Model Checking Report

TianyuanWu 63305667

I. Pacemaker Model

According to the specification of DDD pacemaker, a timed-automaton based pacemaker model was created in UPPAAL. The structure of the model is shown in Fig.1.

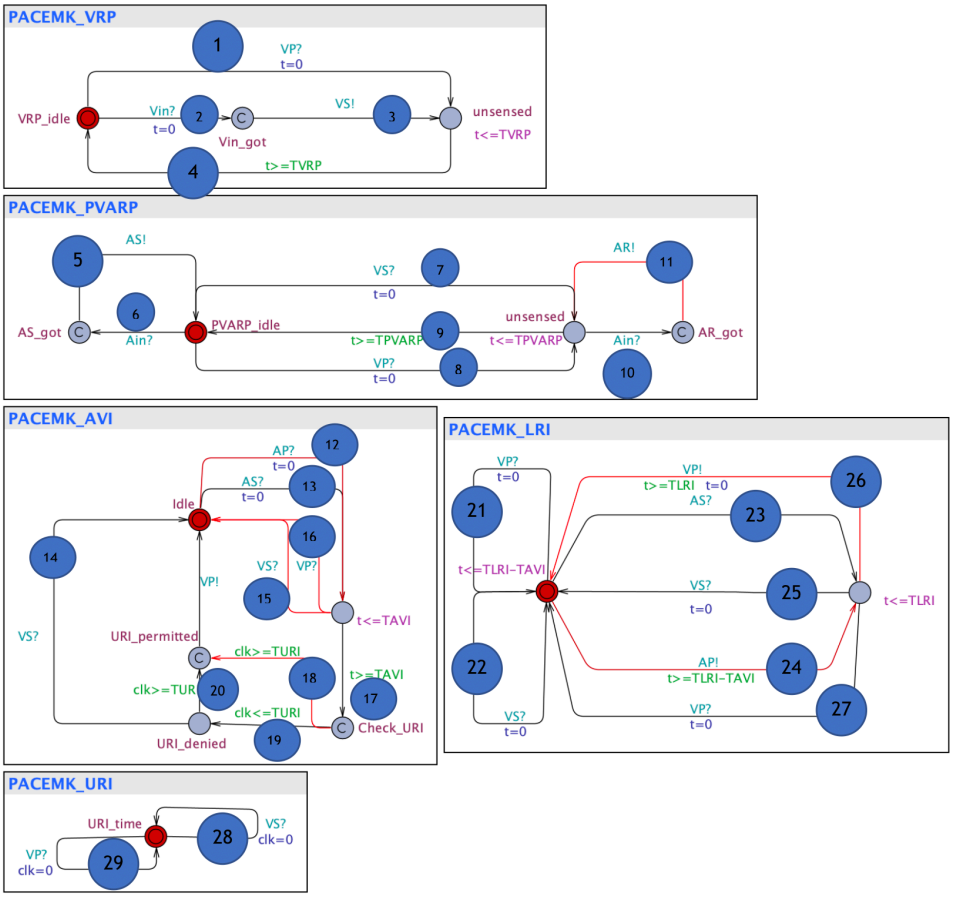

Fig.1. Pacemaker model

This model contains 5 subsystems, named `PACEMK_VRP`, `PACEMK_PVARP`, `PACEMK_AVI`, `PACEMK_LRI` and `PACEMK_URI`. In this design, `PACEMK_VRP` is to ensure to not pace during ventricular refectory period, `PACEMK_PVARP` is to ensure there is an interval between a ventricular event and an atrial event (i.e. not make Atrial and ventricular contractions are too close to each other). `PACEMK_AVI` is to ensure that after an atrial event, if no ventricular event was detected during AVI, then give a VP to ventricular. `PACEMK_LRI` is to maintain heart rate no less than minimum ventricular rate, and `PACEMK_URI` is to maintain heart rate no larger than maximum ventricular rate.

II. Heart Model

A heart model should be able to generate Ain and Vin signal at any time we want. So we generate different abstract level of heart (shown in Fig.2.).

