Full HCPN Model Autonomous Route Management

I. DECLARATIONS

1) colset SwitchPos= with position|reverse;

2) colset orders =with Order|Convert|Lock|Occupy|Release;

3) colset Command= product SwitchPos*orders;

4) colset StateRes=with free|lock|occupy|convert|release;

5) colset ResourceSwitch = product SwitchPos*StateRes;

6) colset ResourceResult = product ResourceSwitch*orders;

7) colset Communication= with establish|disconnect;

8) colset Response = with response|unresponse;

9) colset Result = with success|fail;

10) colset Task= with register|cancel;

11) colset Procedure= product STRING*Task;

12) colset StateCom = with online|offline;

13) colset TypeRes= with track|switch;

14) colset Approach = product TypeRes*INT;

15) colset ApproachList = list Approach;

16) var TrainID,TrainID1,map,mes:STRING;

17) var sw,sw':SwitchPos;

18) var or:orders;

19) var sta:StateRes;

20) var resp,resp1:Response;

21) var resu,resu1:Result;

22) var task:Task;

23) var com:StateCom;

24) var n,num,num1,num2:INT;
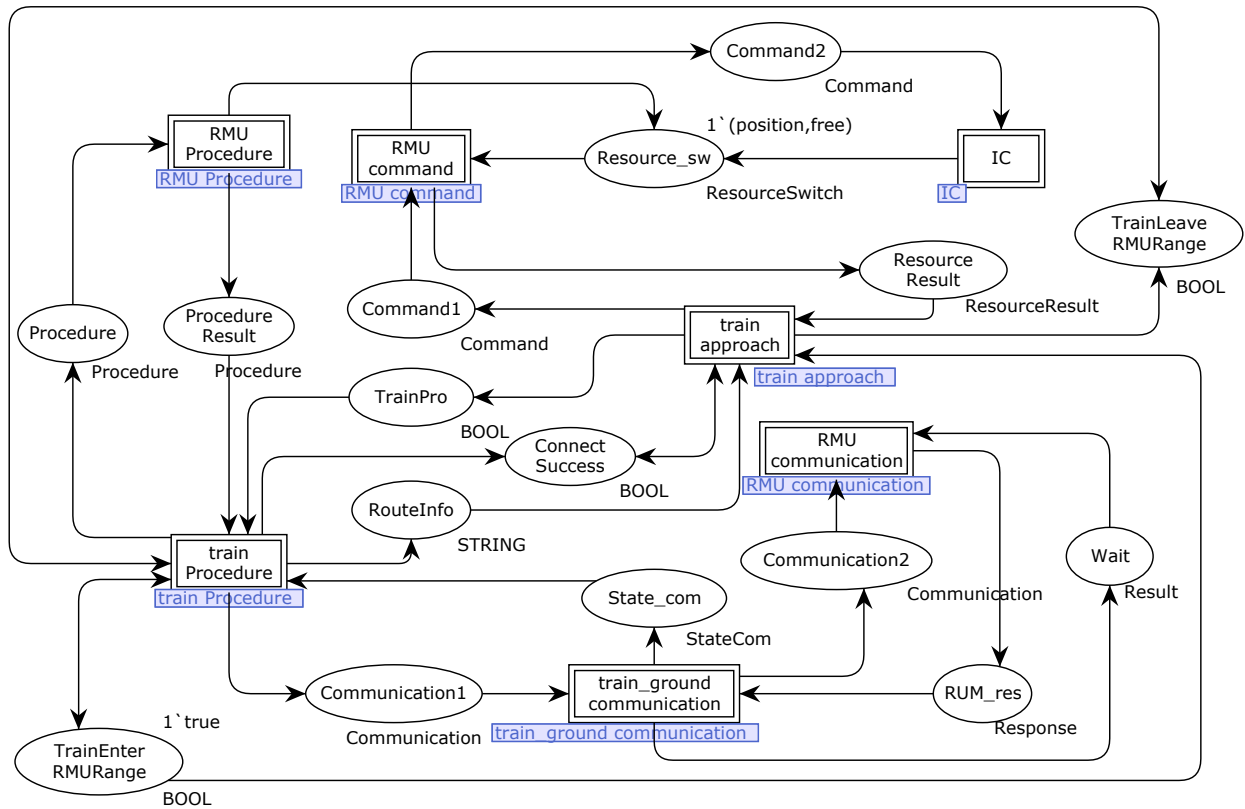
25) var ty:TypeRes;

