

Developing Tool Support for  
Problem Diagrams with  
CPN and VDM++



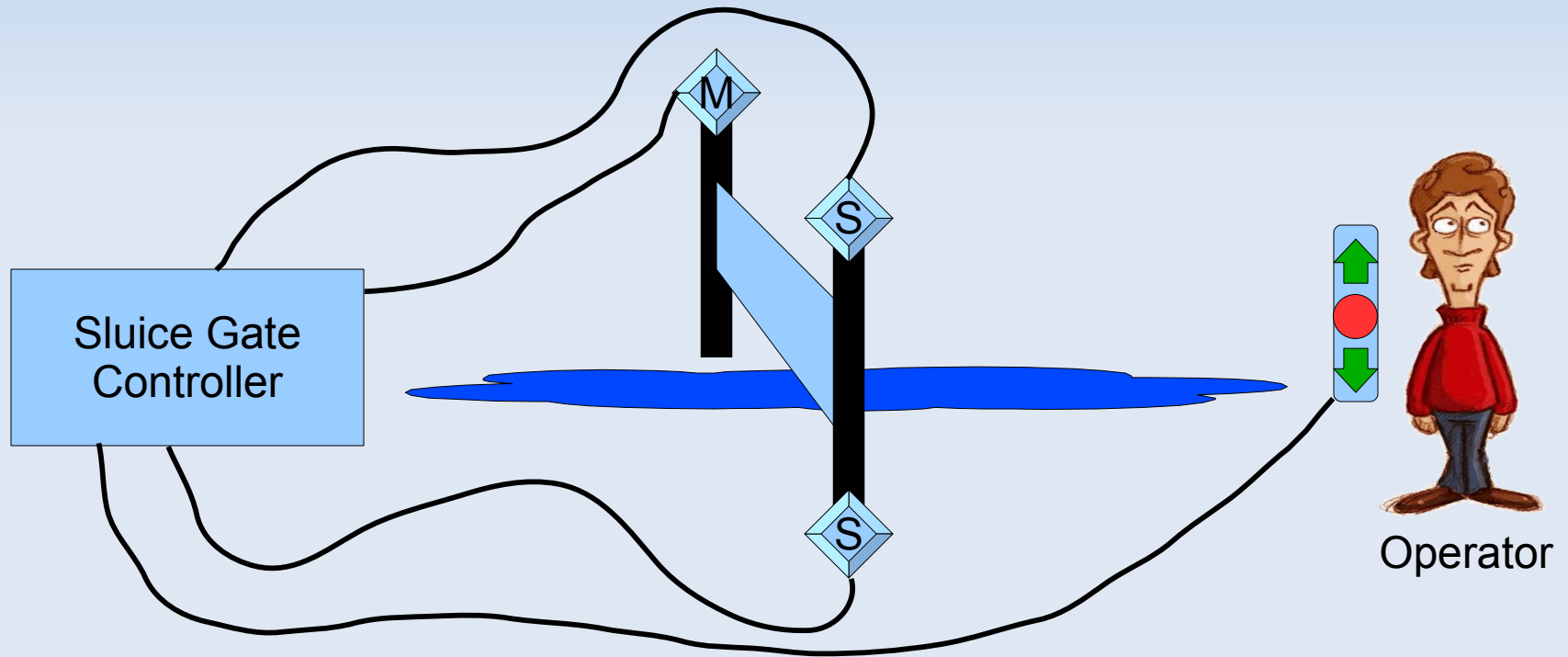
# Outline

- An example of a problem
- Coloured Petri Nets (CPN)
- Problem Diagrams
- Walk through a typical workflow
- Future work

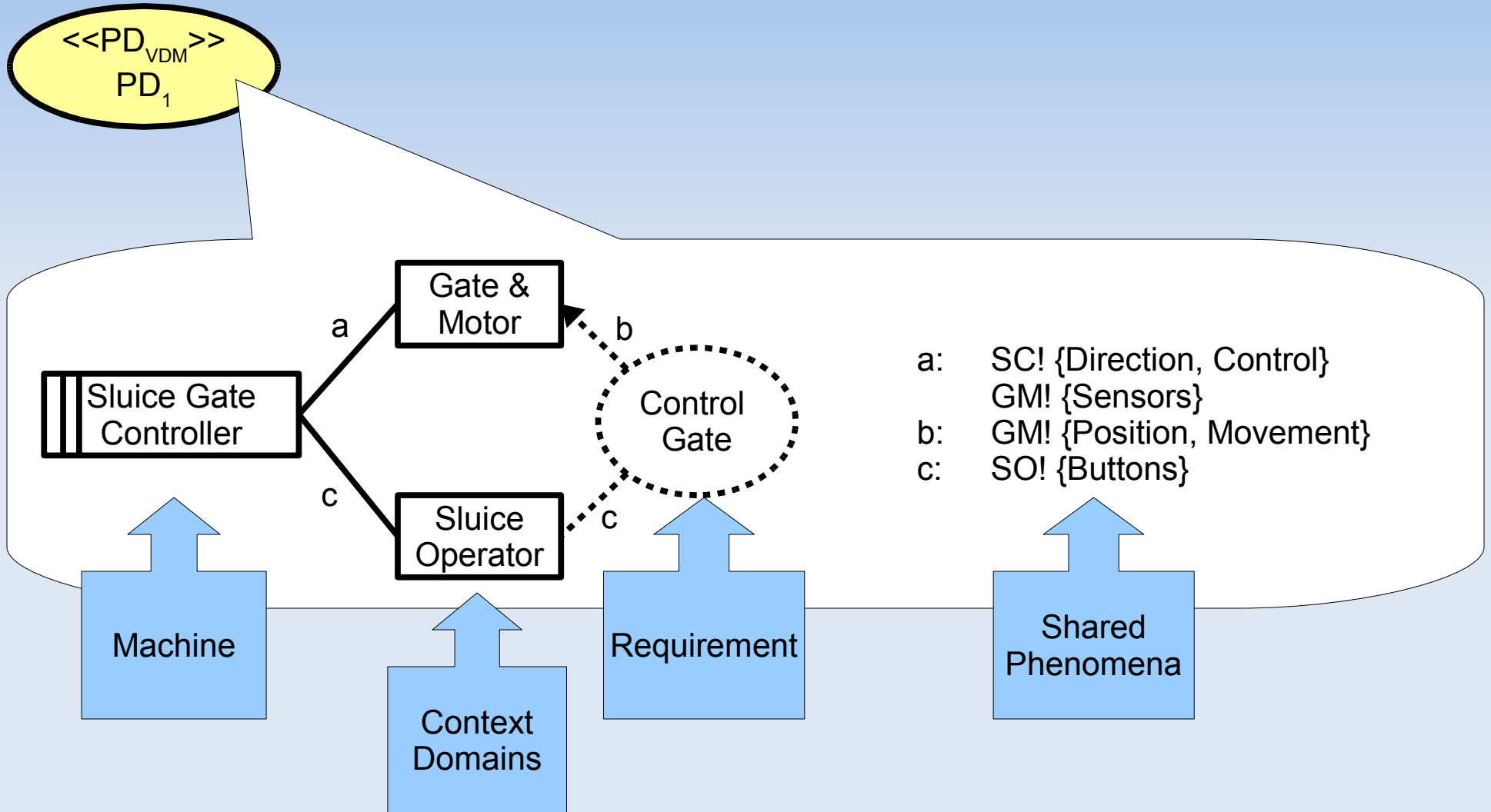
# Coloured Petri Nets (CPN)

