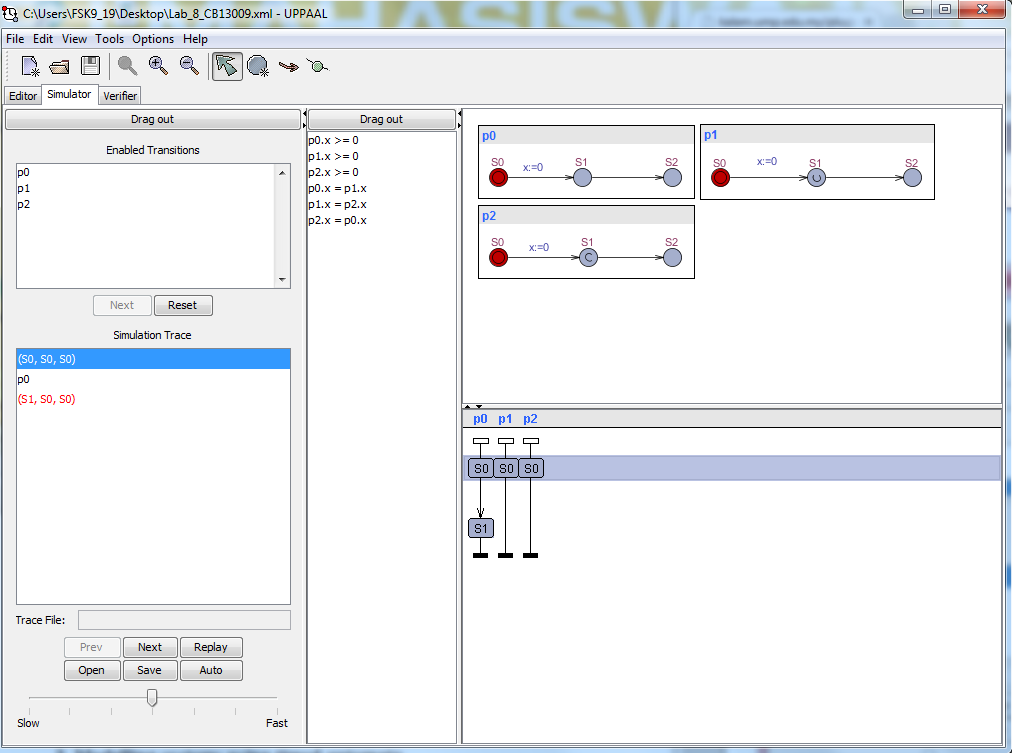
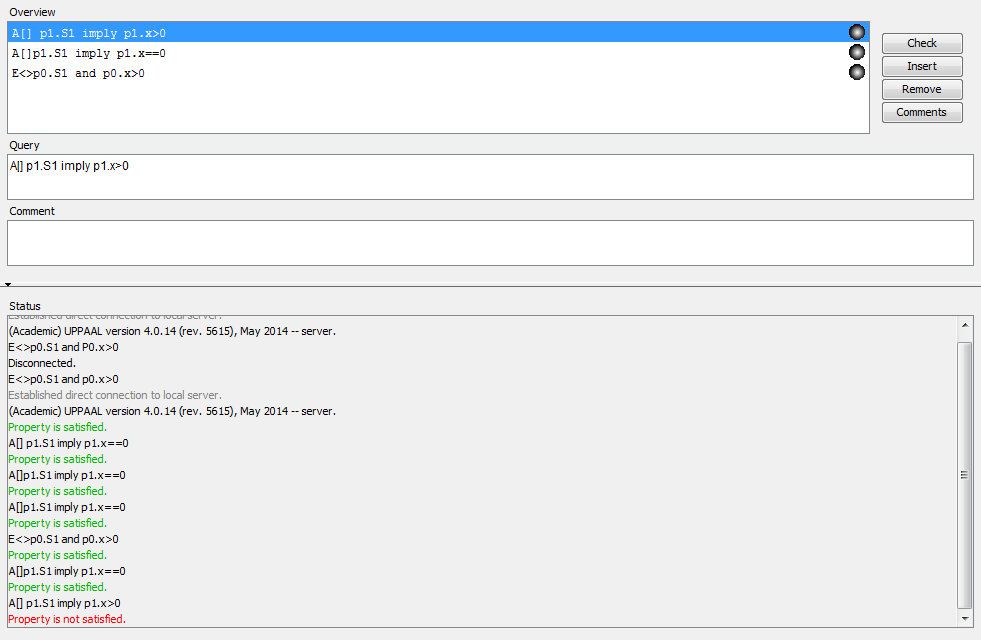
**1.There are 3 different type of location in UPPAAL**

