School of Electrical Engineering

Department of Electrical Engineering and Automation

ELEC 8201 Control & Automation

Design of Automation Applications

Valeriy Vyatkin, Pranay Juhnjhunwala

Aalto University

2022

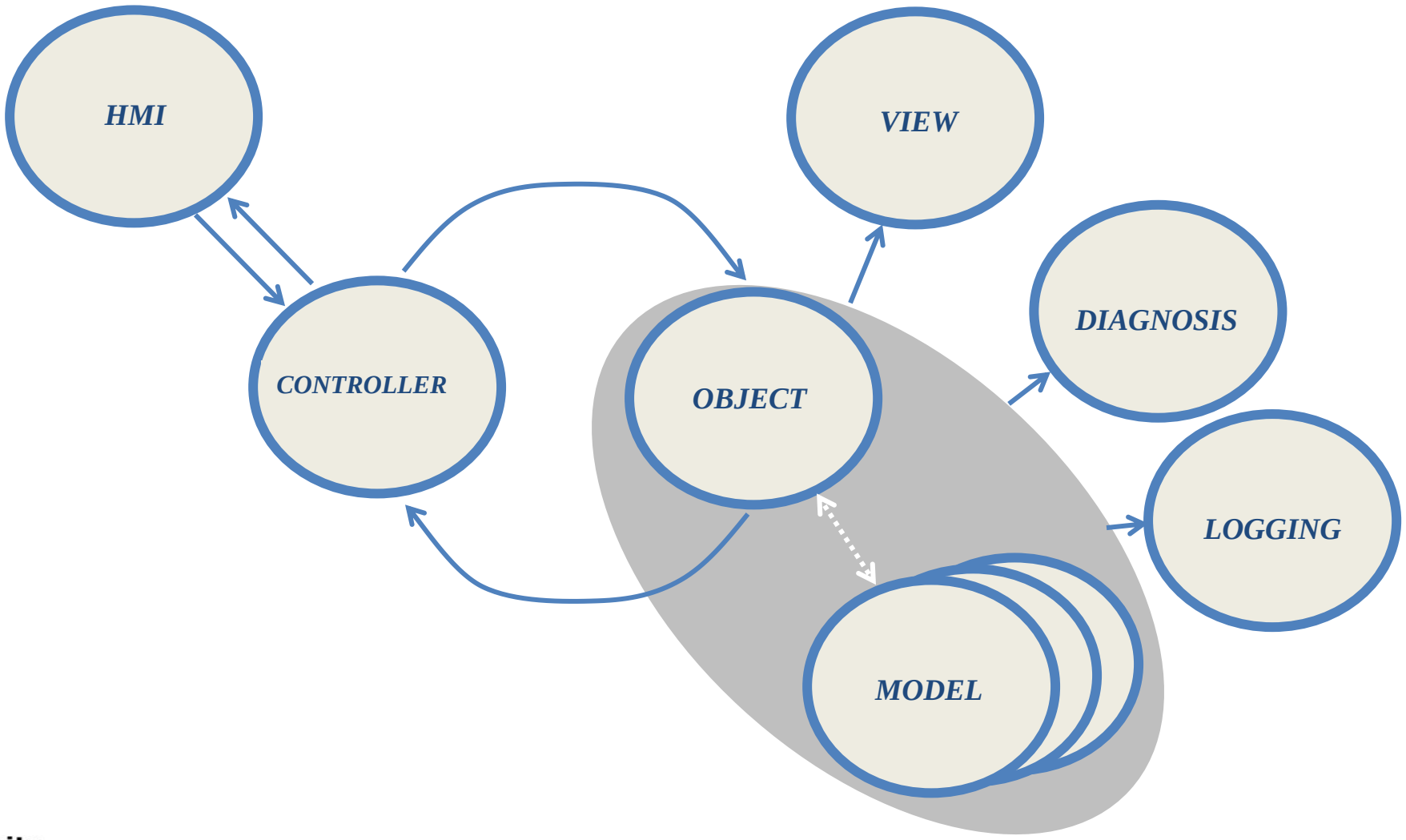
Plan

1. Develop complex applications with function blocks
2. How to implement continuous control in function blocks?
3. Object-based design
4. Service-oriented design