26) var approa:ApproachList;

Command2

Command

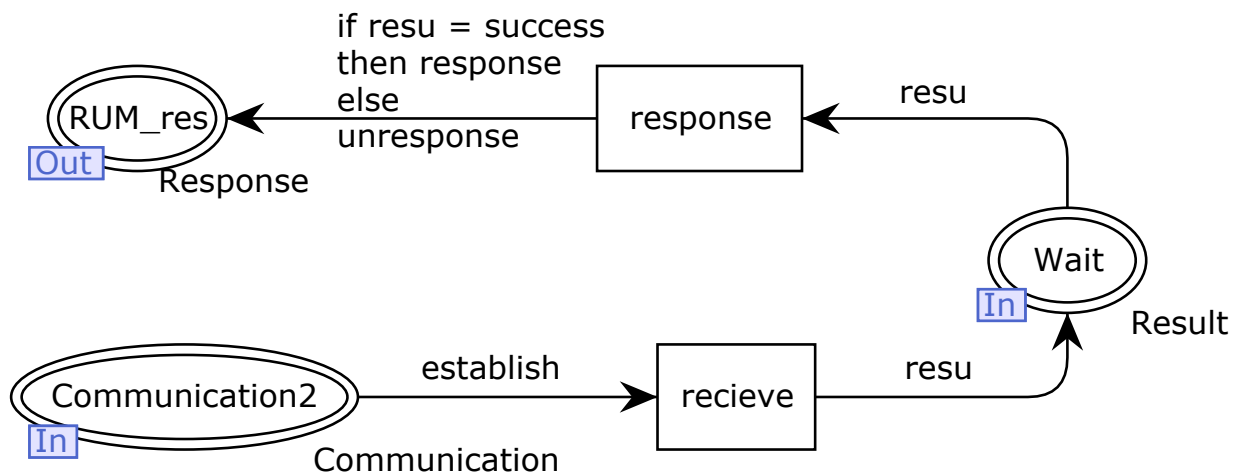1`(position,free)

RMU
Procedure
RMU Procedure

RMU
command
RMU command

Resource_sw

ResourceSwitch

IC
IC

TrainLeave
RMURange

BOOL

Procedure

Procedure
Result

Resource
Result

ResourceResult

Procedure

Procedure

Command1

Command

train
approach
train approach

TrainPro

BOOL

Connect
Success

BOOL

RMU
communication
RMU communication

RouteInfo

STRING

train
Procedure
train Procedure

State_com

StateCom

Communication2

Communication

Wait

Result

Communication1

Communication

train_ground
communication
train_ground communication

RUM_res

Response

1`true

TrainEnter
RMURange

BOOL

Fig. 1. Top-Level

if resu = success
then response
else
unresponse

response

resu

RUM_res
Out
Response

Wait
In
Result

Communication2
In
Communication

establish

recieve

resu

Fig. 2. RMU communication

TrainInfo

STRING

1`"V2.1"

Map

STRING

TrainID

TrainID

register

map

TrainID

(TrainID,register)

Procedure

In

Procedure

get_map

Procedure
Result

Out

Procedure

(map,register)

[TrainID1=TrainID]

(mes,task)

cancel

(TrainID,cancel)

BufferPro

(mes,task)

transmit_pro

(TrainID,cancel)

TrainID1

Procedure

"V2.1"

(position,free)

Resource_sw

Out

ResourceSwitch

Fig. 3. RMU Procedure

n

(sw', Release)

Wait
Confirm

(sw', Release)

[n=1]

transmit_conform

((sw',release),Release)

Command

release

n+1

Flag2

1`0

(sw', Release)

[n=2]

releaseIC

n

1`0

(sw', Release)

Command2

Out

Command

n

INT

n

Release

(sw', Release)

(sw', Convert)

occupy

(sw', Occupy)

(sw', Occupy)

(sw', Occupy)

(sw', Lock)