- A graphical modeling language
- A high-level extension to Petri Nets adding:
  - Complex data types for tokens
  - A functional language for token manipulation/examination
  - Hierarchical constructs
  - Time (delays, timestamps)
- Tool support:
  - Graphical editing
  - Simulation by execution
  - State space analysis

# Sluice Gate Controller



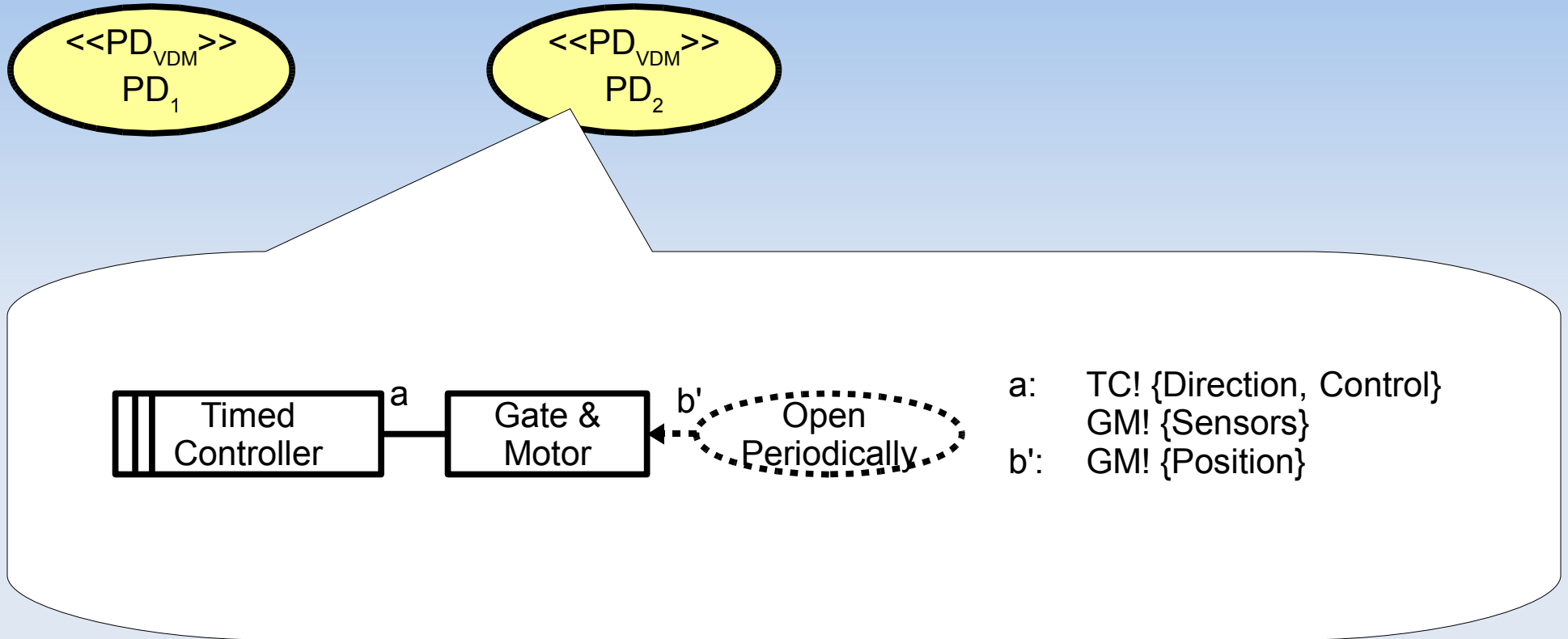
# Problem Diagrams (1)



# Problem Diagrams (2)

- Problem Diagrams:
  - Identify relevant entities in the near environment
  - Identify knowledge about structure in the environment
  - Identify shared phenomena (~interaction channels)
  - Do not describe behavior
  - Can be used to document Problem Decomposition

# Problem Diagrams (3)

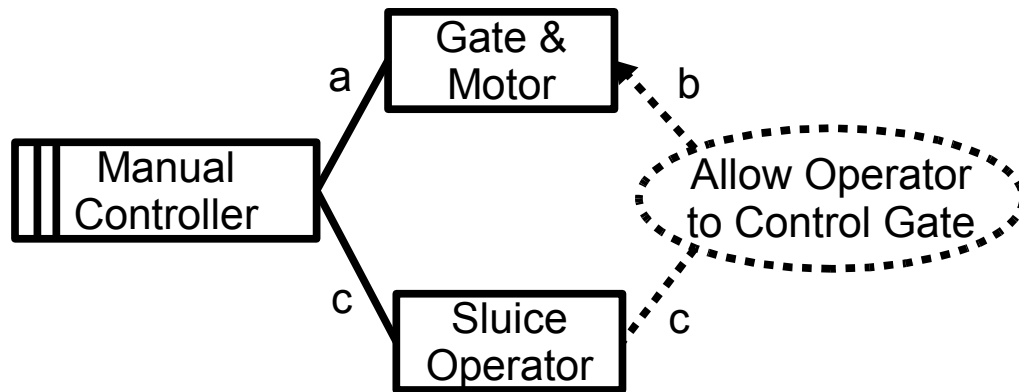


# Problem Diagrams (4)

<<PD<sub>VDM</sub>>>  
PD<sub>1</sub>

<<PD<sub>VDM</sub>>>  
PD<sub>2</sub>

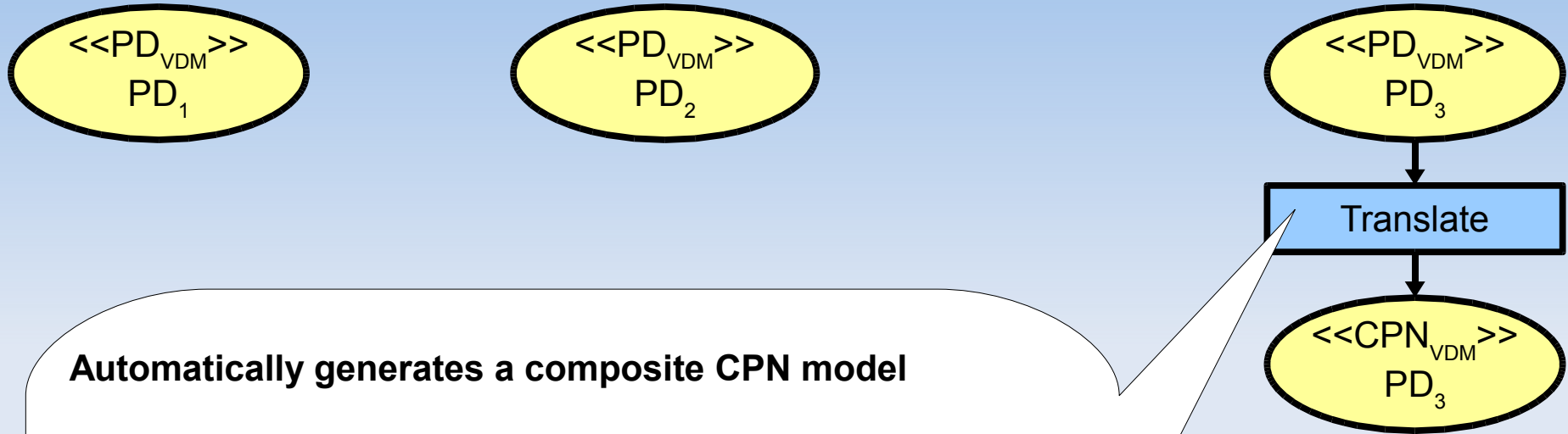
<<PD<sub>VDM</sub>>>  
PD<sub>3</sub>



- a: MC! {Direction, Control}
- GM! {Sensors}
- b: GM! {Position, Movement}
- c: SO! {Buttons}