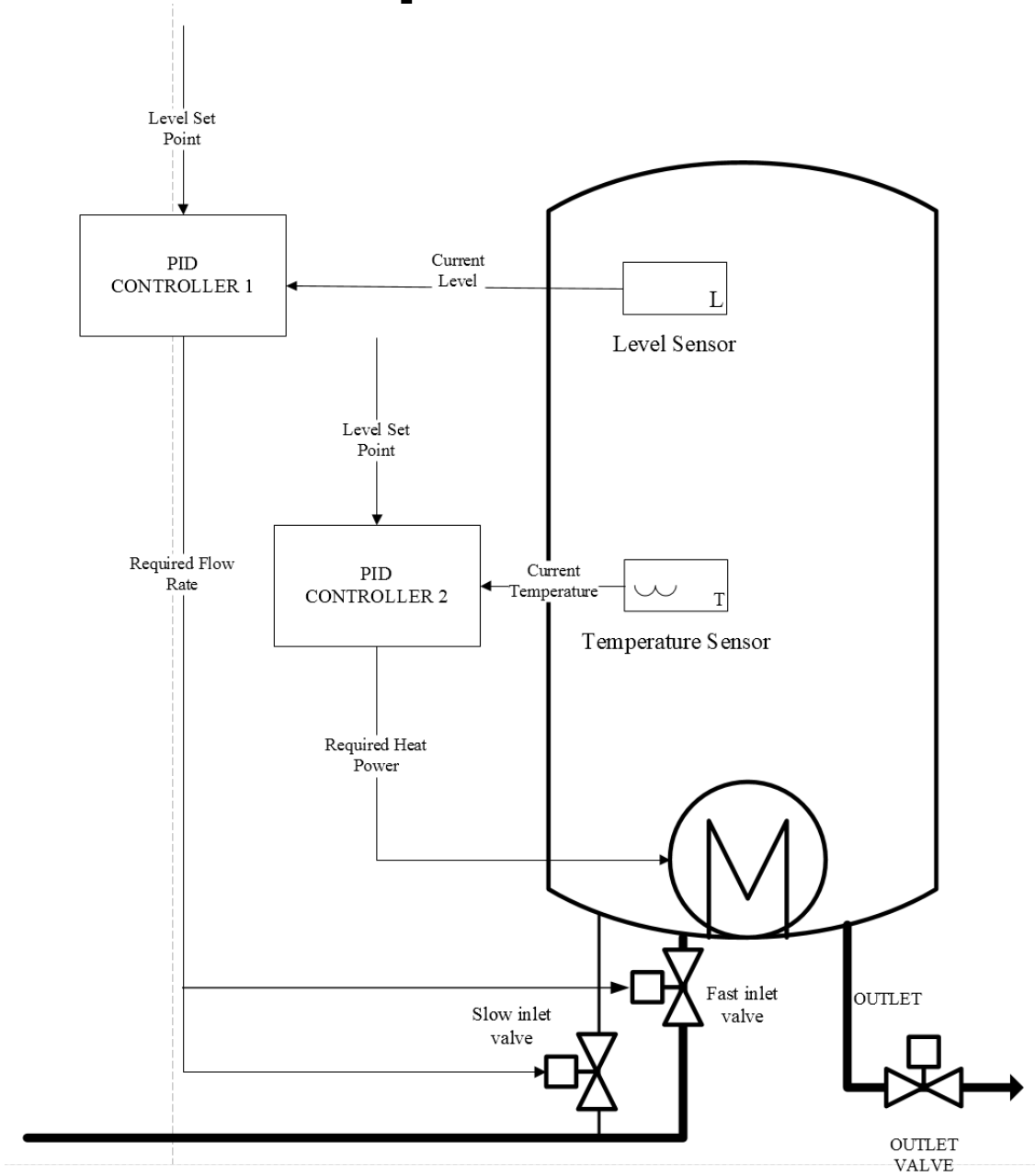
Start with interfaces

- Plant inputs and outputs
- Human-machine interfaces
  - Buttons and switches
  - Display process variables and trends
  - …

