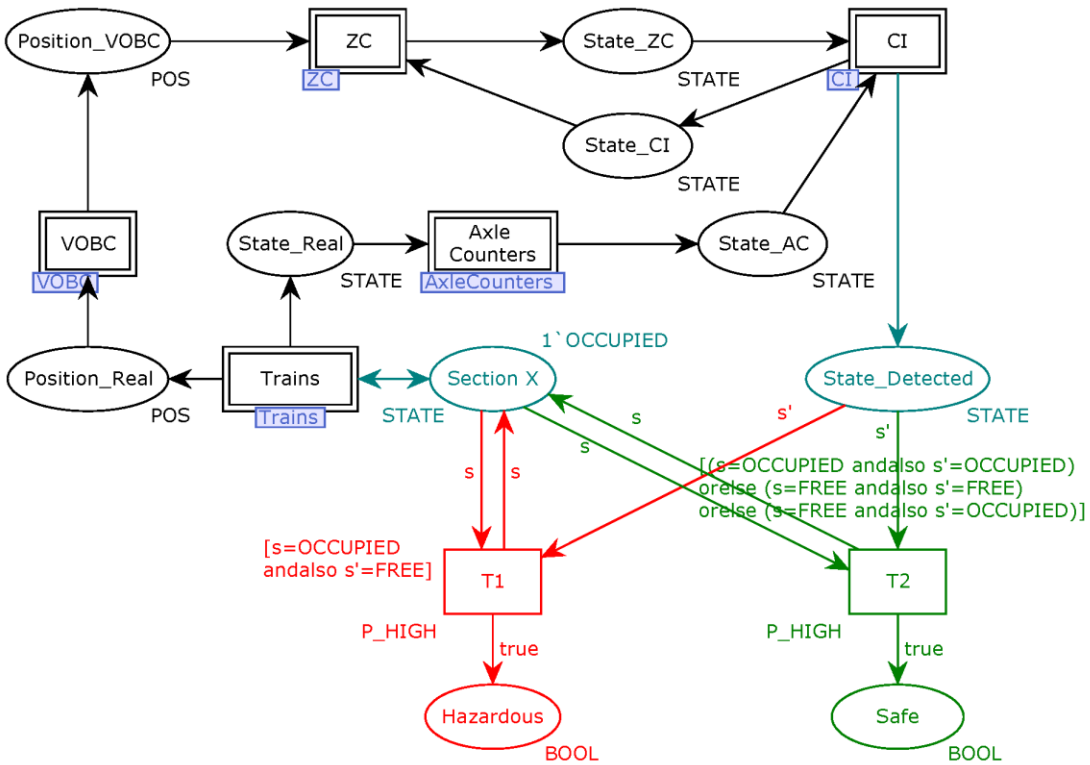
1. Declarations

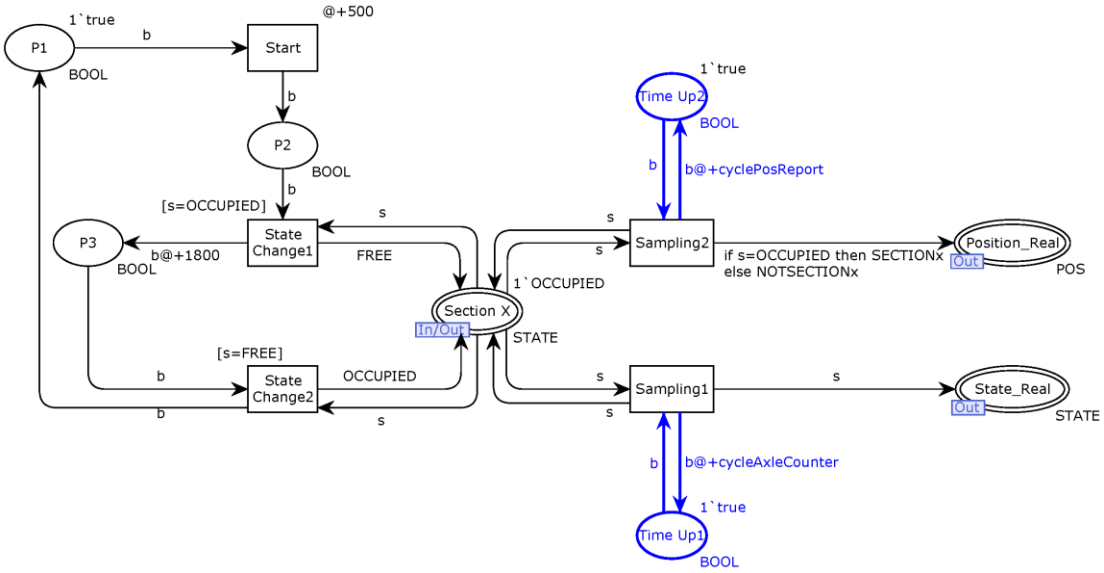
```
colset UNIT = unit;
colset BOOL = bool timed;
colset INT = int timed;
colset STATE = with OCCUPIED |    FREE timed;
colset POS = with SECTIONx | NOTSECTIONx timed;
var u, u' : UNIT;
var n, m, delay, k: INT;
var s, s': STATE;
var p, p': POS;
var b: BOOL;
fun RelayDelay () = discrete (0,20);
fun CI_ZC_Delay() = discrete (0,60);
fun VOBC_ZC_Delay() = discrete (0,50);
val cycleCI = 2;
val cycleZC = 3;
val cycleAxleCounter = 1;
val cyclePosReport = 1;
```