# Translating



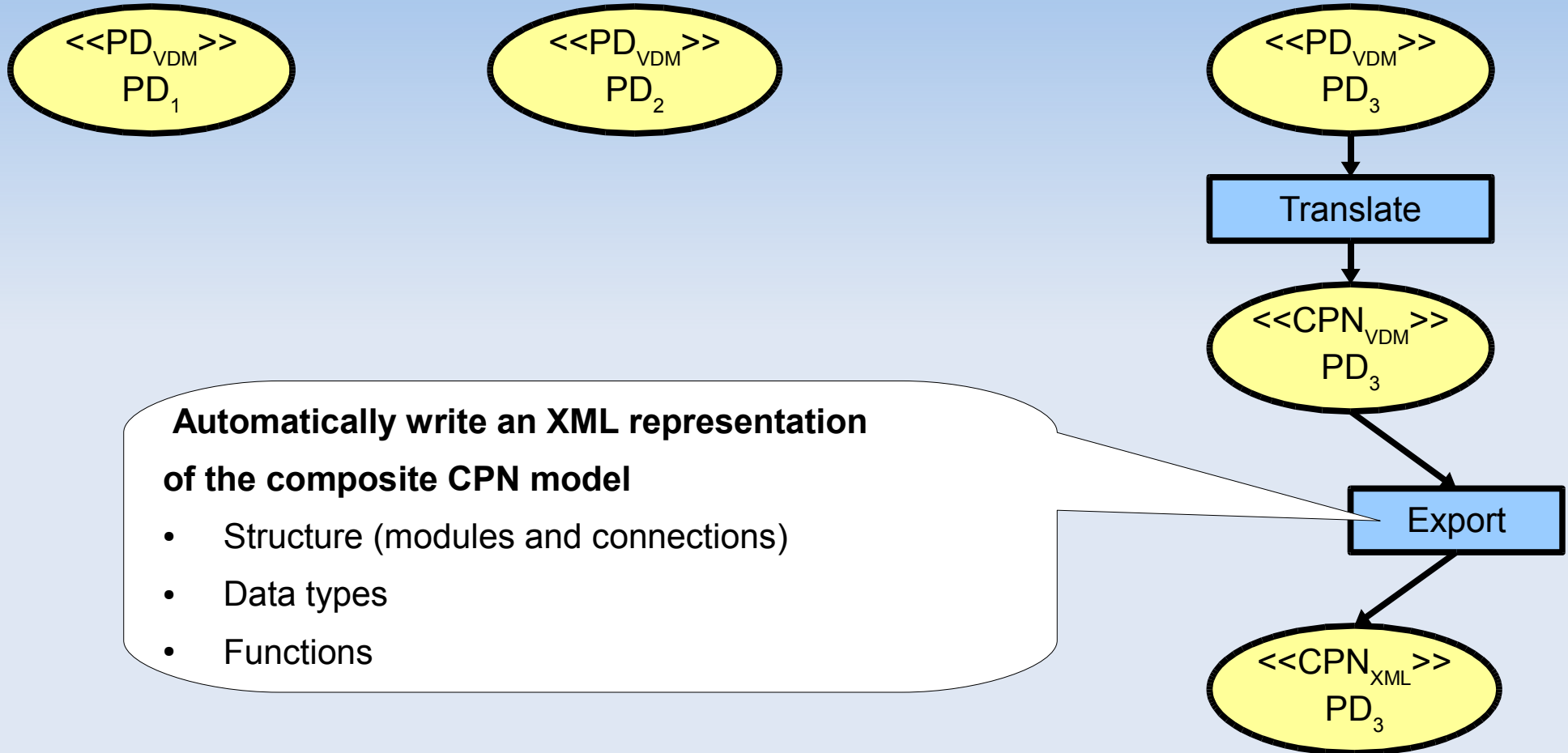
## Automatically generates a composite CPN model

### Contents:

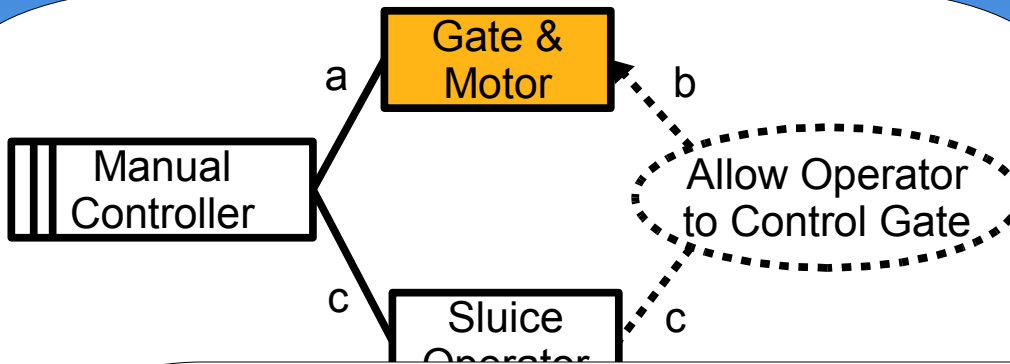
- a module per context domain in the PD
- a machine module
- a link module
  - enables communication through shared phenomena (only)
  - preserves the structure of the PD
  - records traces of phenomena activity
- a requirement module

Behavior of context and machine domains is initially free/spontaneous

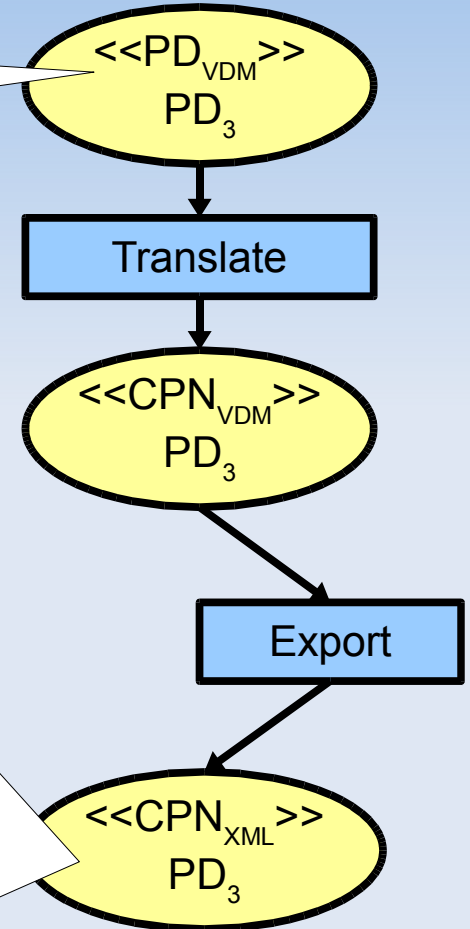
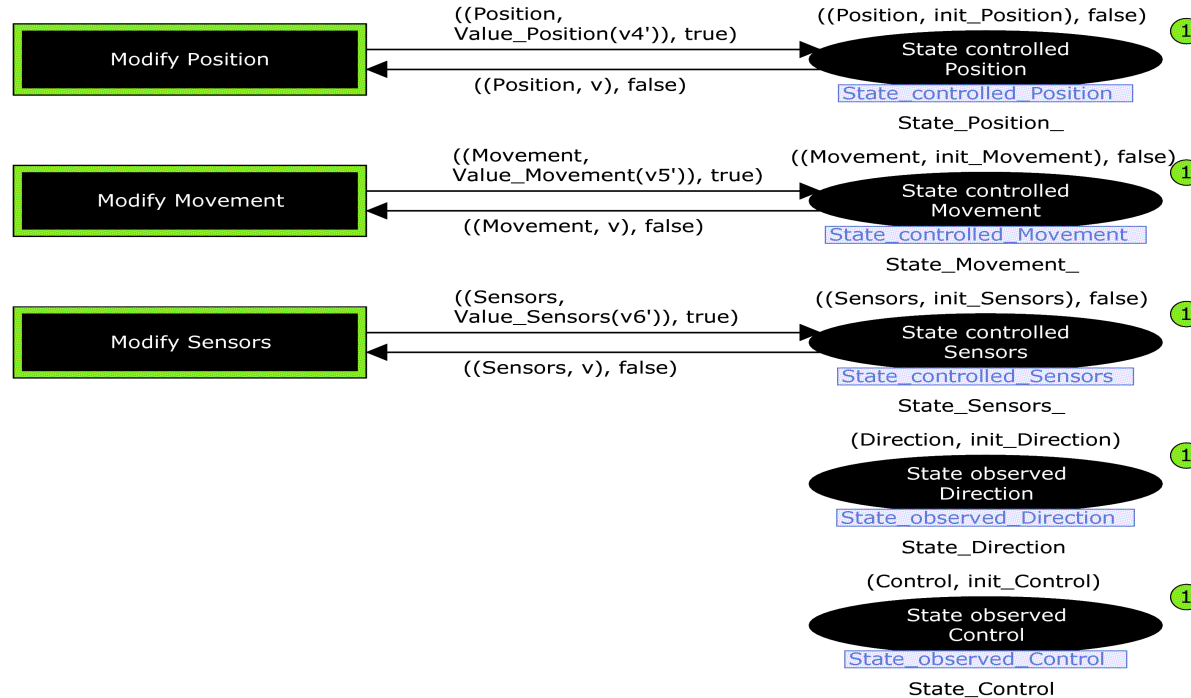
# Exporting (1)



