

1. PROJECT

ROOT_OBJECTS

--|examples_AADL\Pacemaker|--

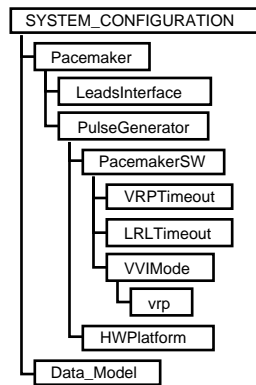
END

1.1. Project Description

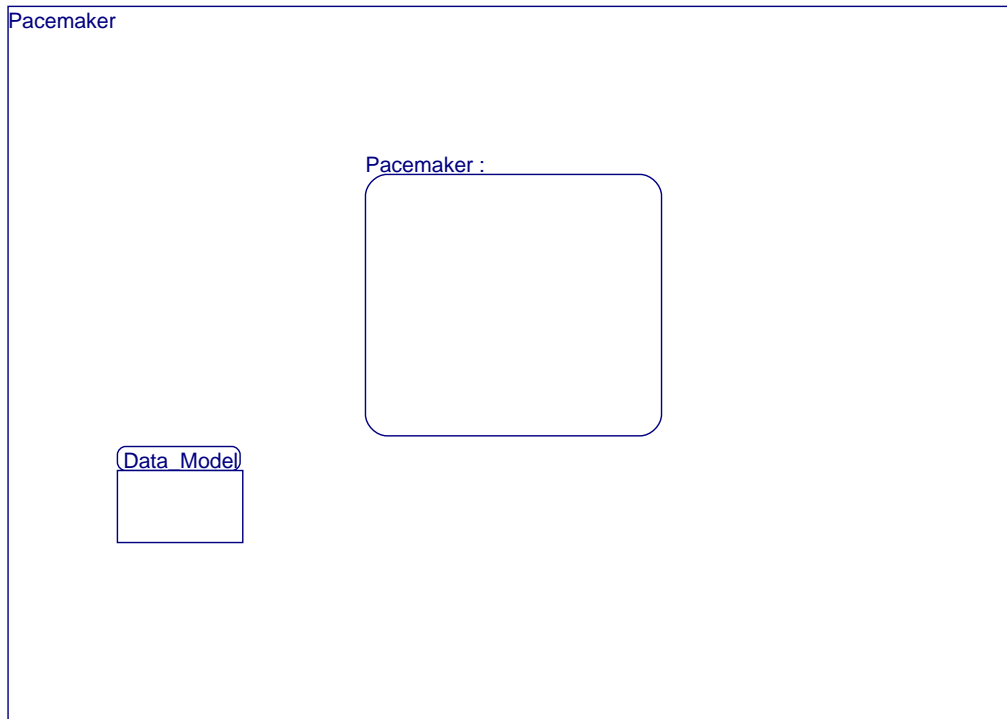
PACEMAKER CONTROL SOFTWARE

This example shows a simplified model of a pacemaker control software.
The AADL Behavior Annex is used to express the precise timing constraints.
The model can be simulated inAADL Inspector with a set of test scenarios.

1.2. Design Tree



1.3. AADL Diagram



2. SYSTEM Pacemaker IS

2.1. DESCRIPTION

2.1.1. PROBLEM

2.1.1.1. Statement of the Problem (text)

This simplified Pacemaker model has been developed to demonstrate the modelling capabilities of the Stood for AADL tool features of the AADL Inspector tool.

The formal specification of the behavior of the pacemaker are expressed in BLESS, whereas the "executable" implementation uses the AADL Behavior Annex.

The design documentation and the AADL source code are both automatically generated from the Stood model.