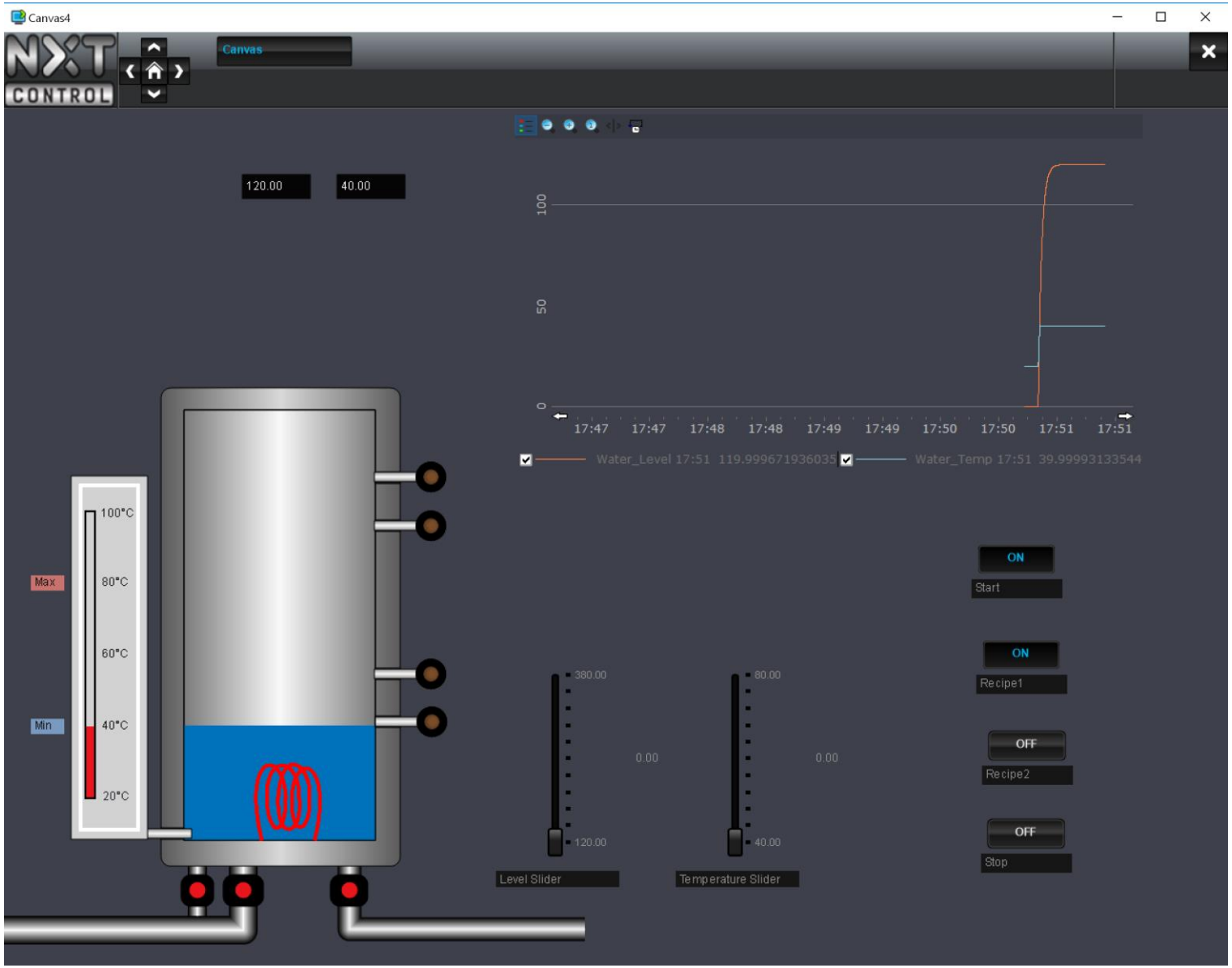
Model-View-Control design pattern