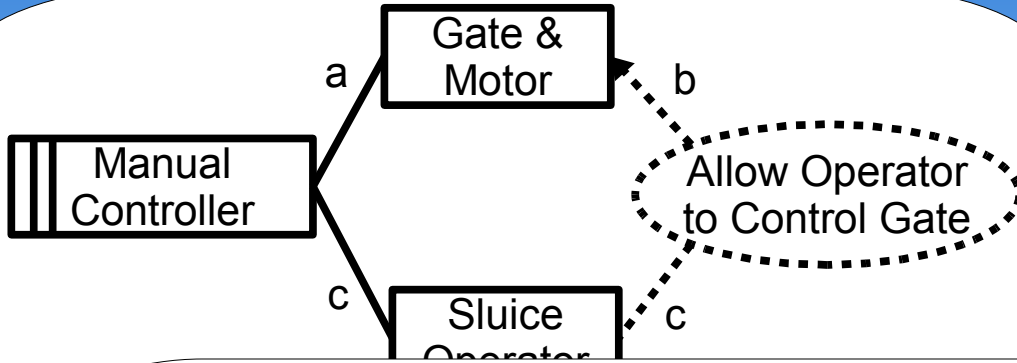
# Exporting (2)



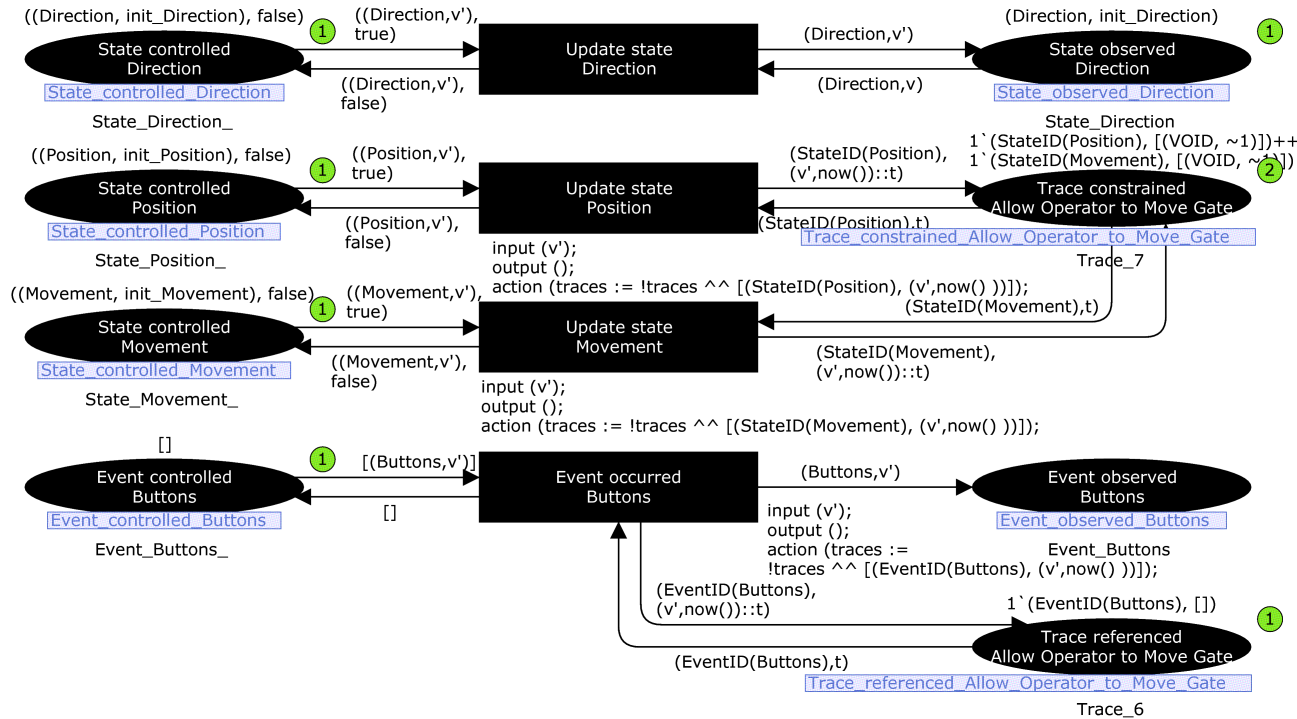
## Gate & Motor



# Exporting (3)



## Link module



$\ll PD_{VDM} \gg$   
 $PD_3$

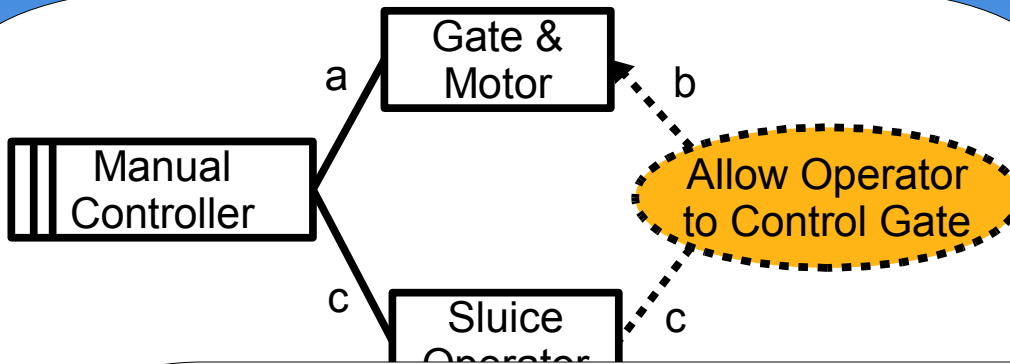
Translate

$\ll CPN_{VDM} \gg$   
 $PD_3$

