CARDIOTECHNIX 2014

2d International Congress on Cardiovascular Technology

Rome, October 25–26, 2014

Integrated Simulation of Implantable Cardiac Pacemaker Software and Heart Models

C. Bernardeschi, A. Domenici

Dipartimento di ingegneria dell'informazione, Università di Pisa

{cinzia.bernardeschi,andrea.domenici}@iet.unipi.it

P. Masci

School of Electronic Engineering and Computer Science (EECS), Queen Mary University of London

p.m.masci@qmul.ac.uk

Validating ICP Software by simulation

Software incorporated in implantable cardiac pacemakers must be **demonstrably** safe and effective.

- *Correctness of design* wrt safety requirements can be assessed by *formal verification*:
  - Will the ICP work without malfunctions?
- *Adequacy of behavior* must be **validated** by *medical domain experts*:
  - Will the ICP perform as expected?

Realistic simulations facilitate interaction between software experts and medical experts.

Modeling techniques

Simulation requires defining a **model** of the simulated system.

Different types of models are best fit for different systems:

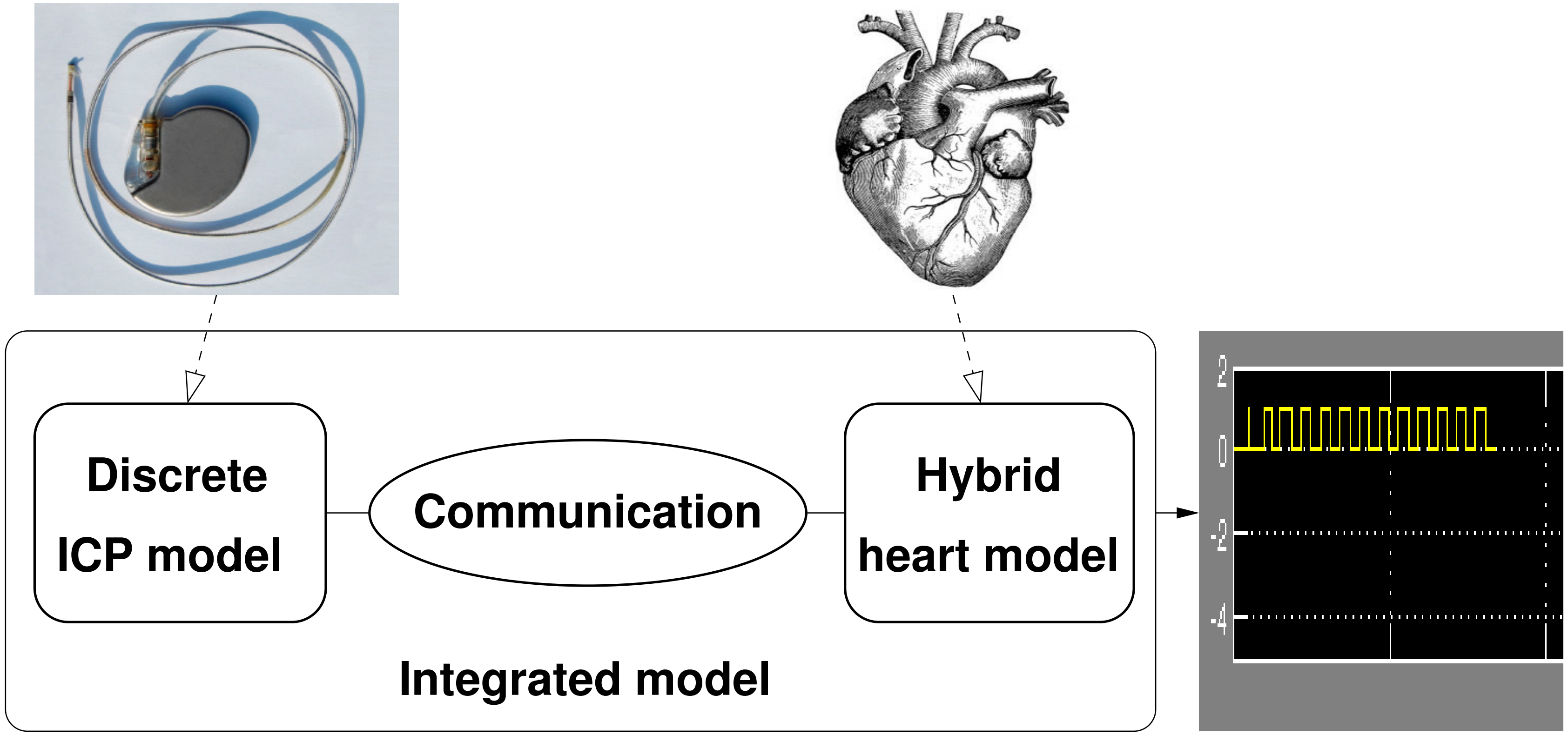
- **Discrete** models for *digital* systems (such as computers, controllers, and their software);
- **hybrid** models for *physiological* systems (such as the heart).