2.1.1.2. Referenced Documents

* PACEMAKER System Specification

Copyright 2007 Boston Scientific

January 3, 2007

* BA_BLESS example: VVI mode pacing

Brian Larson

January 30, 2016

2.1.2. SOLUTION

2.1.2.1. General Strategy (text)

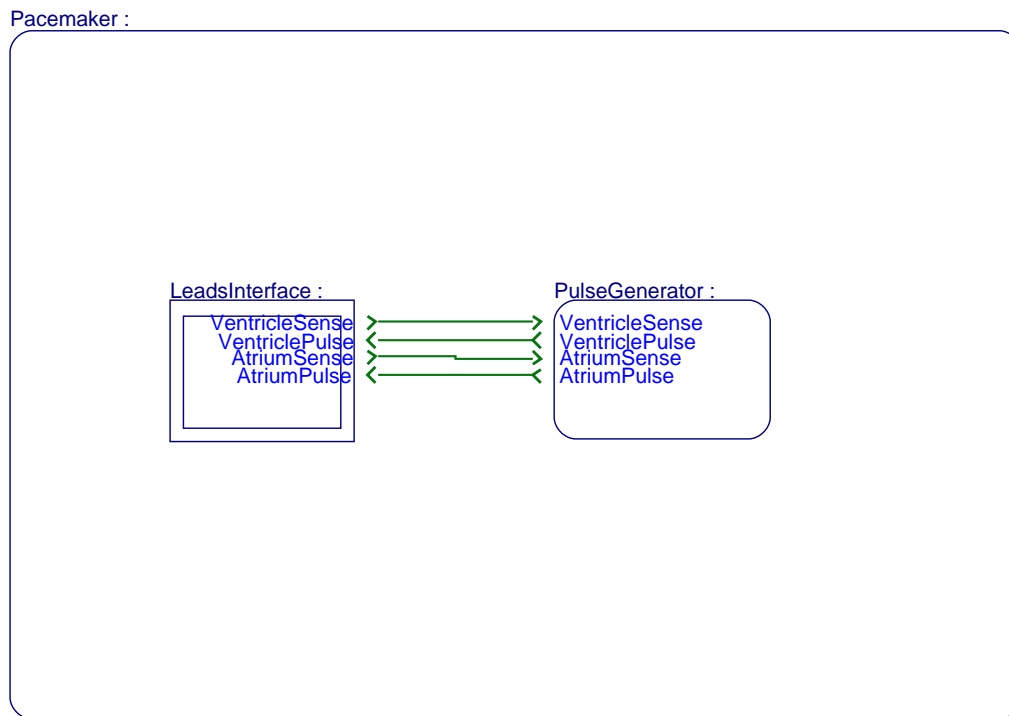
The PACEMAKER system consists of three major components:

- * Device (also called the pulse generator or PG)
- * Device Controller-Monitor (DCM) and associated software

* Leads

In this model, only the embedded part of the system is represented, i.e. the Pulse Generator and its Leads Interface.

2.1.2.2. AADL Diagram



2.1.3. PROPERTIES

2.1.3.1. Predeclared Deployment Properties

2.1.3.2. Predeclared Thread Properties

2.1.3.3. Predeclared Timing Properties

2.2. IMPLEMENTATION

2.2.1. SUBCOMPONENTS

LeadsInterface ;

PulseGenerator ;

2.2.2. BEHAVIOR

2.2.2.1. BEHAVIOR ANNEX

3. DEVICE LeadsInterface IS

3.1. DESCRIPTION

3.1.1. PROBLEM

3.1.1.1. Statement of the Problem (text)

This component represents the analog front end that detects mV signals from the heart through the lead, and causes heart

applied for 0.4 ms

3.1.2. SOLUTION

3.1.3. PROPERTIES

3.1.3.1. Predeclared Deployment Properties

3.1.3.2. Predeclared Thread Properties

3.1.3.3. Predeclared Timing Properties

3.2. IMPLEMENTATION

3.2.1. BEHAVIOR

3.2.1.1. BEHAVIOR ANNEX

4. SYSTEM PulseGenerator IS

4.1. DESCRIPTION

4.1.1. PROBLEM

4.1.1.1. Statement of the Problem (text)

The device monitors and regulates a patient's heart rate.

The device detects and provides therapy for bradycardia conditions.

The device provides programmable, single- and dual-chamber, rate-adaptive pacing, both permanent and temporary.

In adaptive rate modes, an accelerometer is used to measure physical activity resulting in a sensor indicated rate for pacing

The device is programmed and interrogated via bi-directional telemetry from the Device Controller-Monitor (DCM).

This allows the physician to change the operating mode or parameters of the device non-invasively after implantation.