Occupy

lock

(sw', Lock)

convert

Convert

Copy_orders

Lock

(sw', Convert)

orders

true

Command1

(sw', Order)

order

Order

In

Command

true

true

Buffer1

true

Resource
Result

Out

ResourceResult

true

true

BOOL

true

Resource_sw

In

ResourceSwitch

(sw,sta)

transmit_result

or

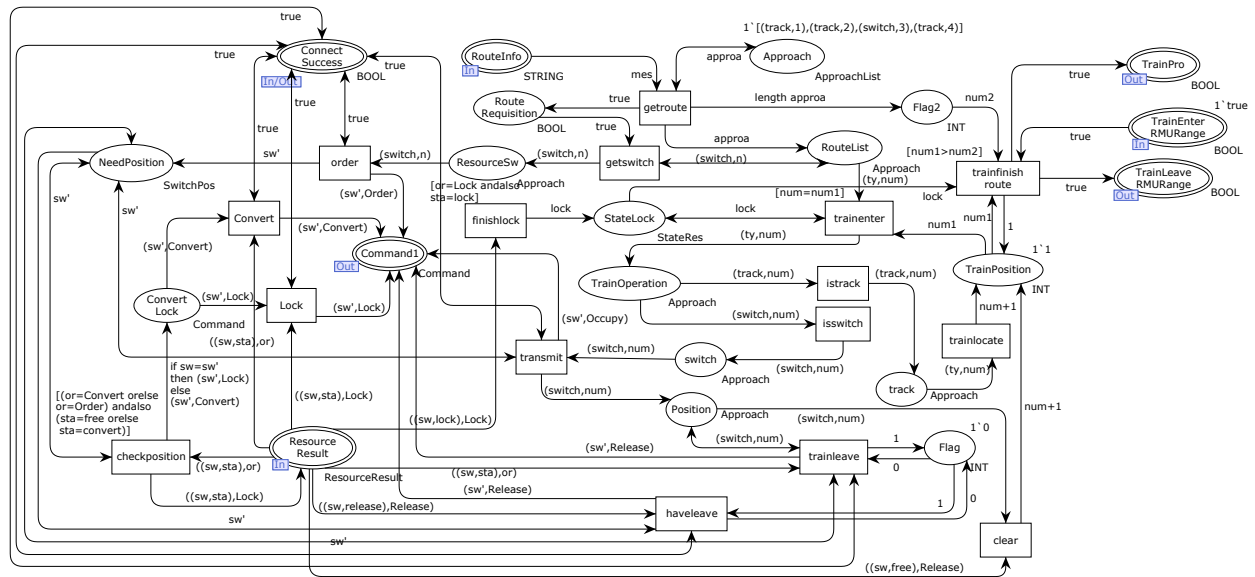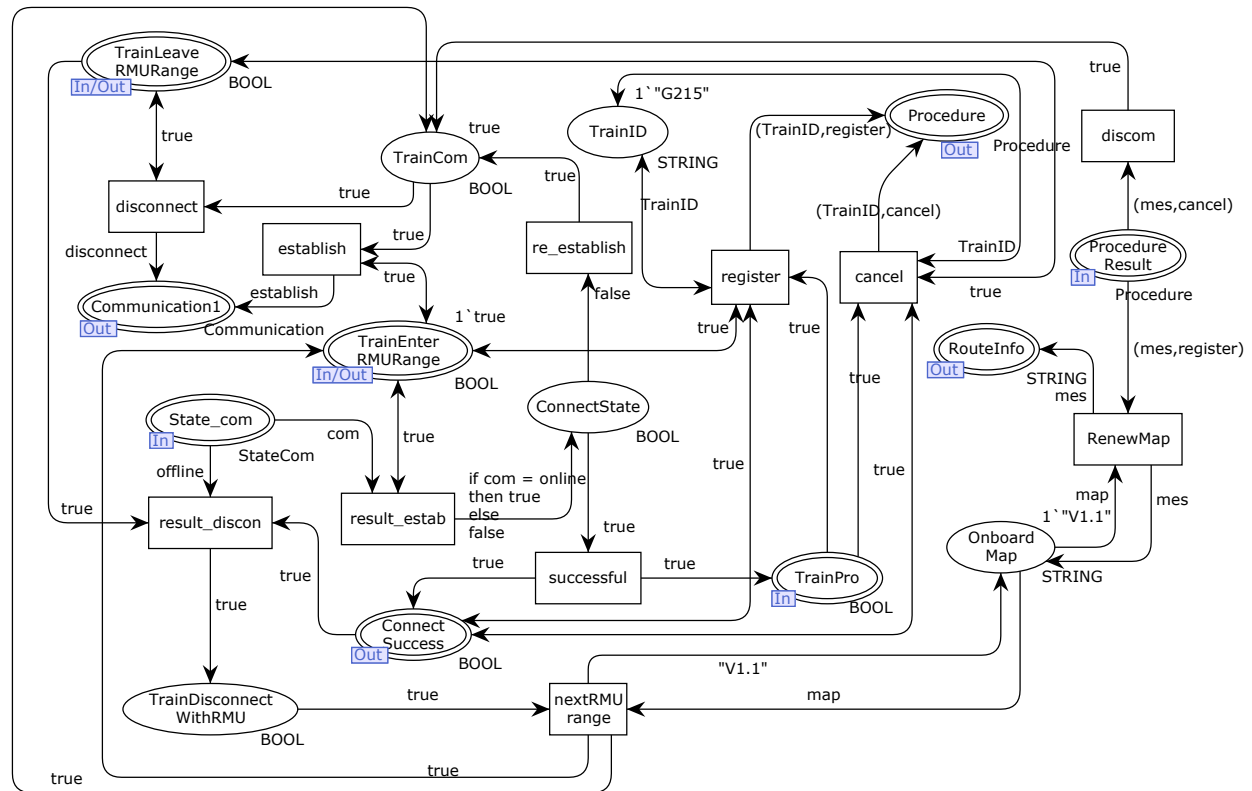