Automata for Real-time Systems

B. Srivathsan

Chennai Mathematical Institute
In this lecture

An academic case-study that investigates methods to build more reliable pacemakers
Lecture 10: Towards reliable pacemakers
References

Modeling and verification of a dual chamber implantable pacemaker

Jiang, Pajic, Moarref, Alur, Mangharam. TACAS’12

Heart-on-a-chip: A closed-loop testing platform for implantable pacemakers

Jiang, Radhakrishnan, Sampath, Sarode, Mangharam. 2013

mlab.seas.upenn.edu
Heart and pacemaker basics

Presentation of Zhihao Jiang (U Penn)
Pacemaker software

In-built *algorithms* to *detect* and *terminate* various abnormal heart conditions
Pacemaker software

In-built algorithms to detect and terminate various abnormal heart conditions

At least 6 implanted medical devices were recalled in 2010 due to likely software defects

Killed by Code: Software Transparency in Implantable Medical Devices

Karen Sandler, Lysandra Ohrstrom, Laura Moy, Robert McVay
Two possible solutions for more reliable devices:

- **Model-based system/software design**
- **Closed-loop testing**
Model-based system/software design

Heart automaton

Pacemaker automaton

Simulink model

UPPAAL

Verification

UPP2SF tool

Simulation

Code generation

Conformance testing

Heart on chip

Pacemaker

Simulink

(Simulink is a commercial tool developed by Mathworks Inc.)
Closed-loop testing

In this figure, the platform is testing a Boston Scientific pacemaker. The heart-on-a-chip testing platform consists of a heart implementation on a programmable chip (FPGA) and an analog interface for signal isolation and attenuation to interface with a pacemaker. The heart-on-a-chip platform Fig. 1 utilizes the model-based framework for verification and testing of medical devices. The open-loop heart signals to the device and evaluating the device output, is not taken into account. Thus, device software interactions will not be captured during testing.

Heart-on-a-Chip: A Closed-loop Testing Platform for Implantable Pacemakers
Zhihao Jiang, Sriram Radhakrishnan, Varun Sampath, Shilpa Sarode, Rahul Mangharam

The heart-on-a-chip platform consists of a heart implementation on an programmable chip (FPGA) and an analog interface for signal isolation and attenuation to interface with a pacemaker. In this figure, the platform is testing a Boston Scientific pacemaker.

Testbench

Heart on chip

Pacemaker

Conformance testing
Coming next: Modeling and verification of heart and pacemaker
Heart as a timed automaton
Abstract electrical conduction system of heart into nodes and paths

Picture credits: A Simulink hybrid heart model for quantitative verification of cardiac pacemakers

Chen et. al. HSCC’13
Parameters $T_{rest\_max}$, $T_{RRP\_min}$, etc. chosen acc. to node placement and patient history.
Heart automaton $\textbf{H}$: $N_1 \parallel P_1 \parallel N_2 \parallel P_2 \parallel \ldots \parallel N_k$

$N_i$ Node automaton

$P_i$ Path automaton

$k$ Number of nodes to which heart is abstracted

$\parallel$ Parallel composition (asynchronous product construction)
Pacemaker as a timed automaton
Heart-pacemaker interaction

\[ N^1. Act\_Path! \rightarrow Aget! \]
\[ N^2. Act\_Path! \rightarrow Vget \]

\( N^1 \) node at atrial lead
\( N^2 \) node at ventricular lead
In Abstraction 3, we replace the blocking behavior of the Non-deterministic Conduction 3.5 Abstraction 3: Replacing Blocking with...
Fig. 7. (a) LRI component delivers AP! event if the V-A delay exceeds TLRI-TAVI; (b) AVI component delivers VP! if the A-V delay exceeds TAVI while the V-V delay is longer than TURI; (c) URI component keeps track of the V-V delay; (d) PVARP component filters certain Aget! from the heart and generates AS!; (e) VRP component filters certain Vget! from the heart and generates VS! and for ventricular lead N2. Act path! Vget! VS! VP! N2.Act node! VP! N2.Act node! The pacemaker accordingly generates atrial or ventricular pacing actions AP! N1.Act node! VP! for ventricular lead N2. Act path! Vget! VS! VP! N2.Act node! VP! N2.Act node! We now present our pacemaker model within the closed-loop heart-pacemaker system.

4.1 Basic DDD pacemaker modeling

The DDD pacemaker has 5 basic timing cycles triggered by events, as shown in Fig. 6(b). We decomposed our pacemaker model into 5 components which correspond to the 5 counters.

$$P = LRI \parallel AVI \parallel URI \parallel PVARP \parallel VRP.$$
Heart-pacemaker automaton: $H \parallel P$
An algorithm for Endless Loop Tachycardia
Endless Loop Tachycardia (ELT)

Slides of Zhihao Jiang
**ELT-detection:** If VP-AS pattern within 500ms for at least 8 times

**ELT-termination:** Increase PVARP to 500ms once

---

**Pacemaker** $P_1$: $LRI || AVI || URI || PVARP' || VRP || ELTct || VPAS
Is the modified pacemaker safe?

**Question 1:** Are 2 ventricular events within time?

Check in UPPAAL if in $H \parallel P_1 \parallel PLRI_{test}$, all paths satisfy

$$PLRI_{test}.t \leq TLRI$$
Is the modified pacemaker safe?

**Question 2:** Are 2 ventricular events very fast?

Check in UPPAAL if in $H \parallel P_1 \parallel \text{PURI}_{test}$, all paths satisfy

$$\text{PURI}_{test}.t \geq TURI$$
Each time new algorithm is added, model it and check if basic safety properties are satisfied
Take-home

▶ Model-based system/software design
▶ Closed-loop testing