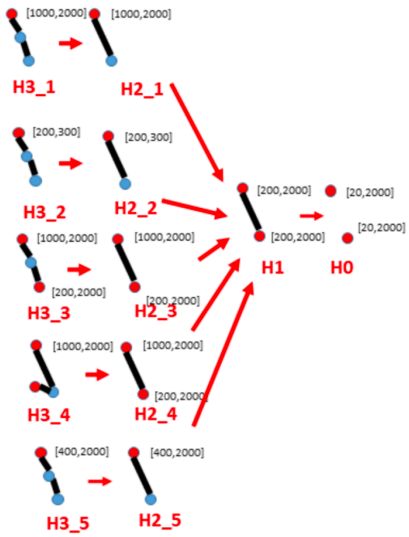

Fig.2. Different Abstraction Levels of Heart Level

The following heart model (Fig. 2) is the top abstraction level of heart model, and is paired with the pacemaker model shown at first page.

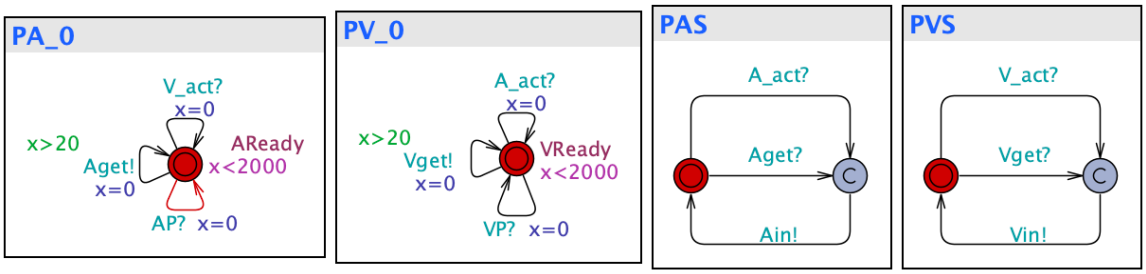

Fig.3. Heart Model

As shown, It can generate Ain and Vin signal, each signal has possible interval from 20 to 2000. So it can generate signals as expected (i.e. all possible inputs of pacemaker are able to be covered).

III. Model Checking With TCTL Query

According to the specifications of DDD pacemaker, 4 TCTL queries can be generated.

Query 1. (According to Spec.1. (No deadlocks)):

```
A[] (not deadlock)
```

This query can be easily checked by cover all transitions and check if the pacemaker get "stuck" at some state.

Query 2. (According to Spec.2. (Maintain minimum heart rate)):

```
A[] (PLRL.two_v imply PLRL.t<=TLRI)
```

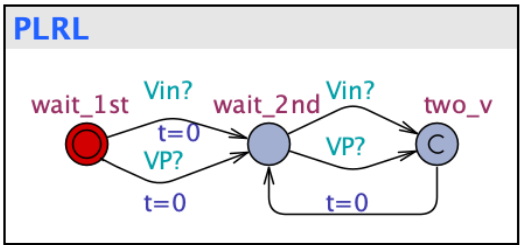

Fig.3. Monitor for query 2

To check this property, we need an extra monitor (shown in Fig. 3.)

Query 3. (According to Spec.3. (Maintain maximum heart rate)):

```
A[] (PURL.interval imply PURL.t>=TURI)
```

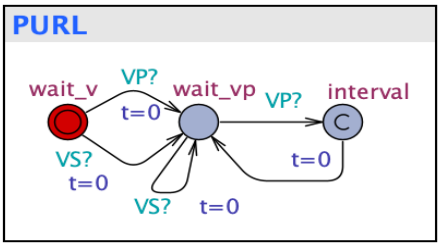

Fig.4. Monitor for query 3

To check this property, we need an extra monitor (shown in Fig. 4.)