The heart-ICP system requires two interacting models, a discrete and a hybrid one.

But the two models *“speak” different languages*!

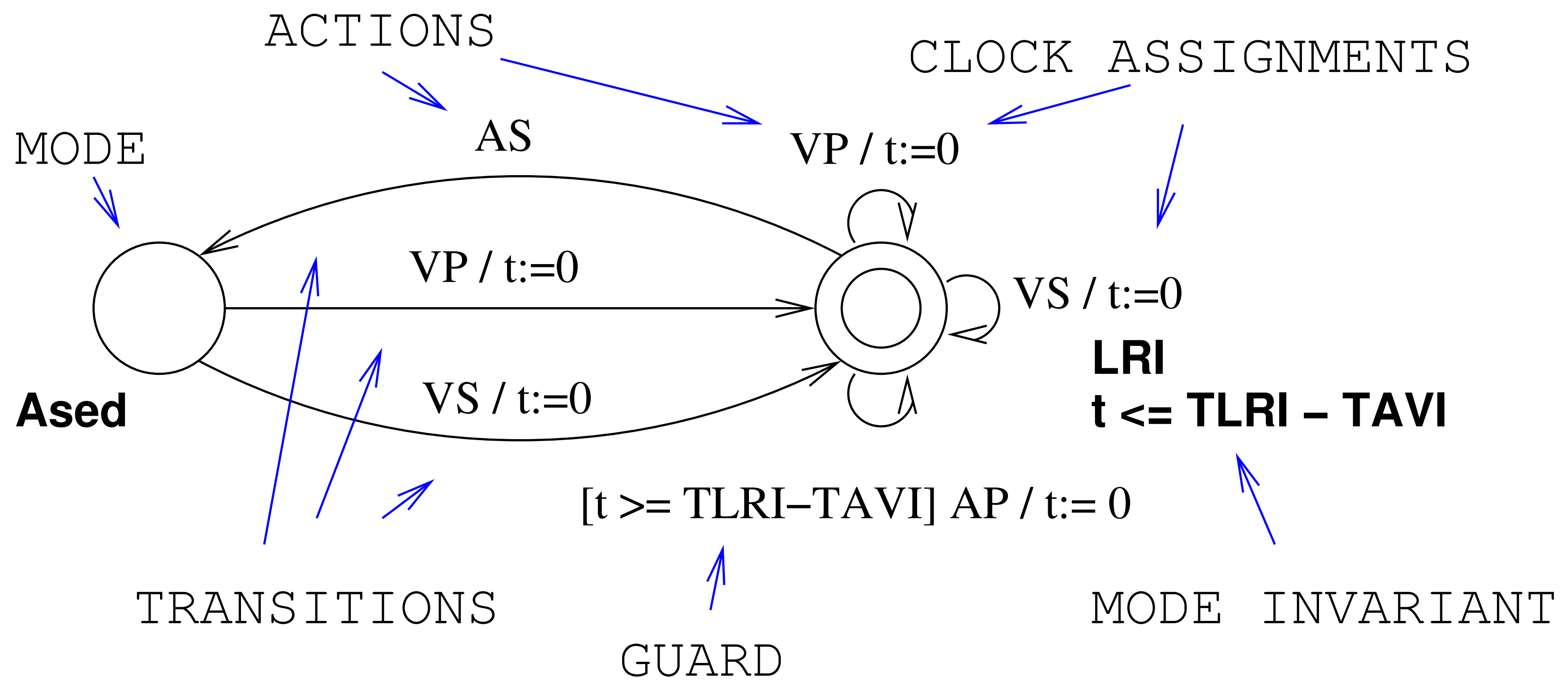
Contribution

Tools and methods to compose models of different type into an integrated model of the heart-pacemaker system.