4.1.2. SOLUTION

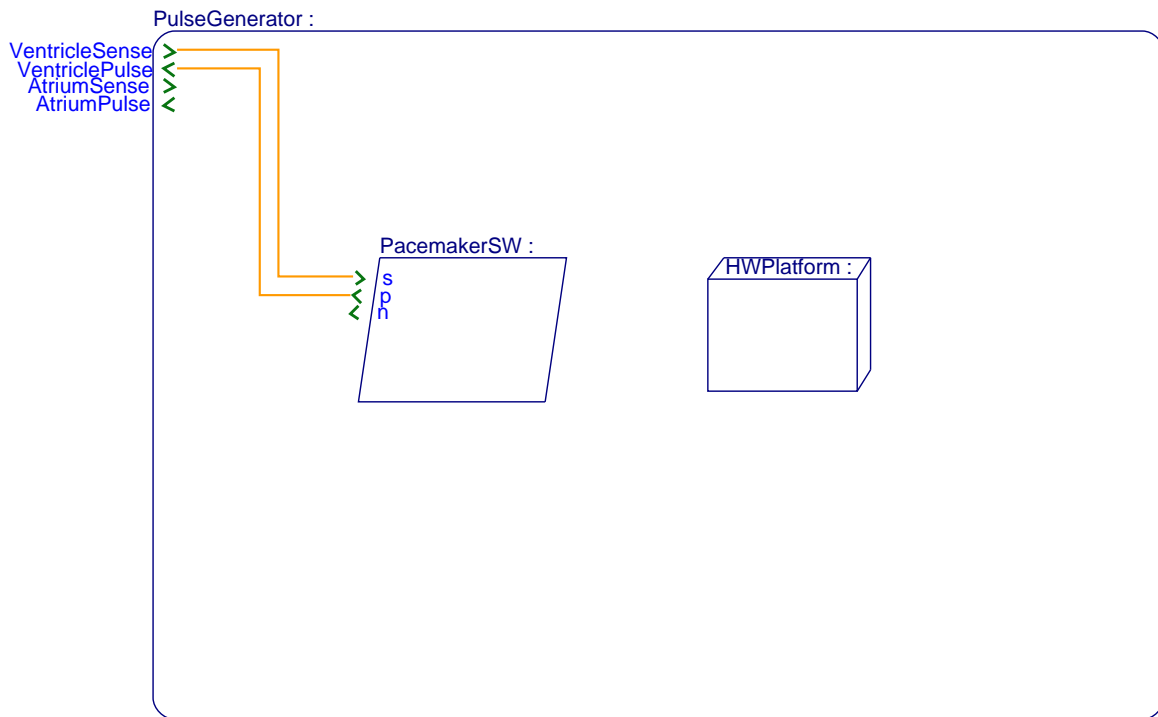
4.1.2.1. General Strategy (text)

The hardware platform is represented by a processor

The pacemaker application software is represented by a process

An Actual Processor Binding property allocates the software application to the processor

4.1.2.2. AADL Diagram



4.1.3. PROPERTIES

4.1.3.1. Predeclared Deployment Properties

4.1.3.1.1. Actual_Processor_Binding

(`reference(HWPlatform)`) applies to `PacemakerSW`

4.1.3.2. Predeclared Thread Properties

4.1.3.3. Predeclared Timing Properties

4.2. IMPLEMENTATION

4.2.1. SUBCOMPONENTS

```
PacemakerSW;
HWPlatform;
```

4.2.2. BEHAVIOR

4.2.2.1. BEHAVIOR ANNEX

5. PROCESS PacemakerSW IS

5.1. DESCRIPTION

5.1.1. PROBLEM

5.1.1.1. Statement of the Problem (text)

The pacemaker software implements the bradycardia operating modes.

In this version, only the VVI mode is supported:

- * the Ventricular chamber is sensed.
- * the Ventricular chamber is paced.
- * a sense Inhibits a pending pace.

5.1.2. SOLUTION

5.1.2.1. General Strategy (text)

The Pacemaker Software behavior can be tested as follows (in VVI mode):

- * Test 1) No sensing.

The thread will put out an event on the "p" port every 1000 ms.

- * Test 2) Normal rhythm.

Put an event on the "s" port every 900 ms. The thread will put an event out the "n" port each dispatch.

- * Test 3) Ignore sense in VRP.