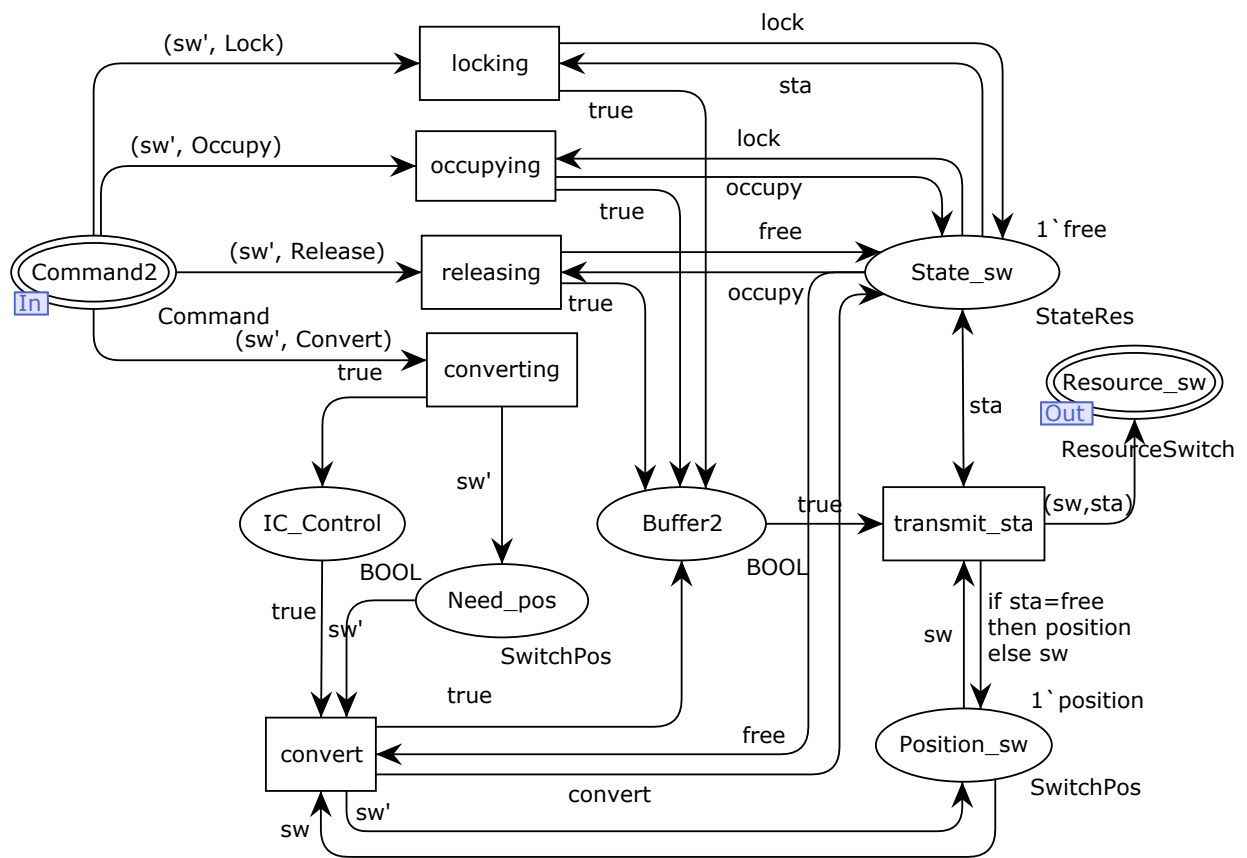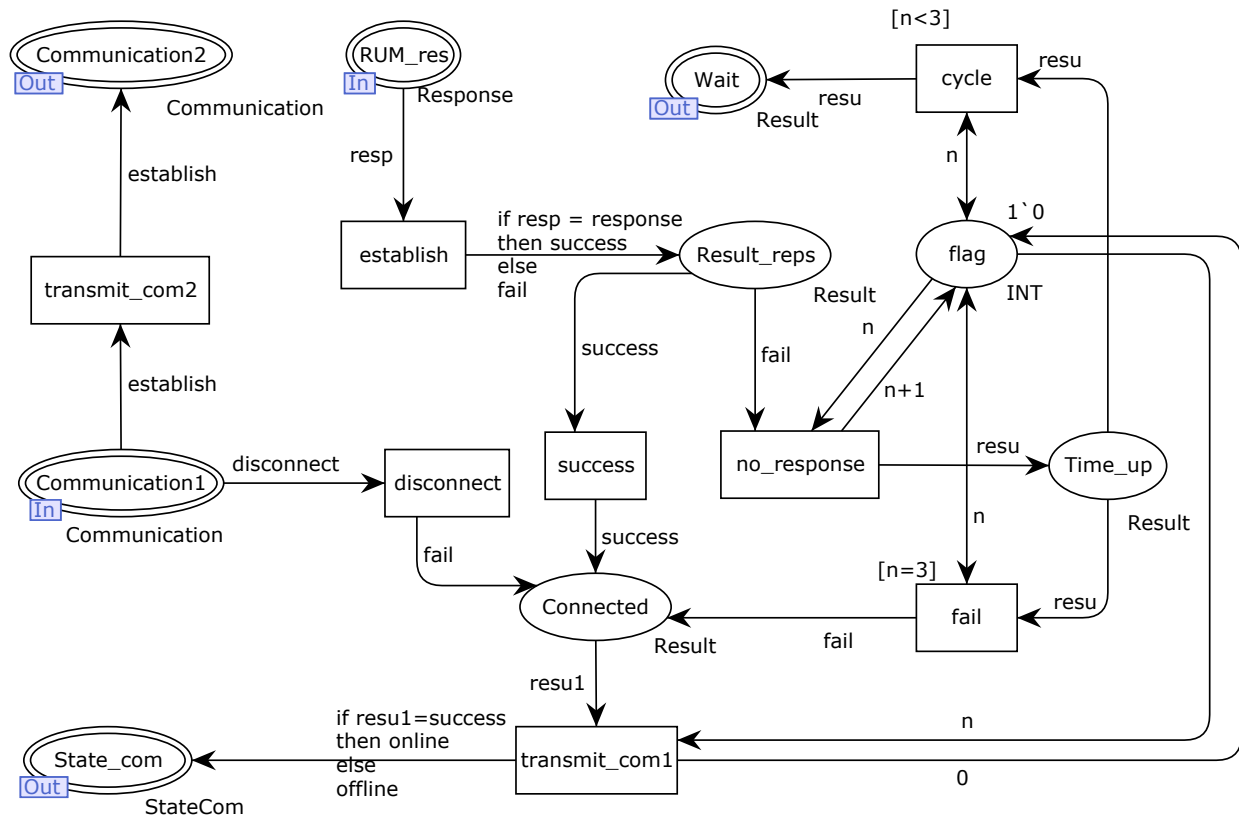((sw,sta),or)

Fig. 4. RMU command

Fig. 5. train approach

Fig. 6. train Procedure

Fig. 7. IC

Fig. 8. train_ground communication