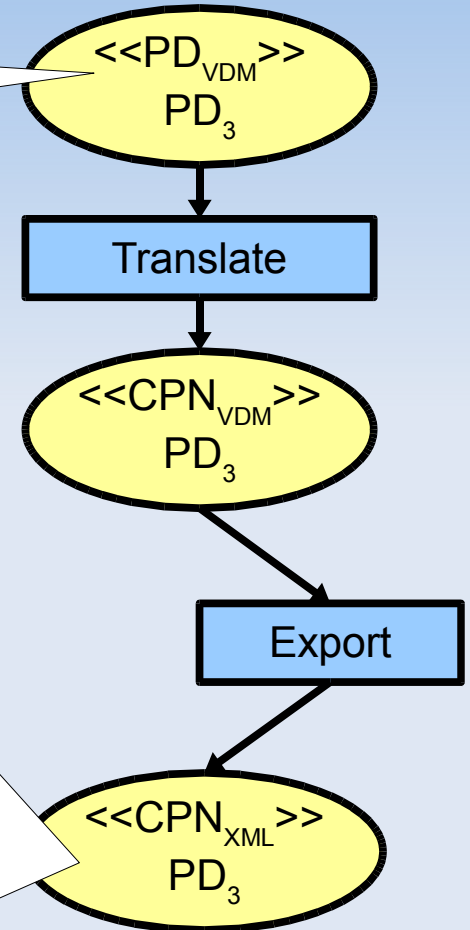
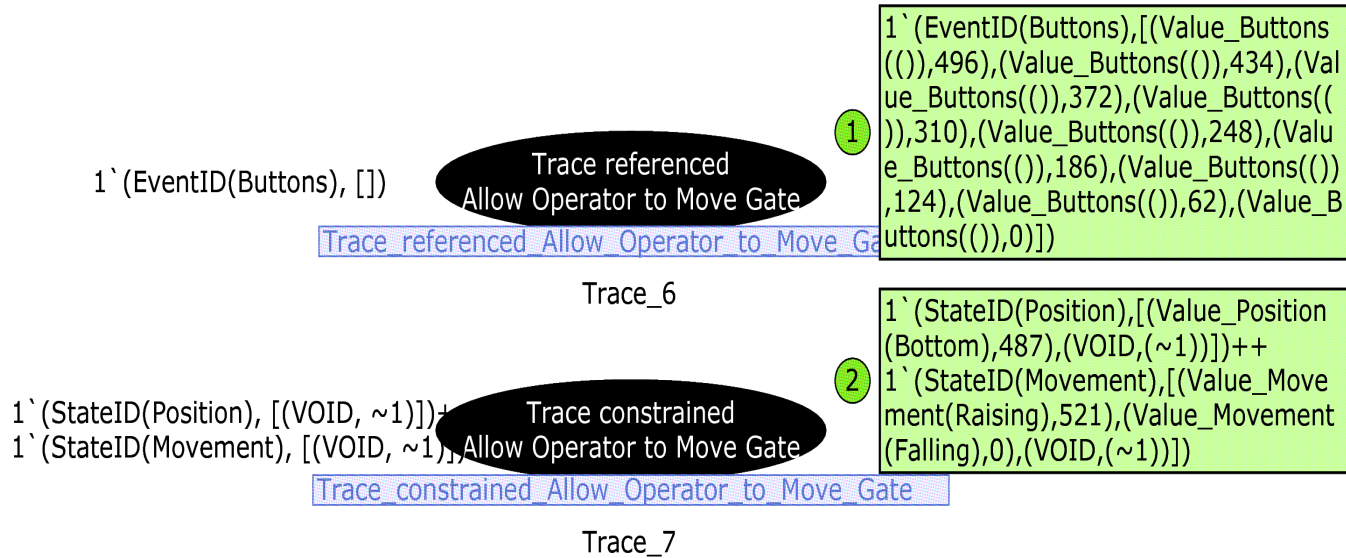
Export

$\ll CPN_{XML} \gg$   
 $PD_3$

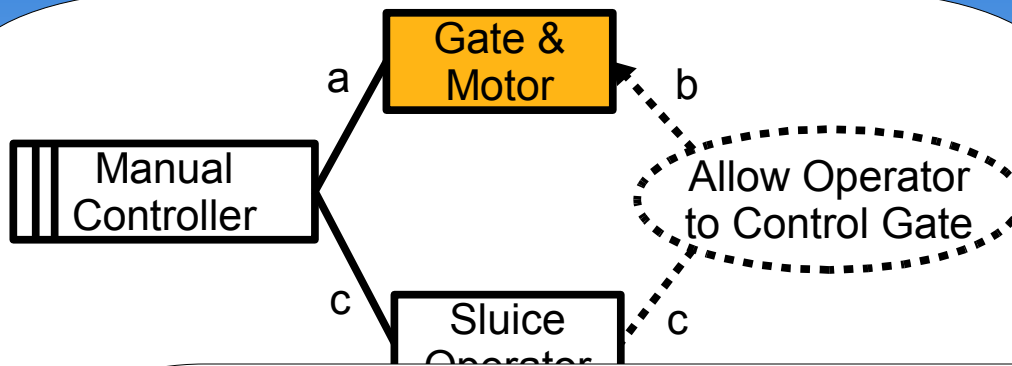
# Exporting (4)



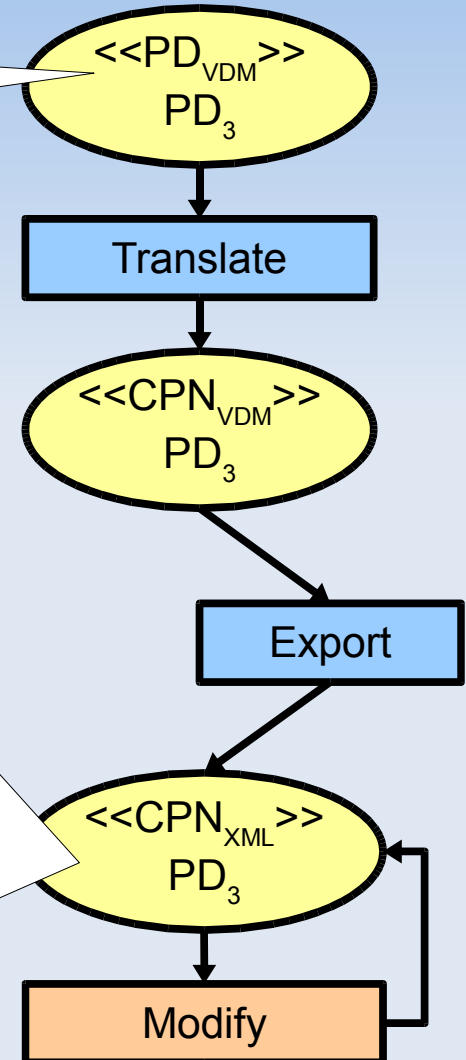
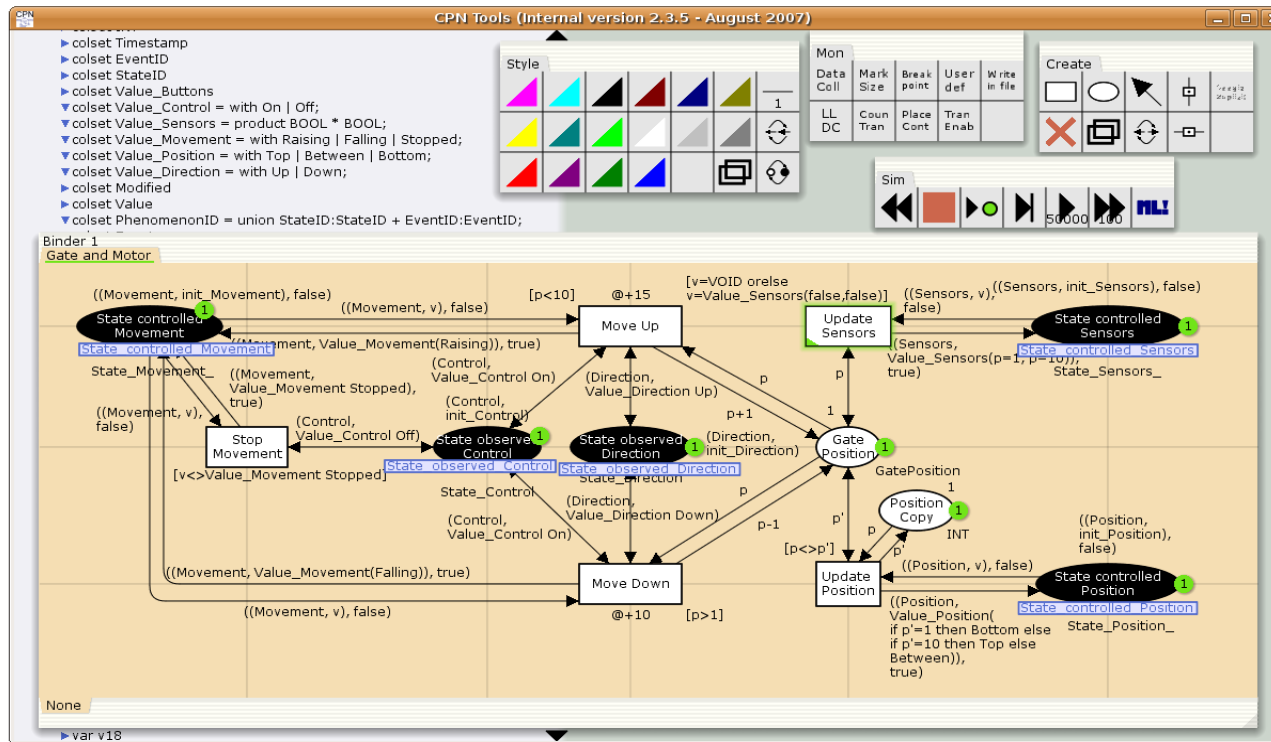
## Requirement module