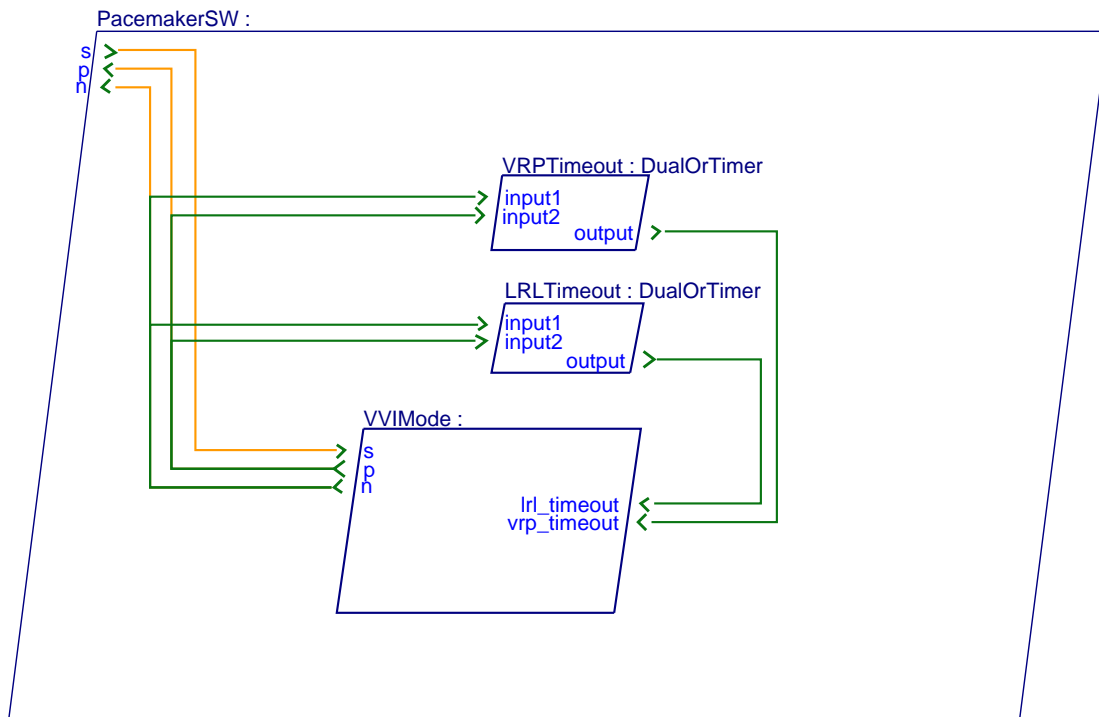
Wait 1000 ms for the first pace; 200 ms later put an event on the "s" port. The next pace will occur at 2000 ms.

- *Test 4) Pace after sense.

Wait 1000 ms for the first pace; 200 ms later put an event on the "s" port, which will be ignored.

At 1400 ms put out another event on the "s" port. Expect the next pace at 2400 ms.

5.1.2.2. AADL Diagram



5.1.3. PROPERTIES

5.1.3.1. Predeclared Deployment Properties

5.1.3.2. Predeclared Thread Properties

5.1.3.3. Predeclared Timing Properties

5.2. IMPLEMENTATION

5.2.1. SUBCOMPONENTS

VRPTimeout ;
LRLTimeout ;
VVIMode ;

5.2.2. BEHAVIOR

5.2.2.1. BEHAVIOR ANNEX

6. THREAD VRPTimeout IS

6.1. DESCRIPTION

6.1.1. PROBLEM

6.1.1.1. Statement of the Problem (text)

The VRPTimeout thread sends an event 300ms after the last normal beat (n) or forced pace (p)
For the purpose of the simulation, this value has been divided by 10.

6.1.2. SOLUTION

6.1.2.1. General Strategy (text)

VRPTimeout is an instance of the DualOrTimer.

The DualOrTimer component has the following behavior:

* receipt of event input1 or input2 sets the timer

* if no event has been received before the specified period, the timer is reset and the output event is sent.

It implemented by a thread with a Timed Dispatch Protocol and a Behavior Annex subclause.