2. Timed CPNs

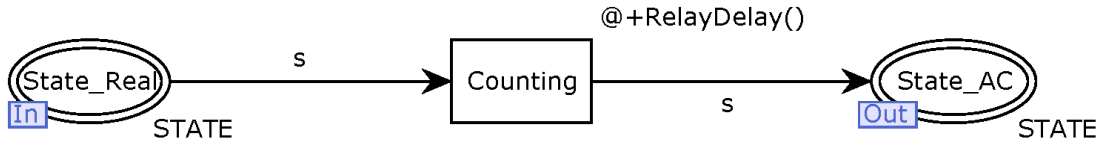
1) Top-level



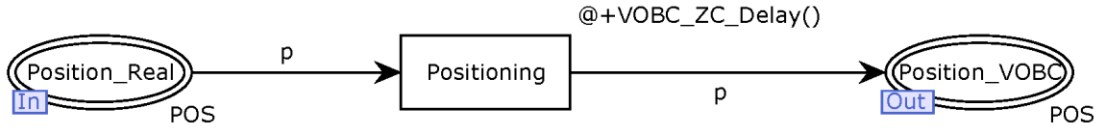
2) Module *Trains*



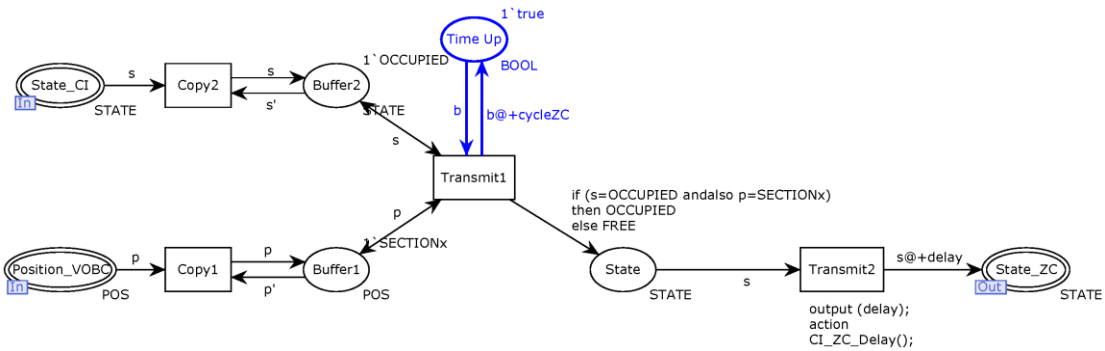
3) Module *AxleCounters*



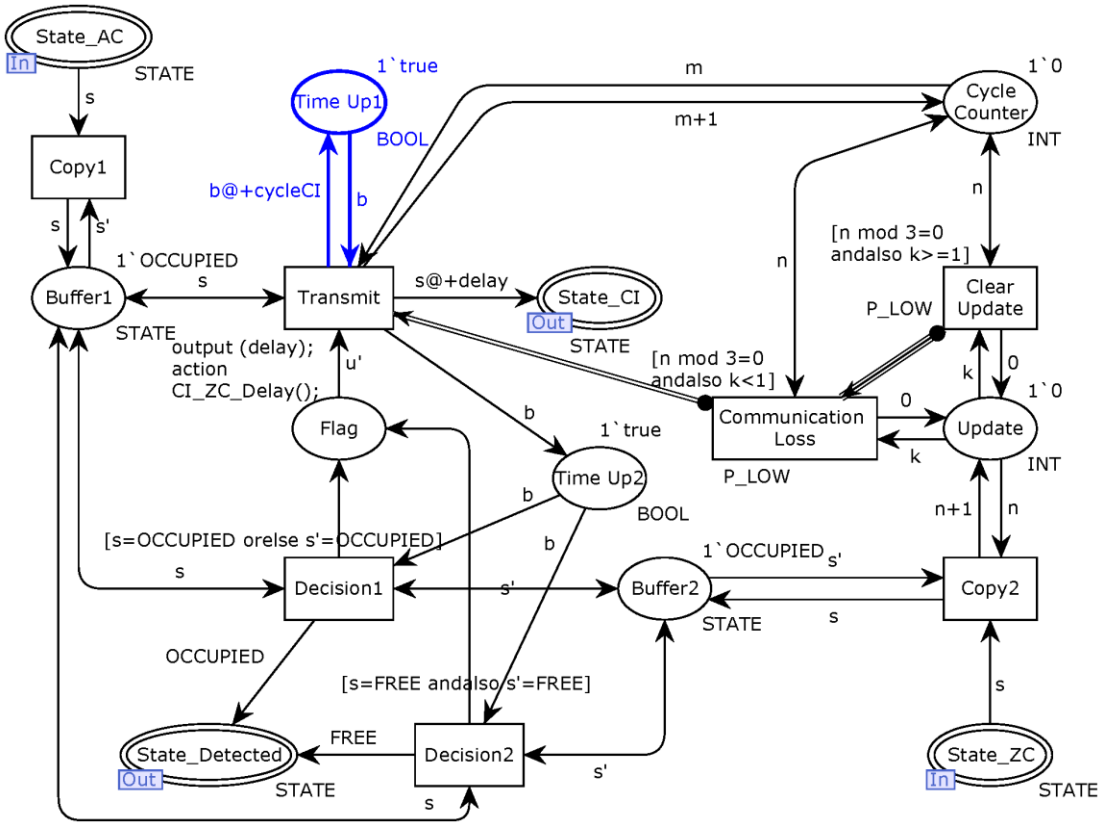
4) Module *VOBC*



5) Module *ZC*

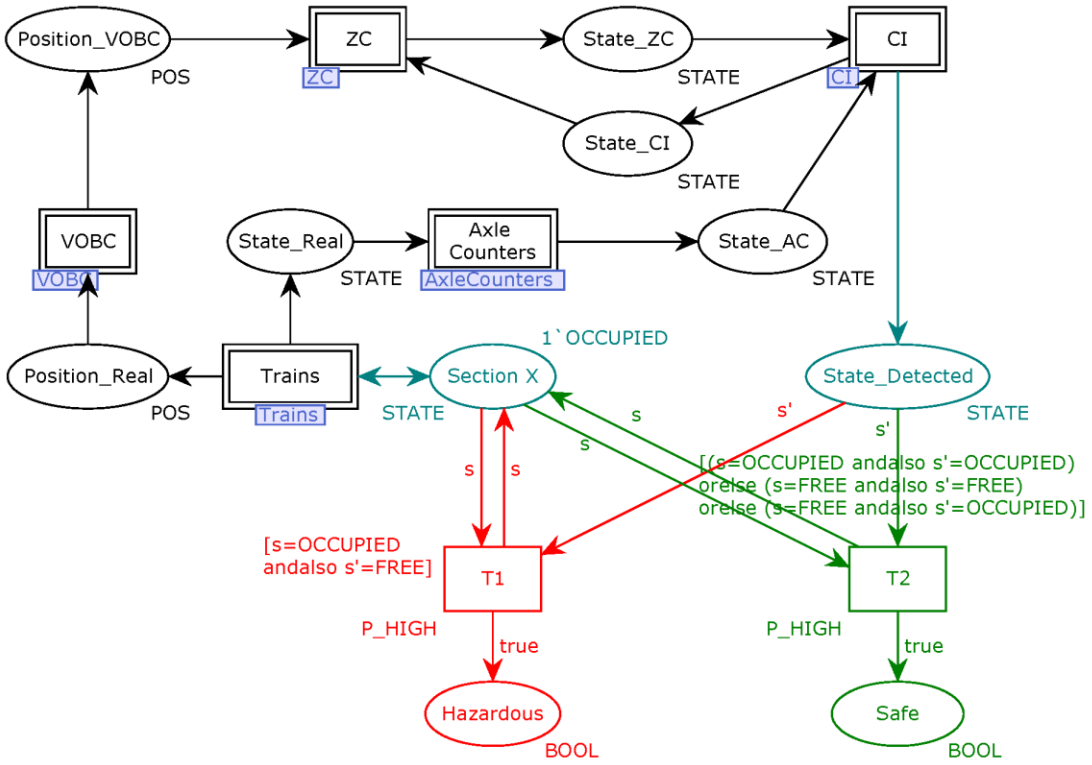