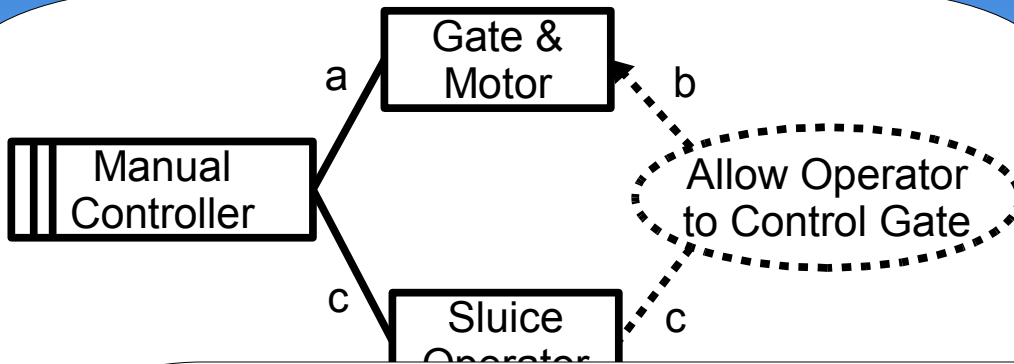
# Modifying (1)



## CPN Tools



# Modifying (2)



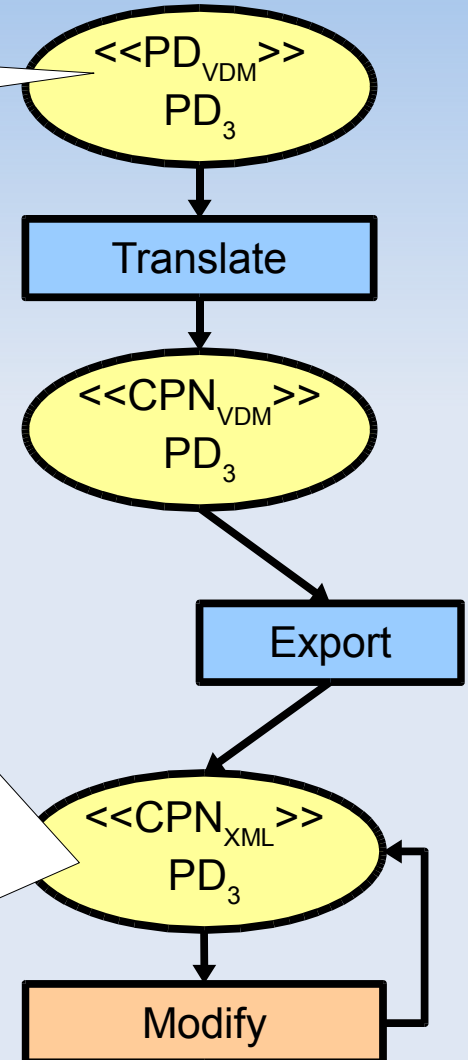
## Automatically generated phenomena color sets (data types):

```
colset Value_Buttons = bool;  
colset Value_Control = bool;  
colset Value_Sensors = bool;  
colset Value_Movement = bool;  
colset Value_Position = bool;  
colset Value_Direction = bool;
```

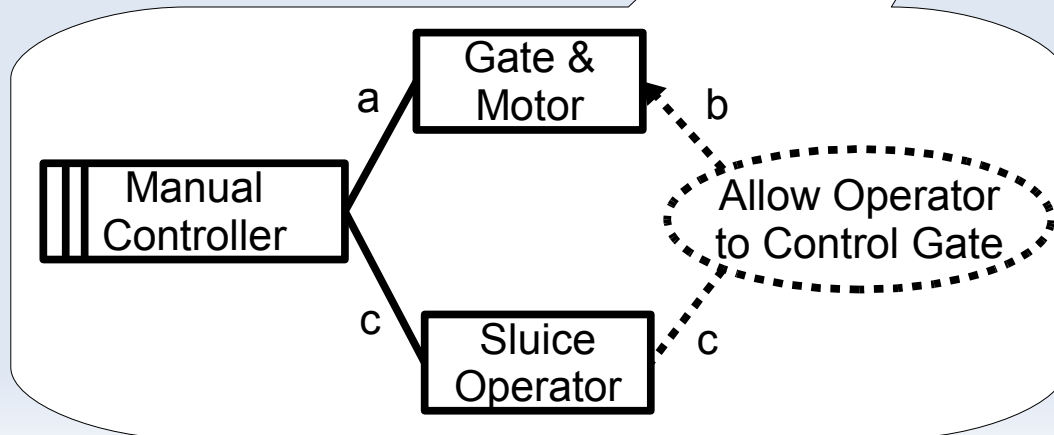
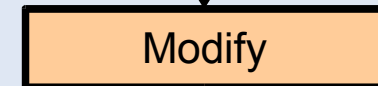
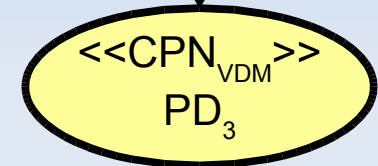
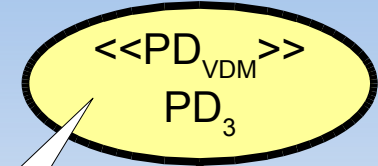
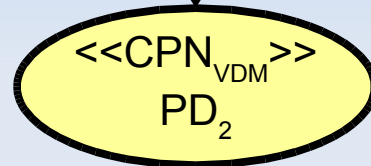
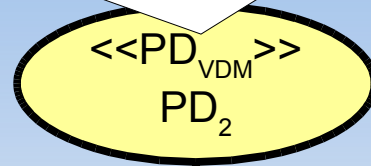
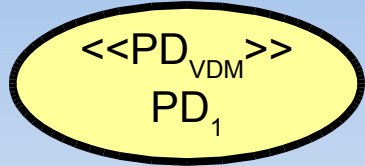
+ color sets for states, events, traces, timestamps etc.  
(need no manual modification)