Example: Tank



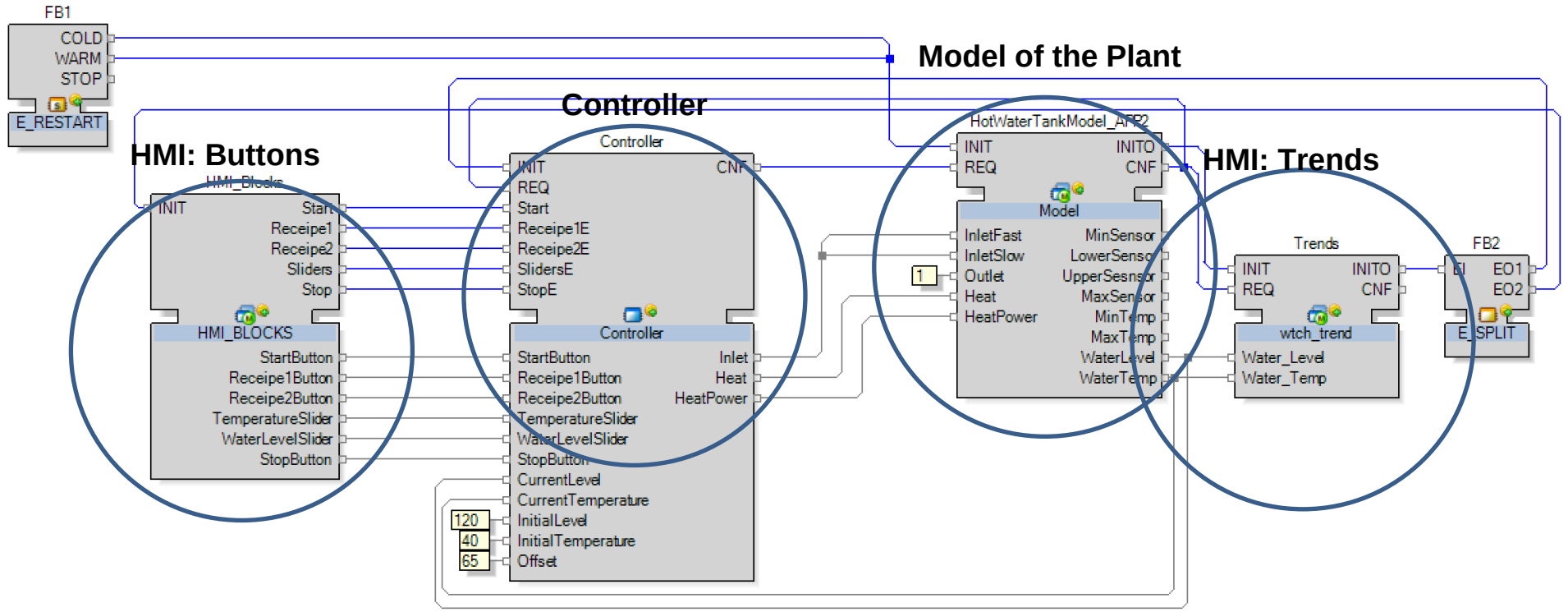
