **Yes**: Set ventricularActivityOccurred
   - **No**: Wait for next release event

4. **Time elapsed since lastAtriumActivity > AVI**
   - **Yes**: Release ventricular pacer
     - **Yes**: Set ventricularActivityOccurred
     - **No**: Request mission termination
   - **No**: Mode change needed

5. **Mode change needed**
   - **Yes**: Request mission termination
   - **No**: Wait for next release event
Algorithm for the Ventricle Pacer Aperiodic Handler

- If Time elapsed since lastVentricularActivity > PVARP+AVI:
  - Set lastVentricularActivity
  - Pulse current on for 1ms
  - Pulse current off
  - Clear atriumActivityOccurred

- Otherwise, Wait for next release event
public void handleEvent() {

    AbsoluteTime now = Clock.getRealtimeClock().getTime();

    interval = now.subtract(DDDR_PMMission.lastAtriumActivityTime).getMilliseconds();

    if(interval > (DDDR_PMMission.PaceInterval - DDDR_PMMission.AVI)){

        pm_A.Pace_ON_A(); //Pace ON

        RealtimeThread.sleep(new RelativeTime(DDDR_PMMission.PacingLength,0)); //sleep

        pm_A.Pace_OFF_A(); //Pace OFF

        DDDR_PMMission.Activity_V_Occured =false; //Reset Ventricle flag

        //Save pacing Time of Atrium
        DDDR_PMMission.lastAtriumActTime.set(now.getMilliseconds(),now.getNanoseconds());
    }
}
Cardiac Pacemaker Simulator
Conclusion

- Implementation of the cardiac pacemaker in SCJ and Ravenscar Ada.

Future Work

Formal verification of the cardiac pacemaker in Circus.
Conclusion

- Implementation of the cardiac pacemaker in SCJ and Ravenscar Ada.
- This work suggests the inclusion of one-shot timer in the final release of SCJ.

Future Work

- Formal veriﬁcation of the cardiac pacemaker in Circus.

Singh, Wellings and Cavalcanti (UoY)  Cardiac Pacemaker in SCJ & Ravenscar Ada  JTRES, 24-26 October 2012
Conclusion

- Implementation of the cardiac pacemaker in SCJ and Ravenscar Ada.
- This work suggests the inclusion of one-shot timer in the final release of SCJ.
- Ada solution requires a dummy task to avoid from the premature termination.
Conclusion

- Implementation of the cardiac pacemaker in SCJ and Ravenscar Ada.
- This work suggests the inclusion of one-shot timer in the final release of SCJ.
- Ada solution requires a dummy task to avoid from the premature termination.

Future Work

Formal verification of the cardiac pacemaker in Circus.
Conclusion

- Implementation of the cardiac pacemaker in SCJ and Ravenscar Ada.
- This work suggests the inclusion of one-shot timer in the final release of SCJ.
- Ada solution requires a dummy task to avoid premature termination.

Future Work

- Formal verification of the cardiac pacemaker in Circus.