## Manually modified:

```
colset Value_Buttons = unit;  
colset Value_Control = with On | Off;  
colset Value_Sensors = product BOOL * BOOL;  
colset Value_Movement = with Raising | Falling | Stopped;  
colset Value_Position = with Top | Between | Bottom;  
colset Value_Direction = with Up | Down;
```

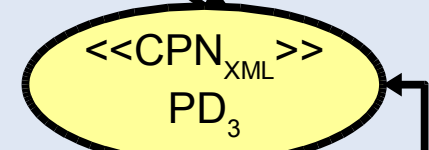
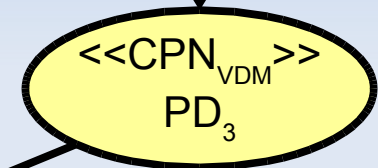
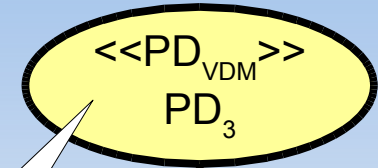
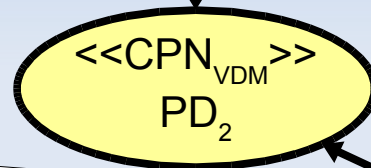
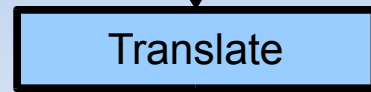
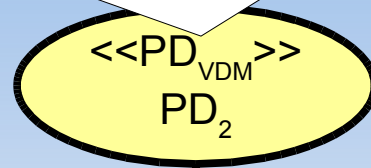
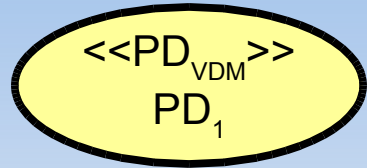


# Importing





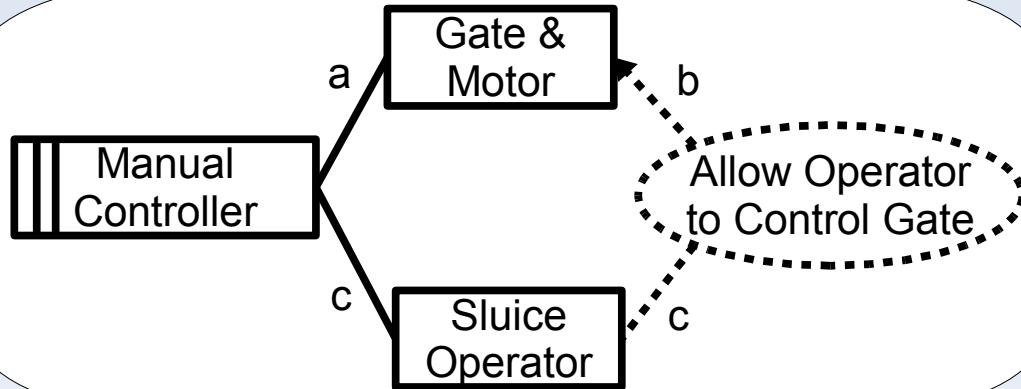
# Combining (1)



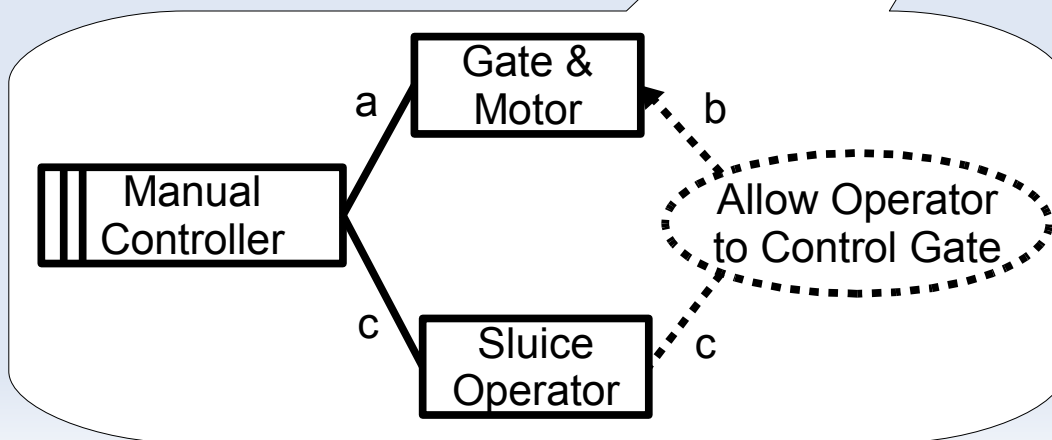
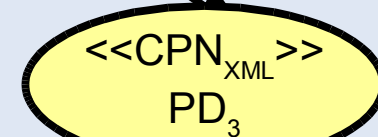
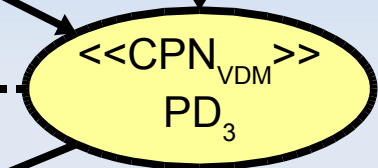
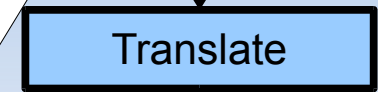
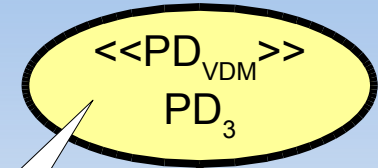
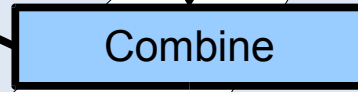
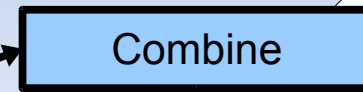
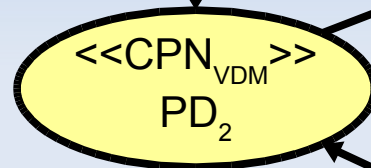
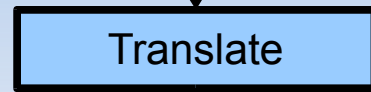
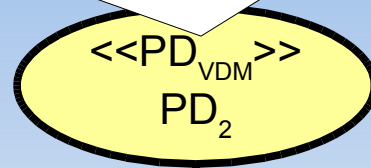
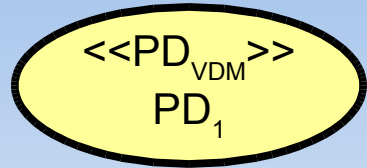
Prerequisites (to  $PD_3$ ):