* Normal locations
* Urgent locations
* Committed locations





The location marked “U” is urgent and the one marked “C” is committed. From the study is it possible to wait in S1 of P0 (Normal location)

* E <> P0.S1 and P0.X>0 is TRUE

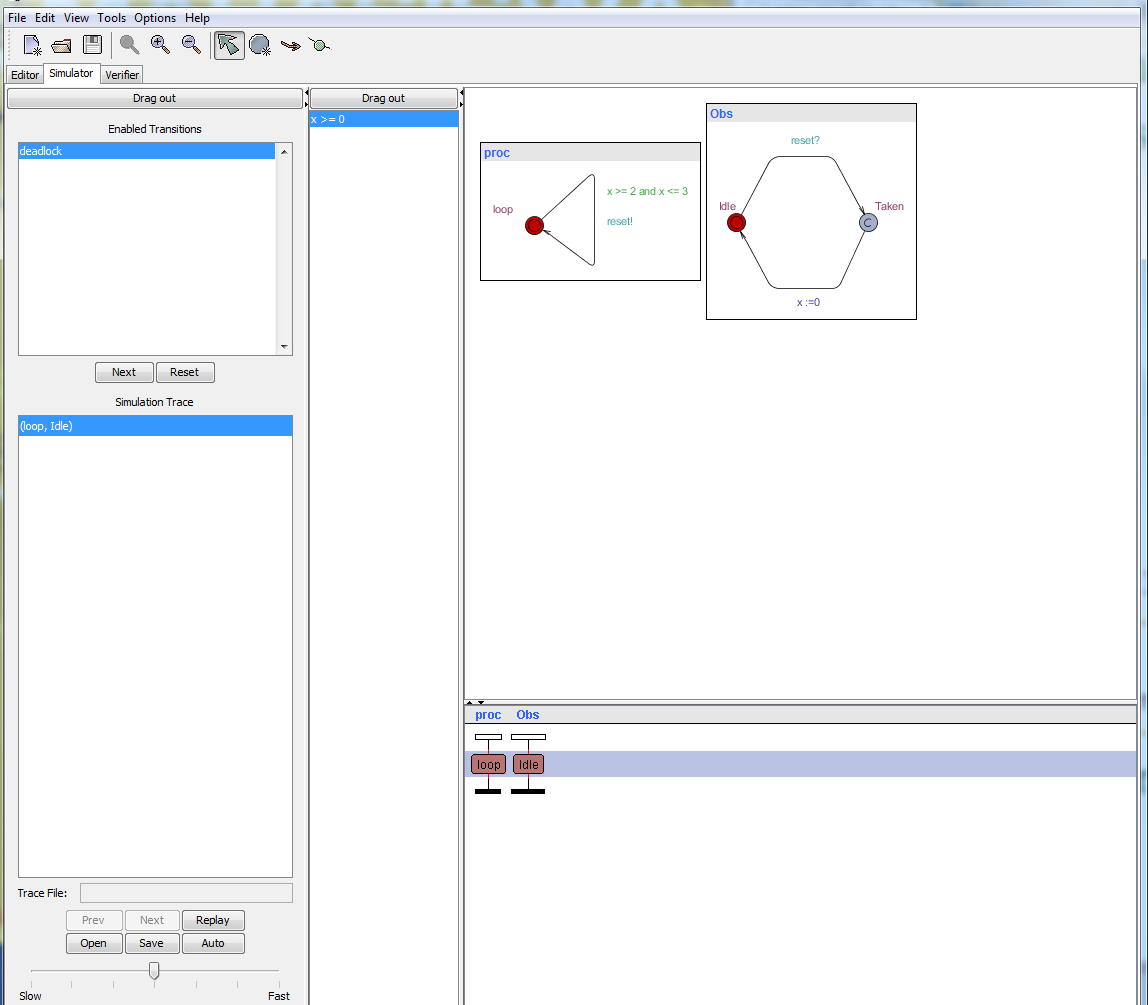
And it is not possible to wait in S1 of P1 (urgent location)