Simulated tank and HMI



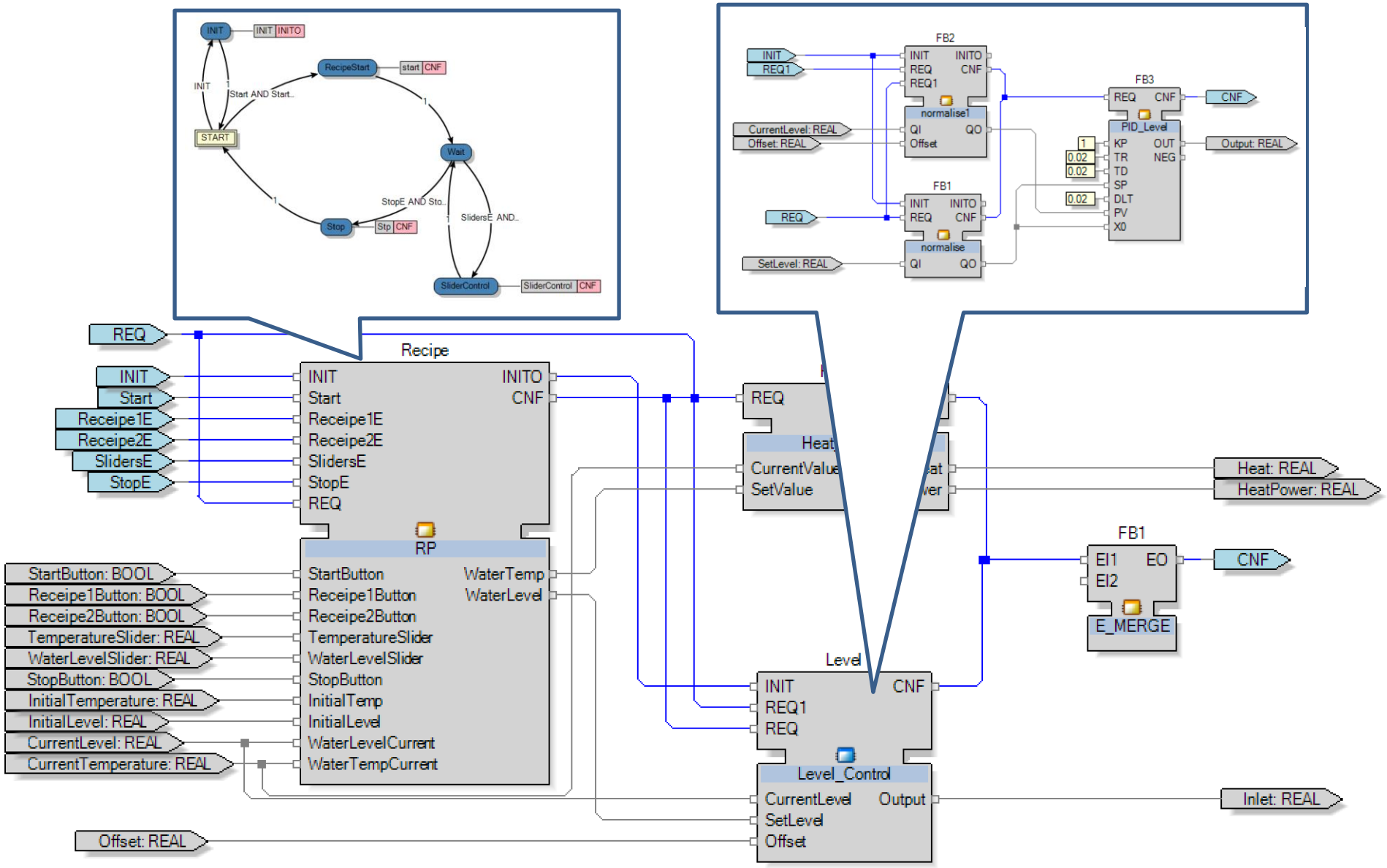
Requiremets?