- Valid structure (wrt. PD)
- No conflicting modules
- No interface extension

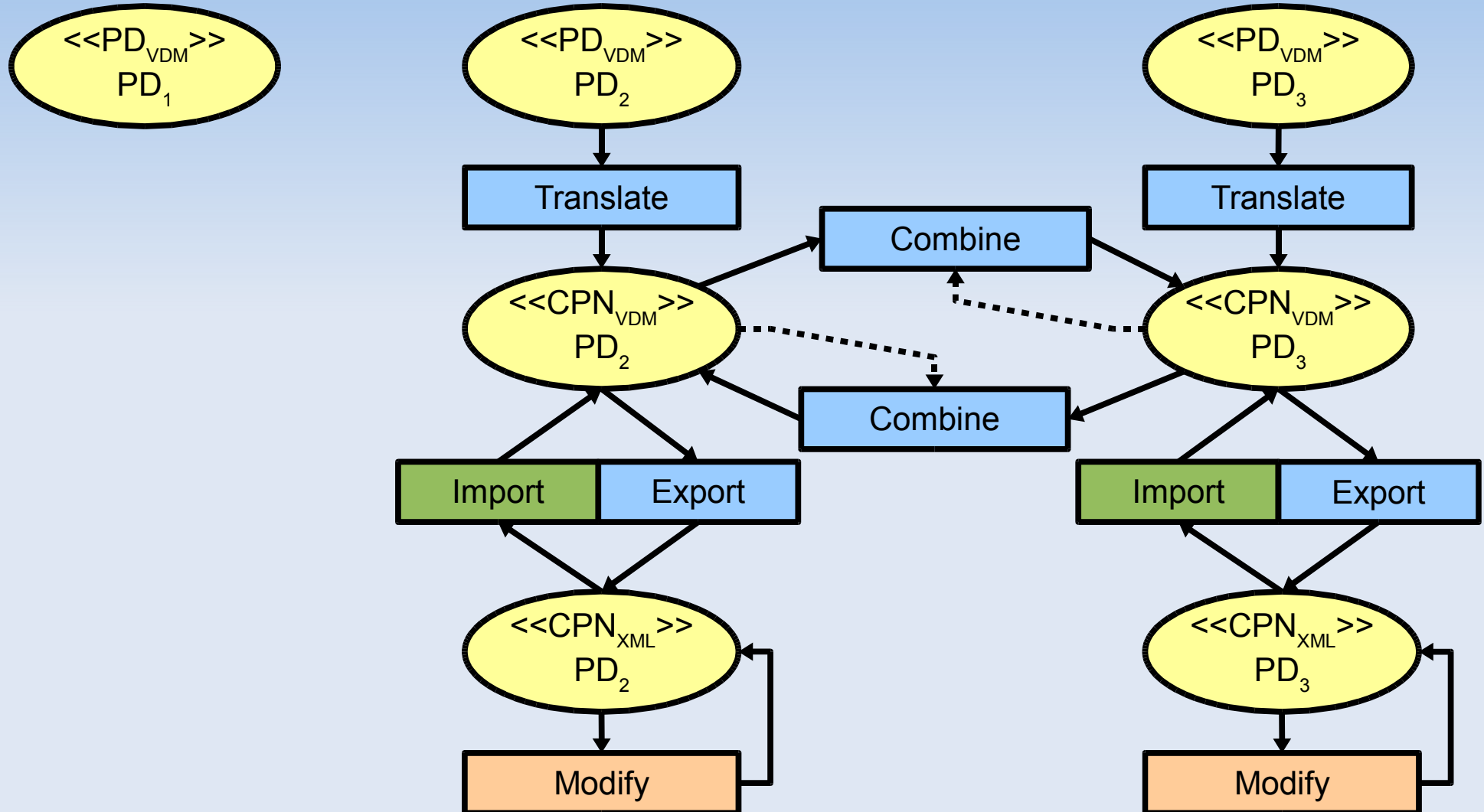
Combine



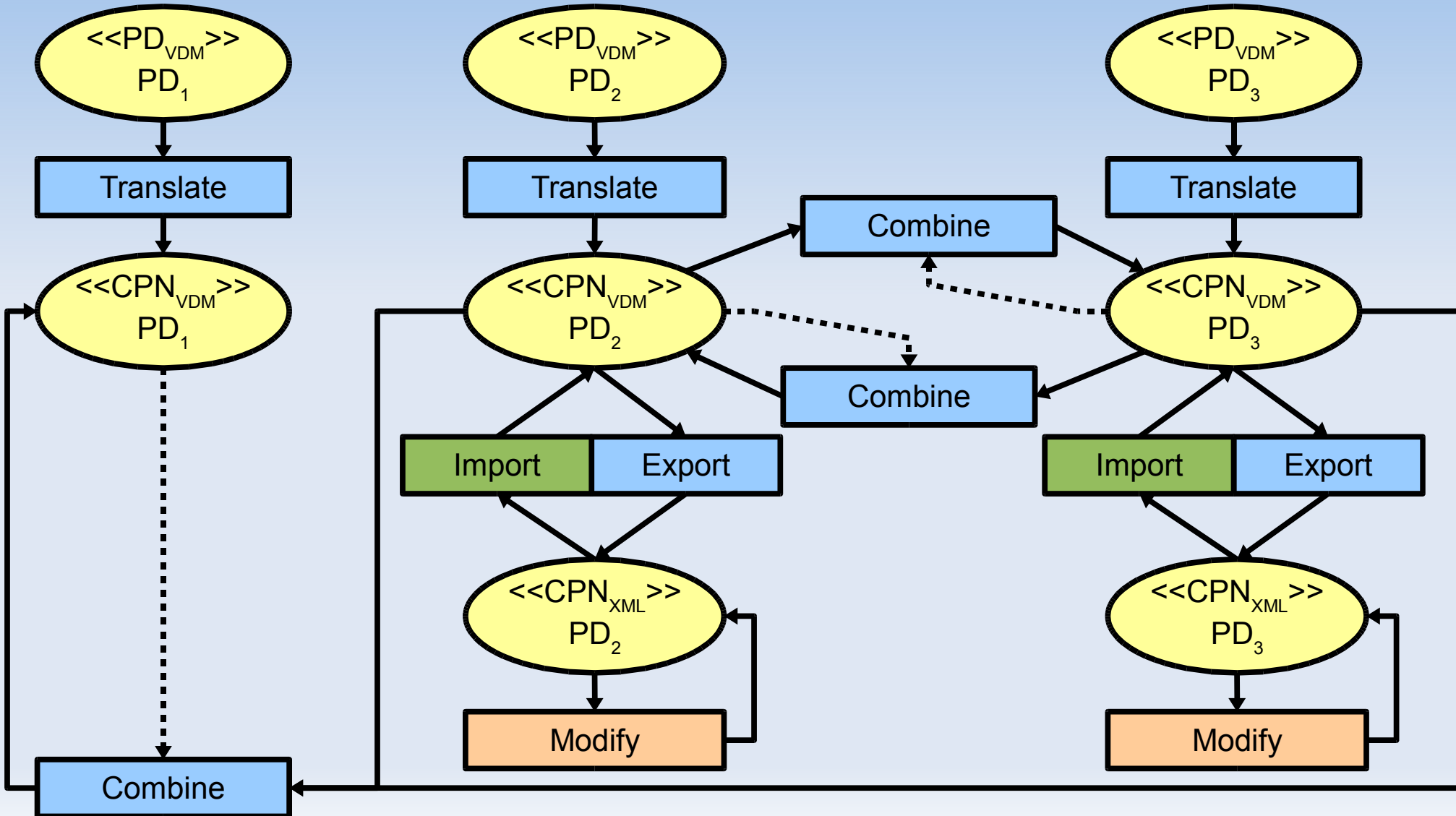
# Combining (2)



# Parallel iterations



# Closing the loop



# The Role of VDM++

- The following is specified using VDM++:
  - The syntax of hierarchical CPN
  - The syntax of Problem Diagrams
  - Algorithms:
    - Translate
      - Generation of link/machine/domain/requirement modules
      - Generation of color sets
    - Combine
      - Structural validation of input models
    - Export
      - XML generation
      - Color set dependency

# Future / Current Work

- Automated checking of traces against real-time requirements expressed as high-level sequence diagrams (i.e. scenarios)
- Integration with the Problem Oriented Engineering approach