* A[] P1.S1 imply P1.x==0 TRUE
* A[] P1.S1 imply P1.x>0 FALSE

Time may not pass in an urgent state, but interleaving’s with normal states are possible .Thus, urgent locations are “less strict” than committed.

**2. Modelling systems using timed automata**

The figure below shows the model of timed automata and the declaration of the system.



The figure below show the declaration of the system.

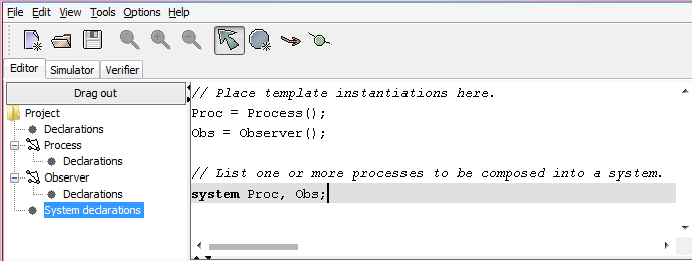
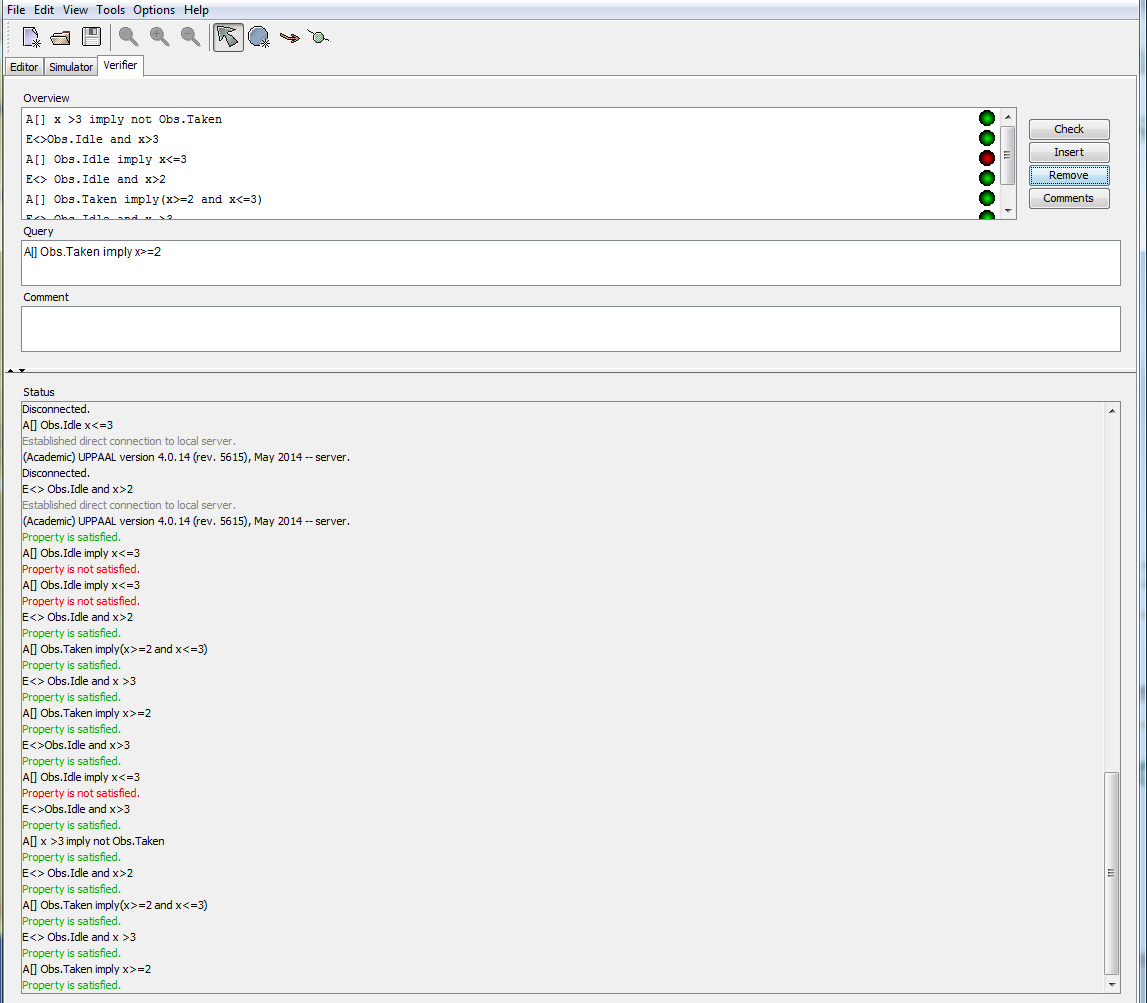