Methods: Timed Automata

Timed automata (TA) are used to model and verify control systems, such as ICP's.



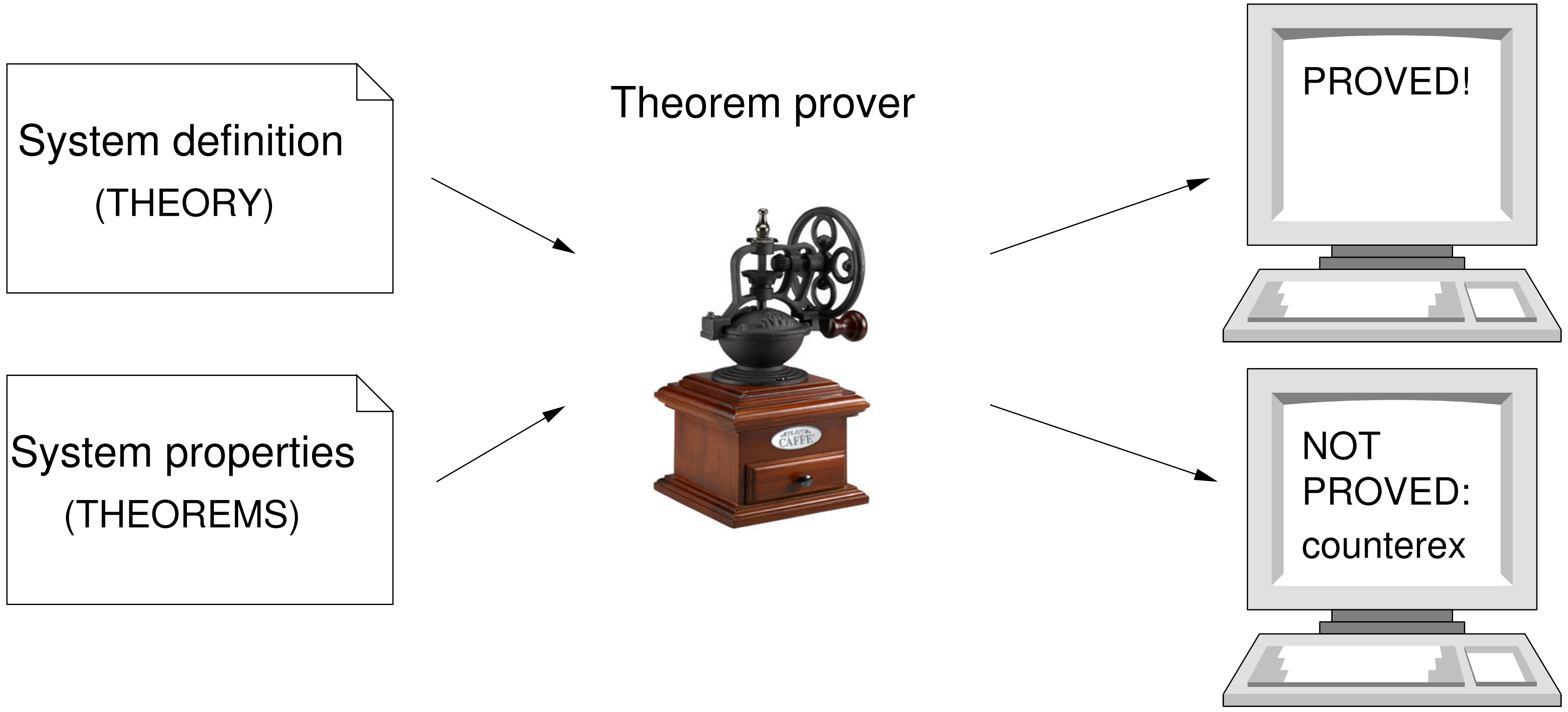
AS: atrial sense	AP: atrial pacing
VS: ventricular sense	VP: ventricular pacing
TLRI: lower rate interval	TAVI: atrio-ventricular interval