Query 4. (According to Spec.4. (No persistent fast ventricular events)):

```
A[] (not PPersist.err)
```

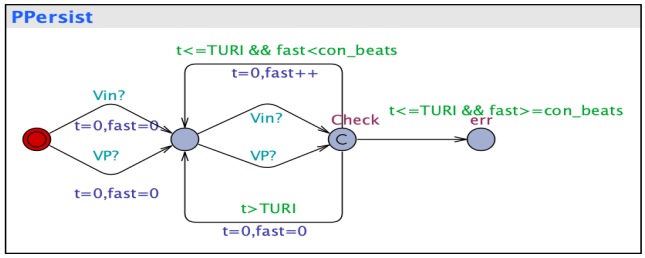

Fig.5. Monitor for query 4

To check this property, we need an extra monitor (shown in Fig. 5.)

IV. Model Checking Result

All above 4 queries were checked by UPPAAL checker, the result is shown in Fig.6. In this figure, green dot indicates the corresponding property is satisfied, while red dot indicates the property is not satisfied. So, we can easily observe that Query 1, 2, 3 is satisfied, but Query 4 not satisfied.

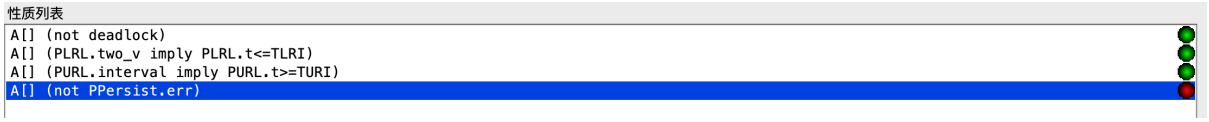

Fig.6. Model Checking Result

According to specification and risk management report, the reason of unsatisfactory of Query 4 is caused by ELT and ATR, which are not fatal to patients. So in this model, this 2 problems were not handled.

V. Counter Examples

(1) ATR caused persistent fast ventricular events

When heart is under ATR condition, pacemaker may cause persistent fast ventricular events, the visualization of this situation is shown in Fig.7.

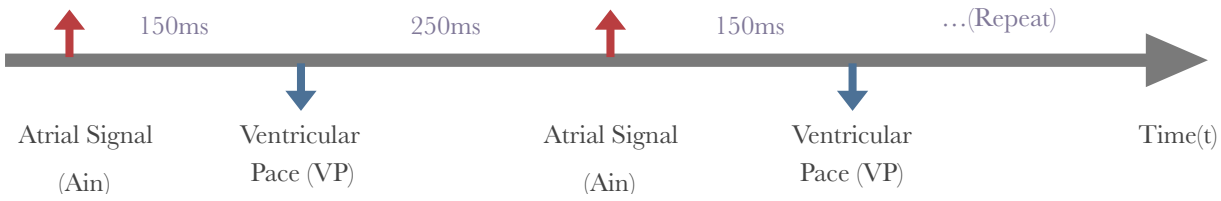

Fig.7. ATR Counter Example Visualization

(2) ELT caused persistent fast ventricular events

When heart is under ELT condition, pacemaker may cause persistent fast ventricular events, the visualization of this situation is shown in Fig.8.

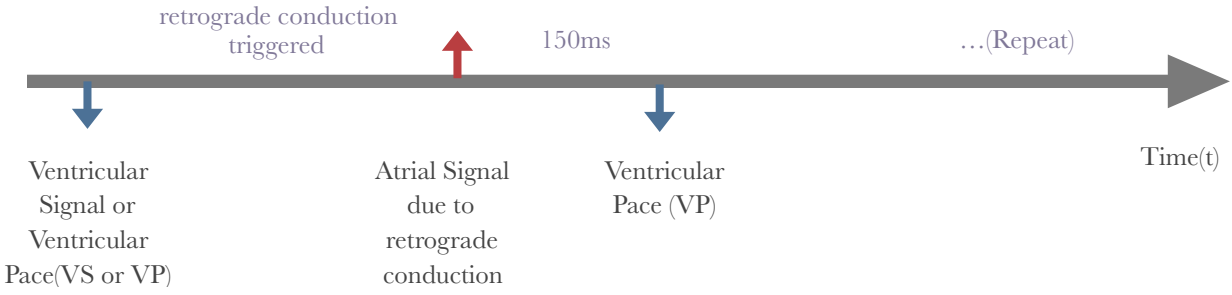

Fig.8. ELT Counter Example Visualization