6) Module *CI*

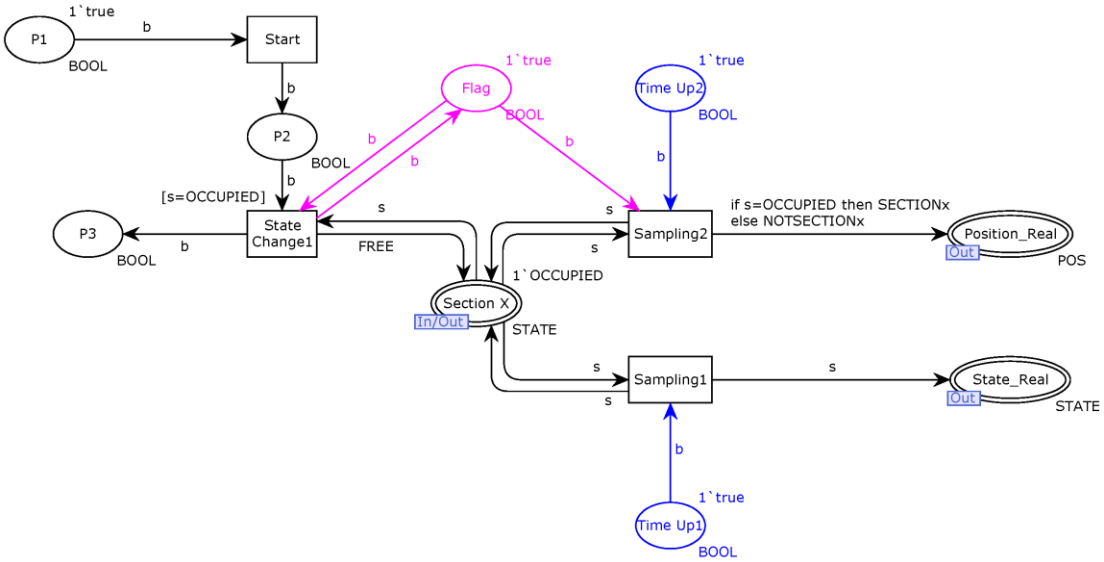

State_AC
In
STATE
s
Copy1
s
s'
1`OCCUPIED
Buffer1
STATE
s
Transmit
1`true
Time Up1
BOOL
b@+cycleCI
b
m
m+1
1`0
Cycle Counter
INT
n
[n mod 3=0 andalso k>=1]
Clear Update
P_LOW
k
0
1`0
Update
INT
n+1
n
s@+delay
State_CI
Out
STATE
output (delay); action CI_ZC_Delay();
u'
[n mod 3=0 andalso k<1]
Communication Loss
P_LOW
0
k
Flag
b
1`true
Time Up2
BOOL
b
[s=OCCUPIED orelse s'=OCCUPIED]
s
Decision1
s'
Buffer2
1`OCCUPIED
s'
STATE
s
Copy2
OCCUPIED
[s=FREE andalso s'=FREE]
s
FREE
State_Detected
Out
STATE
Decision2
s'
s
State_ZC
In
STATE