[adapted from (Jiang et al., 2012b)]

Methods: Logic Specifications

Logic specifications model a system by stating its properties in a formal language.

Logic specifications are used for the formal verification of systems, using **automatic theorem proving**.



Methods: The PVS Environment

The *Prototype Verification System* is an interactive theorem prover developed at SRI International by S. Owre, N. Shankar, J. Rushby, and others.

PVS has a rich **specification language** to define theories.

PVS has many powerful **inference rules** to prove theorems **interactively**.

- A user submits a theorem, then chooses inference rules until the proof ends successfully, or gets stuck.
- Often a single step is sufficient.

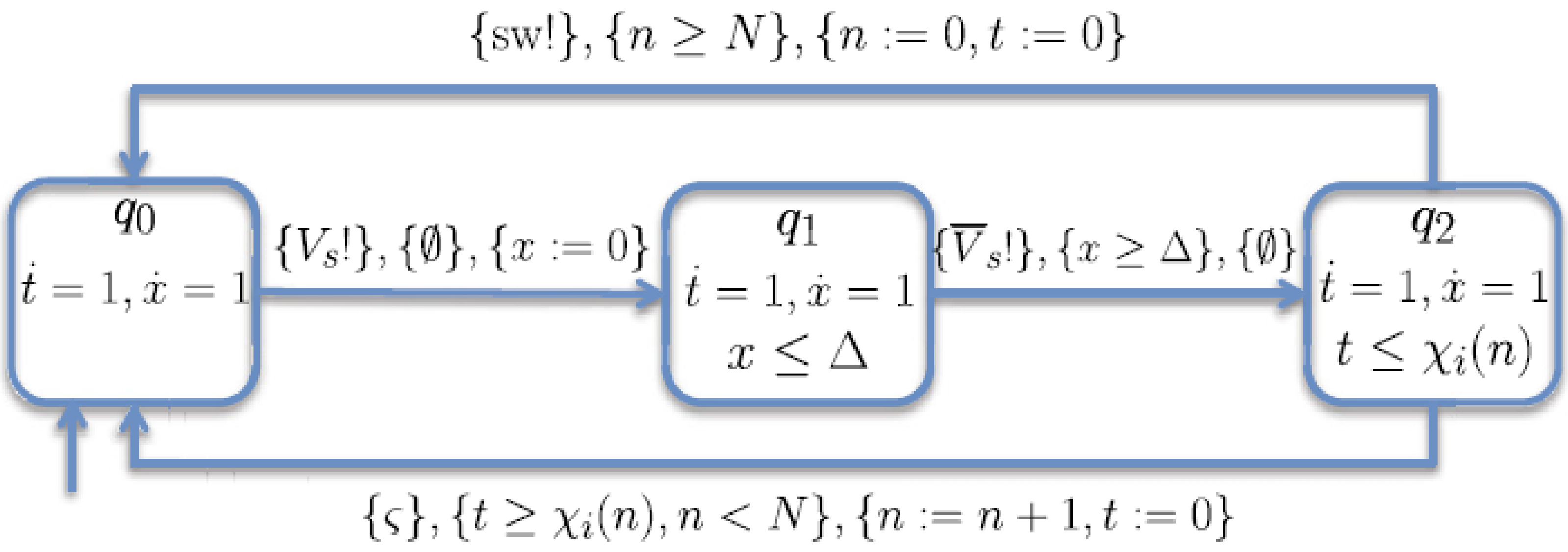
PVS has a **simulation extension** (*PVSio*).