6.1.3. PROPERTIES

6.1.3.1. Predeclared Deployment Properties

6.1.3.2. Predeclared Thread Properties

6.1.3.2.1. Dispatch_Protocol

Timed

6.1.3.2.2. Priority

10

6.1.3.3. Predeclared Timing Properties

6.1.3.3.1. Period

30 ms

6.2. IMPLEMENTATION

6.2.1. BEHAVIOR

6.2.1.1. Behavior Description

The thread can be dispatched on receipt of one of its input ports or after a fixed amount of time since the last dispatch (time). The timeout delay is given by the Period property.

No explicit action is required when the thread is dispatched by an input port (implicit action is that it sets the timer).

A send output action is performed only in the case the thread is dispatched by the timeout event.

6.2.1.2. BEHAVIOR ANNEX

6.2.1.2.1. Behavior Specification (aadl)

```
STATES
  s1 : INITIAL COMPLETE FINAL STATE;
TRANSITIONS
  t1 : s1 -[ ON DISPATCH input1 ]-> s1;
  t2 : s1 -[ ON DISPATCH input2 ]-> s1;
  t3 : s1 -[ ON DISPATCH TIMEOUT ]-> s1 { output! };
```

7. THREAD LRLTimeout IS

7.1. DESCRIPTION

7.1.1. PROBLEM

7.1.1.1. Statement of the Problem (text)

The LRLTimeout thread sends an event 1000ms after the last normal beat (n) or forced pace (p). For the purpose of the simulation, this value has been divided by 10.

7.1.2. SOLUTION

7.1.2.1. General Strategy (text)

LRLTimeout is an instance of the DualOrTimer.

The DualOrTimer component has the following behavior:

* receipt of event input1 or input2 sets the timer

* if no event has been received before the specified period, the timer is reset and the output event is sent.

It implemented by a thread with Hybrid Dispatch Protocol and a Behavior Annex subclause.

7.1.3. PROPERTIES

7.1.3.1. Predeclared Deployment Properties

7.1.3.2. Predeclared Thread Properties

7.1.3.2.1. Dispatch_Protocol

Timed

7.1.3.2.2. Priority

9

7.1.3.3. Predeclared Timing Properties

7.1.3.3.1. Period

100 ms

7.2. IMPLEMENTATION

7.2.1. BEHAVIOR

7.2.1.1. Behavior Description

The thread can be dispatched on receipt of one of its input ports or after a fixed amount of time since the last dispatch (time

The timeout delay is given by the Period property.

No explicit action is required when the thread is dispatched by an input port (implicit action is that it sets the timer).

A send output action is performed only in the case the thread is dispatched by the timeout event.

7.2.1.2. BEHAVIOR ANNEX

8. THREAD VVIMode IS

8.1. DESCRIPTION

8.1.1. PROBLEM

8.1.1.1. Statement of the Problem (text)

The VVIMode thread reacts to sense signal and generates pulse signals.

The expected behavior is:

- * when the heart is beating fast enough, do nothing.
- * when the heart has not had a beat for 1000 ms (Irl), cause a pace
- * if the sense comes too soon after a beat, <300 ms (vrp), ignore it.