3. Untimed CPNs

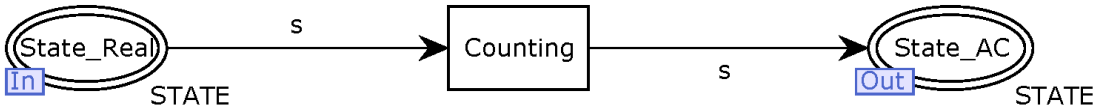
1) Top-level



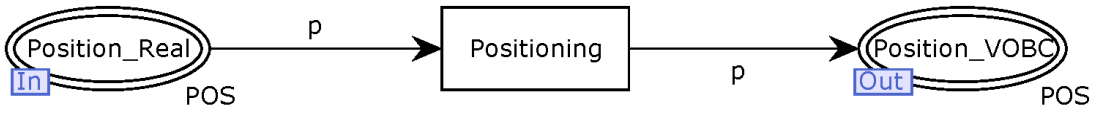
2) Module *Trains*



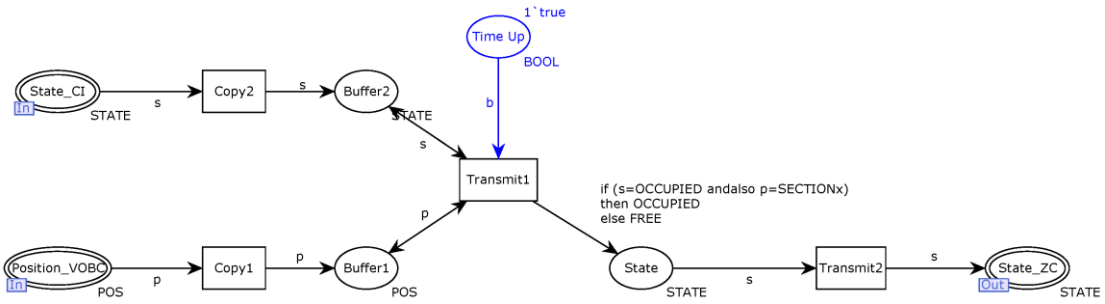
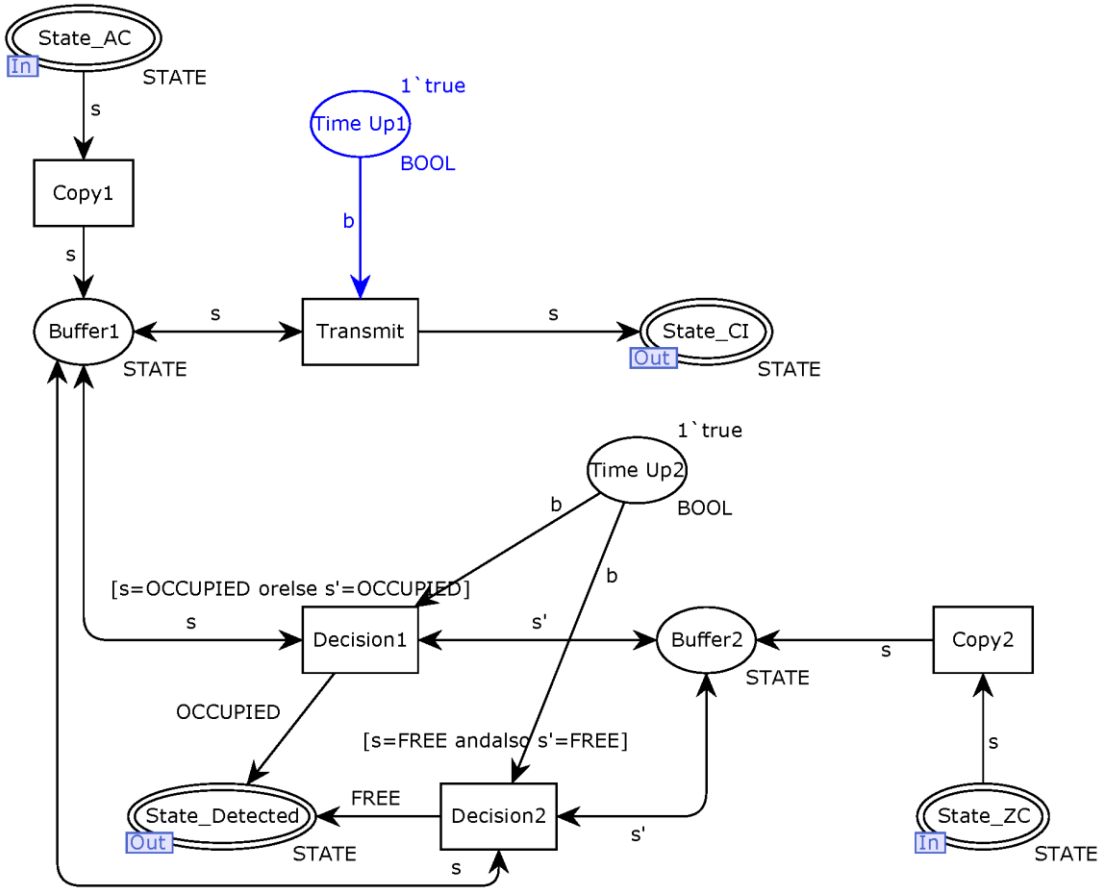
3) Module *AxleCounters*



4) Module *VOBC*



5) Module *ZC*



6) Module *CI*