- Maintain water level
- Maintain certain temperature
- Maintain temperature while increasing water level
- …
- More general:
  - Apply a recipe to a batch of product
  - While ensuring bumpless change of water level and temperature

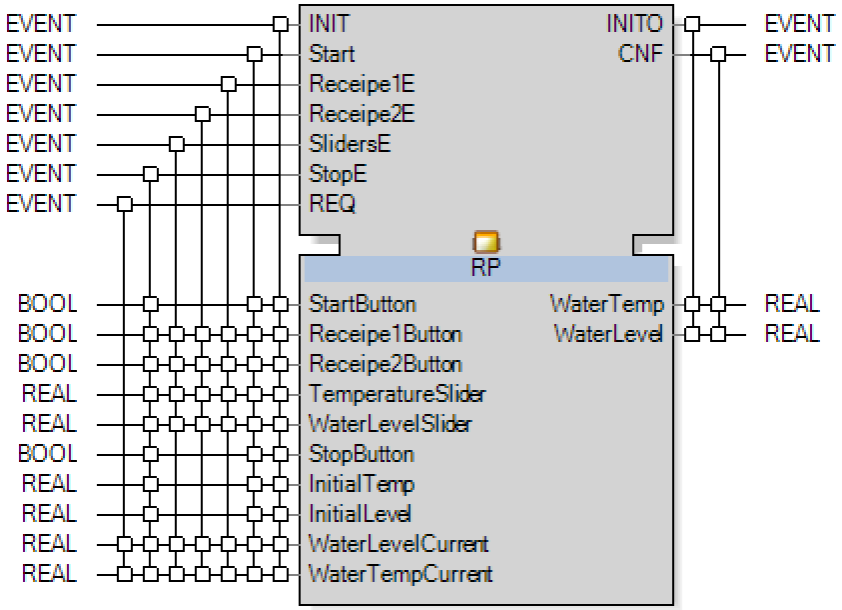
Application structure

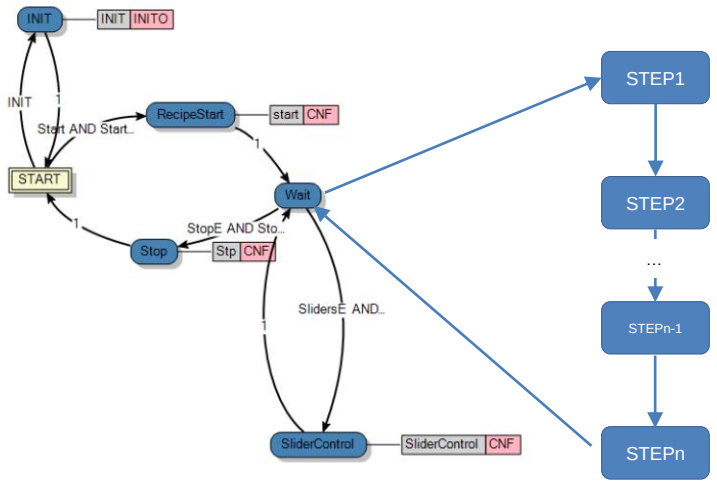


Controller

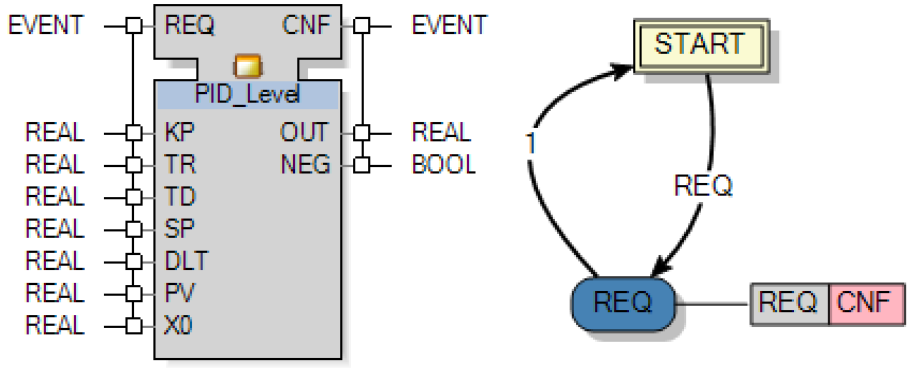

INIT
INIT INITO
RecipeStart
start CNF
START
Start AND Start..
Wait
StopE AND Sto..
Stop
Stp CNF
SliderE AND..
SliderControl
SliderControl CNF
FB2
INIT
REQ1
INITO
CNF
normalise1
QI
QO
Offset
CurrentLevel: REAL
Offset: REAL
FB3
REQ
CNF
PID_Level
KP
TR
TD
SP
DLT
PV
X0
OUT
NEG
1
0.02
0.02
0.02
Output: REAL
FB1
REQ
SetLevel: REAL
normalise
REQ
Recipe
INITO
CNF
Start
Receipe1E
Receipe2E
SlidersE
StopE
RP
StartButton
Receipe1Button
Receipe2Button
TemperatureSlider
WaterLevelSlider
InitialTemp
InitialLevel
WaterLevelCurrent
WaterTempCurrent
WaterTemp
WaterLevel
StartButton: BOOL
Receipe1Button: BOOL
Receipe2Button: BOOL
TemperatureSlider: REAL
WaterLevelSlider: REAL
StopButton: BOOL
InitialTemperature: REAL
InitialLevel: REAL
CurrentLevel: REAL
CurrentTemperature: REAL
Offset: REAL
Heat
CurrentValue
SetValue
Heat: REAL
HeatPower: REAL
FB1
EI1
EI2
EO
E_MERGE
CNF
Level
REQ1
Level_Control
CurrentLevel
SetLevel
Output
Inlet: REAL

Recipe


EVENT