8.1.2. SOLUTION

8.1.2.1. General Strategy (text)

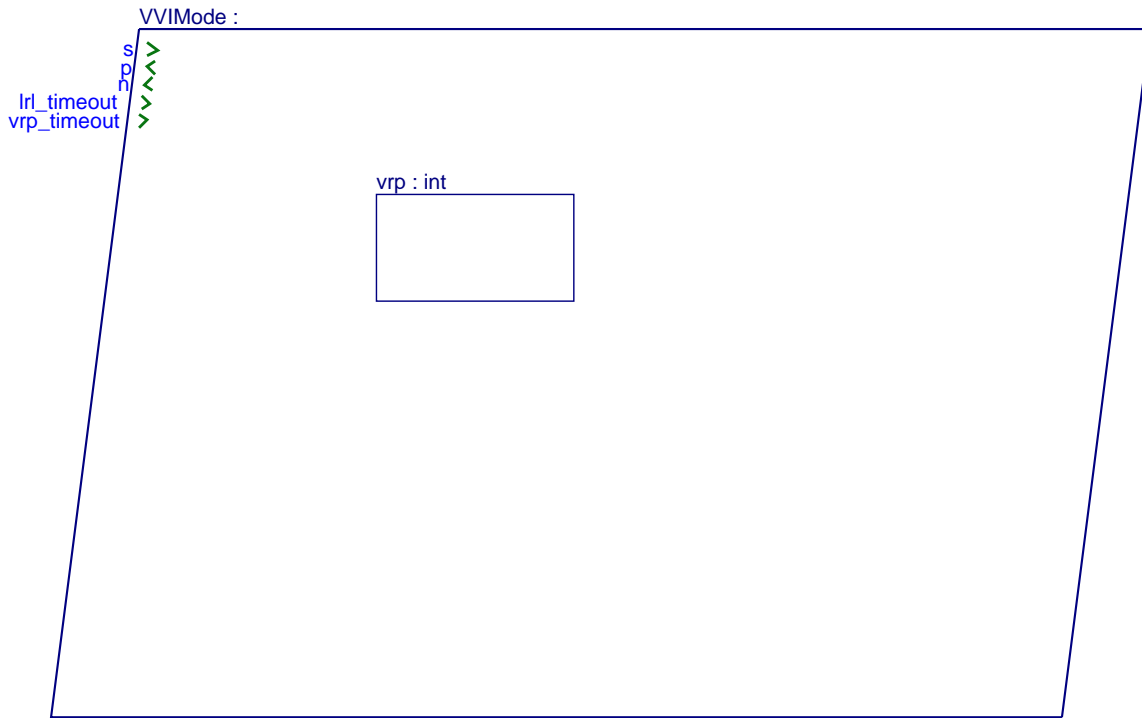
The VVIMode component can be triggered by its input event ports:

- * s: signals a normal heart beat that has been detected by the Pulse Generator device.
- * vrp_timeout: signals that the last beat occurred more than 300ms ago
- * Irl_timeout: signals that the last beat occurred more than 1000ms ago

It is implemented by a thread with Aperiodic Dispatch Protocol and a Behavior Annex subclause.

A data subcomponent is used to store the VRP state of the thread across the successive dispatches.

8.1.2.2. AADL Diagram



8.1.3. PROPERTIES

8.1.3.1. Predeclared Deployment Properties

8.1.3.2. Predeclared Thread Properties

8.1.3.2.1. Dispatch_Protocol

Aperiodic

8.1.3.2.2. Priority

5

8.1.3.3. Predeclared Timing Properties

8.2. IMPLEMENTATION

8.2.1. SUBCOMPONENTS

vrp;

8.2.2. BEHAVIOR

8.2.2.1. Behavior Description

When the thread is dispatched by the vrp_timeout event, the action is to set the vrp data subcomponent value to 0 (out of the Ventricle Refractory Period).

When the thread is dispatched by the s (sense) event, if the vrp value is 1 (in the Ventricle Refractory Period), ignore it; otherwise, generate a p event (normal heart beat), and set the vrp value to 1.

When the thread is dispatched by a lrl_timeout event, generate a p event (pace) and set the vrp value to 1.

8.2.2.2. BEHAVIOR ANNEX

8.2.2.2.1. Behavior Specification (aadl)

```

STATES
  s1 : INITIAL COMPLETE FINAL STATE;
TRANSITIONS
  t0 : s1 -[ ON DISPATCH vrp_timeout ]-> s1
    { vrp := 0 };
  t1 : s1 -[ ON DISPATCH s ]-> s1
    { if (vrp = 0) n!; vrp := 1 end if };
  t2 : s1 -[ ON DISPATCH lrl_timeout ]-> s1
    { p!; vrp := 1 };

```

9. PROCESSOR HWPlatform IS

9.1. DESCRIPTION

9.1.1. PROBLEM

9.1.1.1. Statement of the Problem (text)

The HWPlatform component represents the execution platform of the Pacemaker software.

9.1.2. SOLUTION

9.1.2.1. General Strategy (text)

The HWPlatform is implemented by a Processor running a POSIX HPF scheduler.

9.1.3. PROPERTIES

9.1.3.1. Predeclared Deployment Properties

9.1.3.1.1. Scheduling_Protocol

(HPF)

9.1.3.2. Predeclared Thread Properties

9.1.3.3. Predeclared Timing Properties

9.2. IMPLEMENTATION

9.2.1. BEHAVIOR

9.2.1.1. BEHAVIOR ANNEX