Methods: MathWorks Models

MathWorks® Simulink is a widely used tool to model complex systems.

Heart models have been developed in Simulink with the *hybrid automaton* (HA) paradigm.

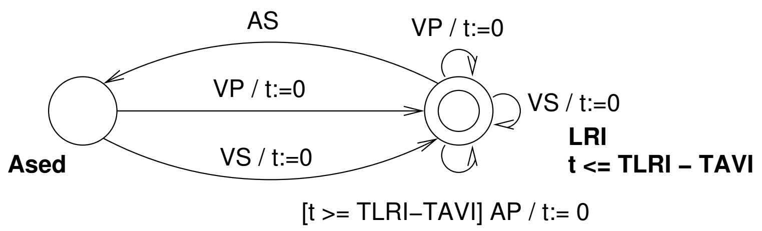
E.g., a *small* part of a heart model, the *SA node stimulation automaton* (Chen et al., 2014):


$\{\mathrm{sw}!\}, \{n \geq N\}, \{n := 0, t := 0\}$
q_0
$\dot{t} = 1, \dot{x} = 1$
$\{V_s!\}, \{\emptyset\}, \{x := 0\}$
q_1
$\dot{t} = 1, \dot{x} = 1$
$x \leq \Delta$
$\{\overline{V}_s!\}, \{x \geq \Delta\}, \{\emptyset\}$
q_2
$\dot{t} = 1, \dot{x} = 1$
$t \leq \chi_i(n)$
$\{\varsigma\}, \{t \geq \chi_i(n), n < N\}, \{n := n + 1, t := 0\}$

Integration: Pacemaker Model in PVS

A procedure has been defined to represent TAs in PVS.

E.g., in the ICP subsystem model from (Jiang et al., 2012b):



```
Mode: TYPE = { LRI, Ased }
```

MODES

A state is a pair (time, mode)

```
state: TYPE = [# time: real, mode: Mode #]
```

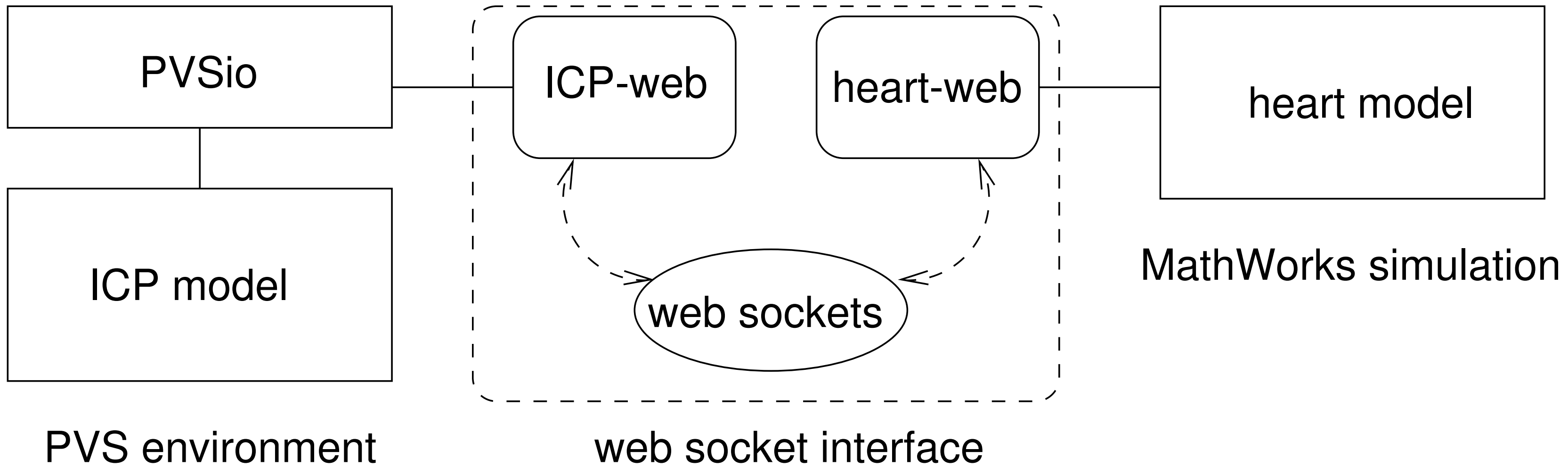
GUARD of the AP transition from mode LRI

```
per_APout(st: state): boolean =
    mode(st) = LRI AND time(st) >= TLRI-TAVI
```