Figure below to test the properties to exhibit the behaviour



From the figure above we can conclude that

* A[ ] Obs.Taken imply x>=2(for all states,being in the location Obs.Taken implies that x>=2)
* E<> Obs.idle and x >3 this is for waiting period if we try like x > 3000 and we will get the same result

The invariant is a progress condition the system is not allowed to stay in the state more than 3 time units, so the transition has to be taken and the clock reset such as:

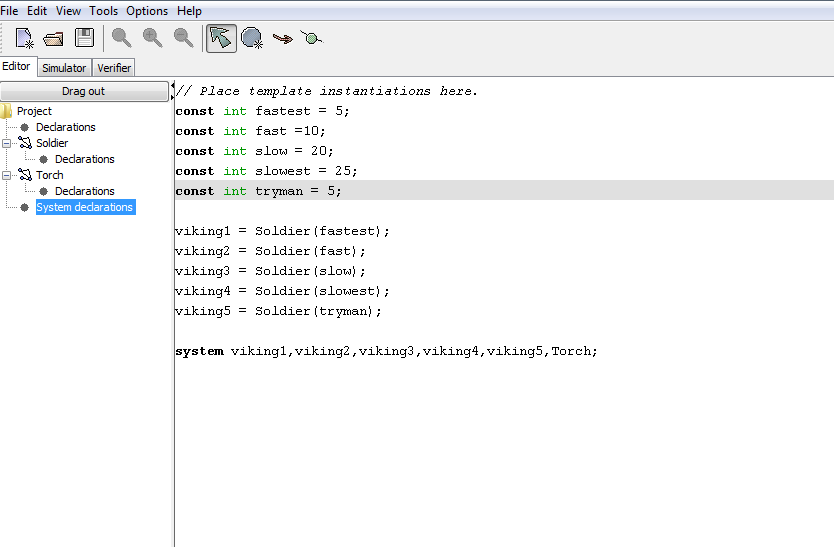
* A [] Obs.Taken imply (x >=2 and x <=3) to show that the transition is taken when x is in the interval (2 and 3)
* E<> Obs.Idle and x > 2 it is possible to take the transition with x in the interval (2 and 3)
* A [] Obs.idle imply x <=3 to show that the upper bound is respected.
* The former property E <> Obs.idle and x>3 no longer holds.

The interesting part is when we remove the invariant that hold the value of x <= 3 .We thought the output of the system is the same but not. The system has no progress condition anymore and add the guard is put x<=3.Moreover the system show a deadlock the system may be stuck if it does not take the transition after 3 time units. The deadlock property is:

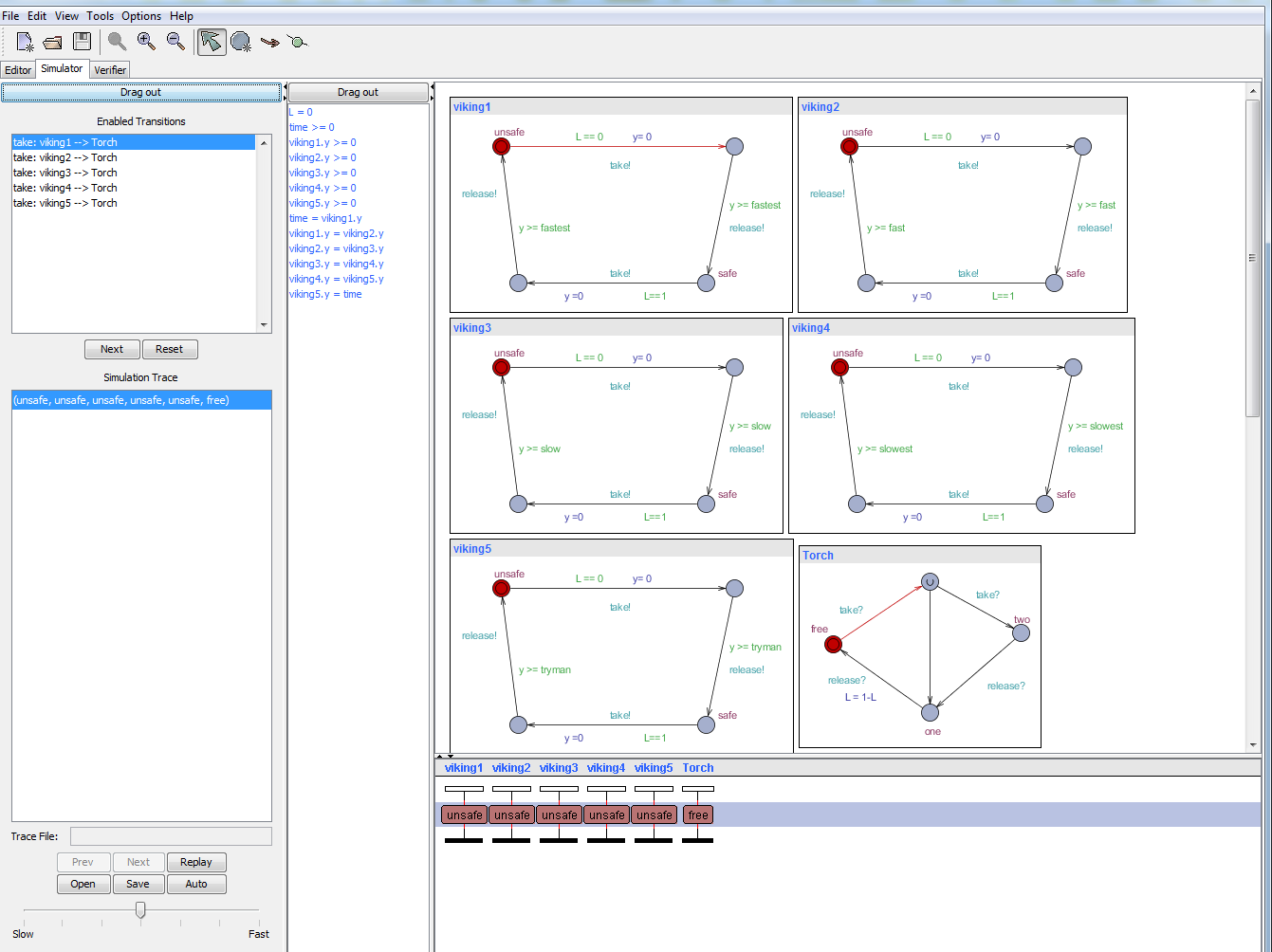
* A[] x>3 imply not Obs.Taken show green in colour(the property is satisfied)
* E<> Obs.idle and x> 3 show red in colour(the property is not satisfied)

**3. Extra Mark for Vikings question**

From the figure below we try the variable “Tryman” is equal to integer and hold the number 5 seconds.

