Algorithm for updating concurrent model from diff(Env,Env')

```
//Brief description of each variables s (ConcurrentStates()): Set of states in concurrent model
e,m,fs,ss:State which has transitions to connected other states
t:Transition which has 'To state' and 'From state' which is connected by this
subTransition: diff(Env,Env') as Transition
```

```
s=new ConcurrentStates()
//make initial concurrent model
$initialize(Env, Moni_{req})$
  e=$Env$.getInitState ()
  m=$Moni_{req}$.getInitState ()
  s.add(makeConcurrentState(e,m))
  $transition(e, m, s)$
}
$transition(e, m, s)${
  while( e.hasNextTransition() ){
    t=e.getNextTransition()
    $compose(e.getNextStateBy(t), t, m, s)$
  }
}
$compose(e, t, m, s)${
    if(m.existsTransition(t)){
      m=m.getNextStateBy(t)
      if(!s.alreadyExists(e,m)){
        s.add(makeConcurrentState(e,m))
        $transition(e, m, s)$
      }
    }else{
      if(!s.alreadyExists(e,m)){
        s.add(makeConcurrentState(e,m))
        $transition(e, m, s)$
      }
    }
}

//Updating concurrent model

concurrentModelUpdate $(Env, Env', Moni_{req})${
  subTransition=getDiff(Env,Env')
  fs= subTransition.getFromState()
  list=s.getStateIncluding(fs)
  while(list.hasNextState()){
    ss=list.getNextState()
    $compose(fs, subTransition, ss.getMoni_{req}State(), s)$
  }
}
```