TRANSITION: if state st satisfies the guard, reset clock t and remain in mode LRI

```
APout(st: (per_APout)): state =
    (# time := 0, mode := LRI #)
```

Integration: Architecture



The ICP and heart model communicate through two **web services** (programs that exchange messages with the protocols of the WWW).

The two models can be executed on the same computer, or be **distributed** on different machines.

Results

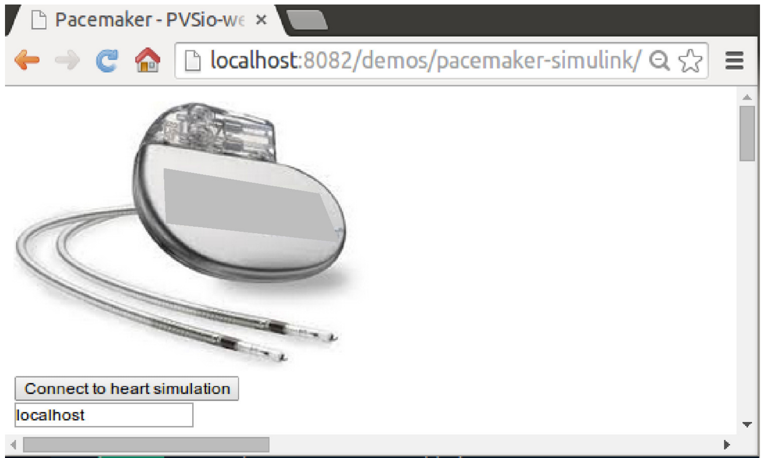
A TA-based model of a commercial ICP (Jiang et al., 2012b) was translated into the PVS language.

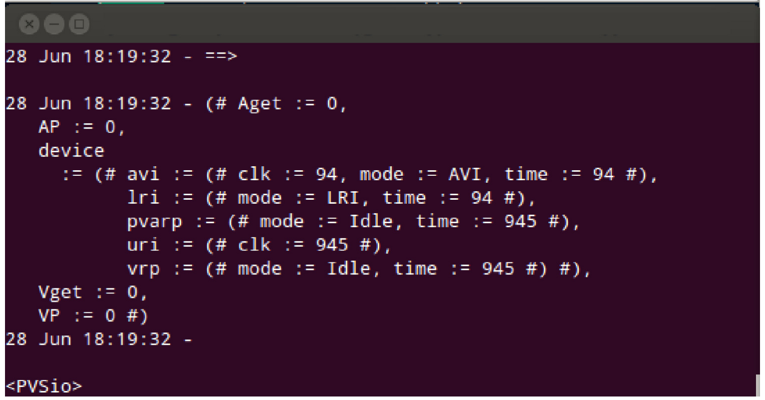
The PVS model was interfaced to a web service module with the PVSio extension.

A HA-based model of the heart, implemented in Simulink (Chen et al., 2014), was interfaced to a web service module.