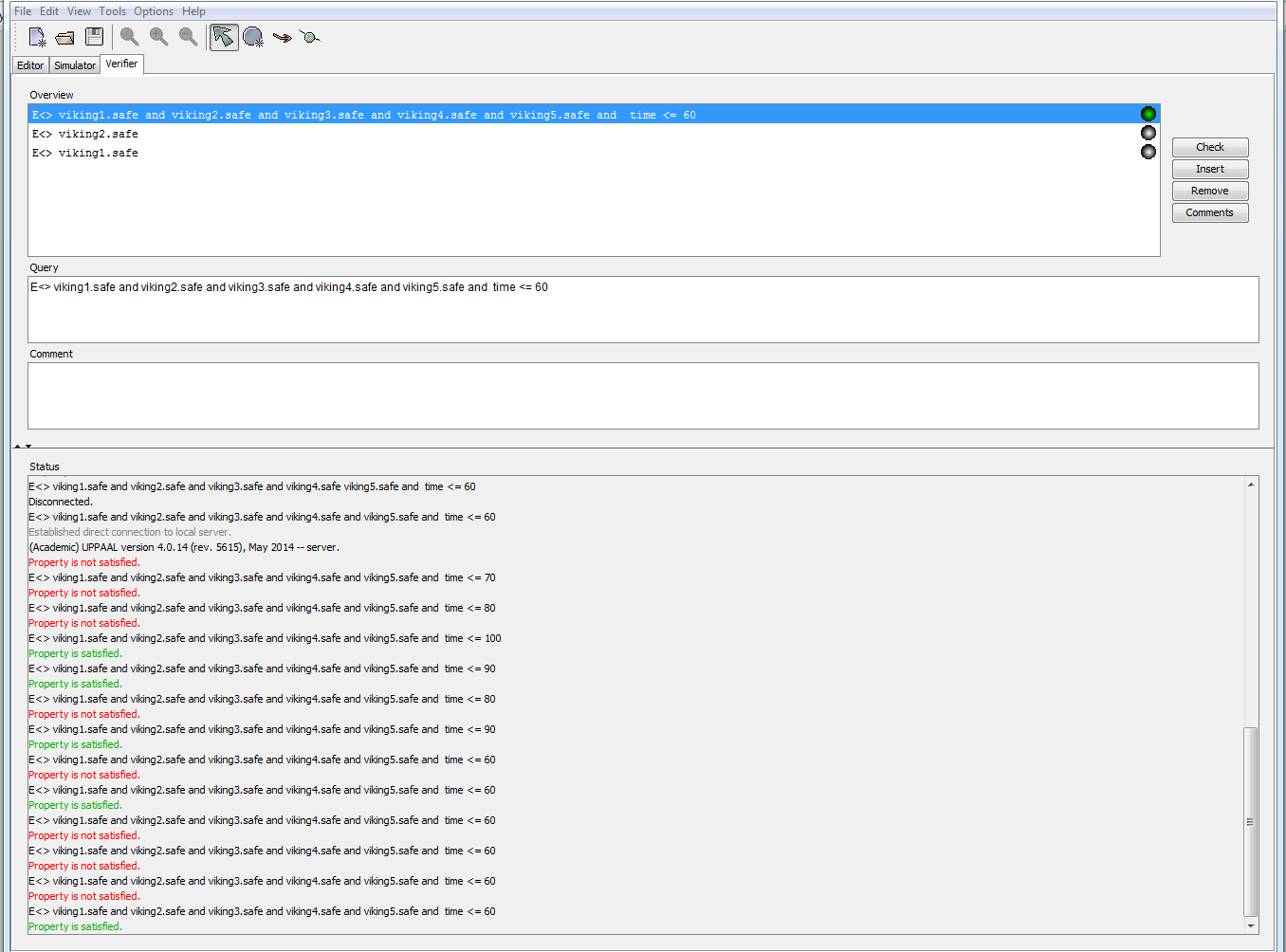
****

The simulator show the 5 Vikings models.



We try all the properties to test the behaviour such as in figure below:

* A [] not deadlock show green colour
* E <> Viking1.safe show green colour
* E <> Viking2.safe show green colour
* E <> Viking3.safe show green colour
* E <> Viking4.safe show green colour
* A[] not (Vikin4.safe and time <slowest) show green colour
* E<>Viking1.safe and Viking2.safe and viking3.safe and viking4.safe and viking4.safe and time<=60 show green colour.



From the properties above, I conclude that the viking1, viking2, Vikings3, viking4 and Vikings 5 can cross the bridge in 60 minutes. The time to cross the bridge by this viking5 is <= 5.