Simulations were executed in the resulting integrated environment (next slide):

Results

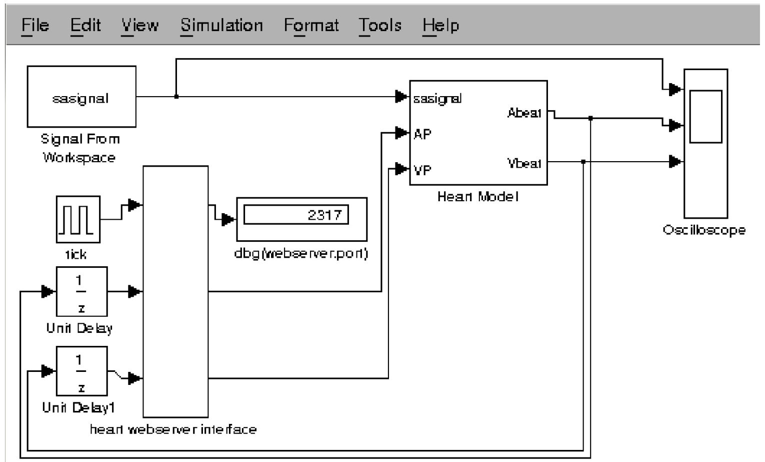

Pacemaker - PVSio-we
localhost:8082/demos/pacemaker-simulink/
Connect to heart simulation
localhost

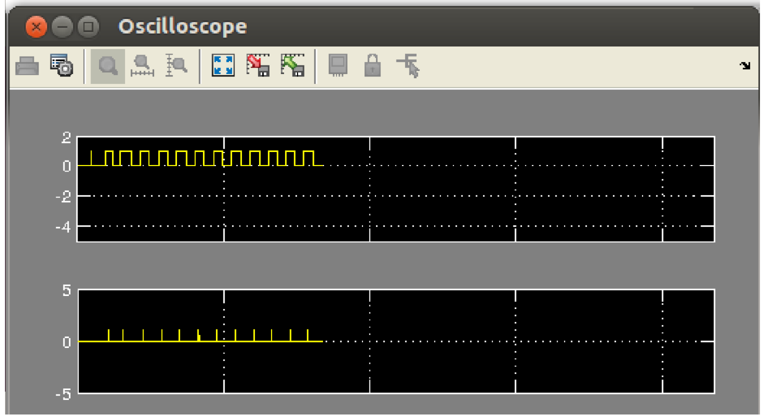


```
28 Jun 18:19:32 - ==>

28 Jun 18:19:32 - (# Aget := 0,
   AP := 0,
   device
     := (# avi := (# clk := 94, mode := AVI, time := 94 #),
           lri := (# mode := LRI, time := 94 #),
           pvarp := (# mode := Idle, time := 945 #),
           uri := (# clk := 945 #),
           vrp := (# mode := Idle, time := 945 #) #),
   Vget := 0,
   VP := 0 #)
28 Jun 18:19:32 -

<PVSio>
```





Formal Verification

Formal verification is an important complement to simulation.

E.g., suppose we want to verify that the previously shown ICP system satisfies this property:

It is always the case that the subsystem is in mode LRI and its clock is reset when transition AP is executed.

```
lri_ap: LEMMA
  FORALL (s0, s1: State):
    per_APout(lri(s0)) AND s1 = APout(s0) IMPLIES
    mode(lri(s1)) = LRI AND time(lri(s1)) = 0
```

A single application of the *grind* rule is sufficient.

```
Rule? (grind)
...
Q.E.D.
Run time  = 0.17 secs.
```

Conclusions

A method and its supporting tools has been developed to integrate ICP and heart models built on different modeling paradigms.

This enables developers of ICPs to model each component with the most appropriate tools and languages.

The resulting simulation environment can be executed on a laptop or on a distributed system. This affords more computing power when needed, and the convenience of switching easily between different models.

Using the PVS environment for the ICP model makes it possible to couple simulation to formal verification.

The method has been used to integrate ICP and heart models described in the literature, developed independently.

Thank you

Grazie