EVENT
EVENT
EVENT
EVENT
EVENT
EVENT
INIT
Start
Receipe1E
Receipe2E
SlidersE
StopE
REQ
INITO
CNF
EVENT
EVENT
RP
BOOL
BOOL
BOOL
REAL
REAL
BOOL
REAL
REAL
REAL
REAL
StartButton
Receipe1Button
Receipe2Button
TemperatureSlider
WaterLevelSlider
StopButton
InitialTemp
InitialLevel
WaterLevelCurrent
WaterTempCurrent
WaterTemp
WaterLevel
REAL
REAL


INIT
INIT INITO
INIT
Start AND Start..
RecipeStart
start CNF
START
Wait
StopE AND Sto..
Stop
Stp CNF
SlidersE AND..
SliderControl
SliderControl CNF
STEP1
STEP2
...
STEPn-1
STEPn

Continuous control of level and temperature




```
ALGORITHM REQ IN ST:
(* Add your comment (as per IEC 61131-3) here
PID with Bumpless Transfer and Anti-Reset Windup, REAL PV+XOUT
*)
ERROR:=PV-SP;

IF FIRSTTIME THEN
   ITERM:=(ERROR-X0)/KP;
   DTERM:=0;
ELSE
   ITERM:=ITERM+ERROR*DLT/TR;
   DTERM:=(3*(PV-X3)+X1-X2)*TD/DLT/10;
END_IF;
OUT:=-KP*ERROR-ITERM-DTERM;
X3:=X2;X2:=X1;X1:=PV;
ITERM:=ITERM-ERROR*DLT/TR;
NEG:=0;
IF OUT<0 THEN
   NEG := 1;
   // OUT:=0;
ELSIF OUT>100 THEN
   OUT:=100;
END_IF;
FIRSTTIME:=FALSE;
END_ALGORITHM
```

KP: Proportional constant

TR: Integral constant
TD: Derivative constant

SP: Setpoint. The target for your process variable
DLT: Cycle time constant
PV: Process Variable, the value you want to control
X0: Default initial value.
OUT: PID output

NEG: Flag indicating that OUT<0

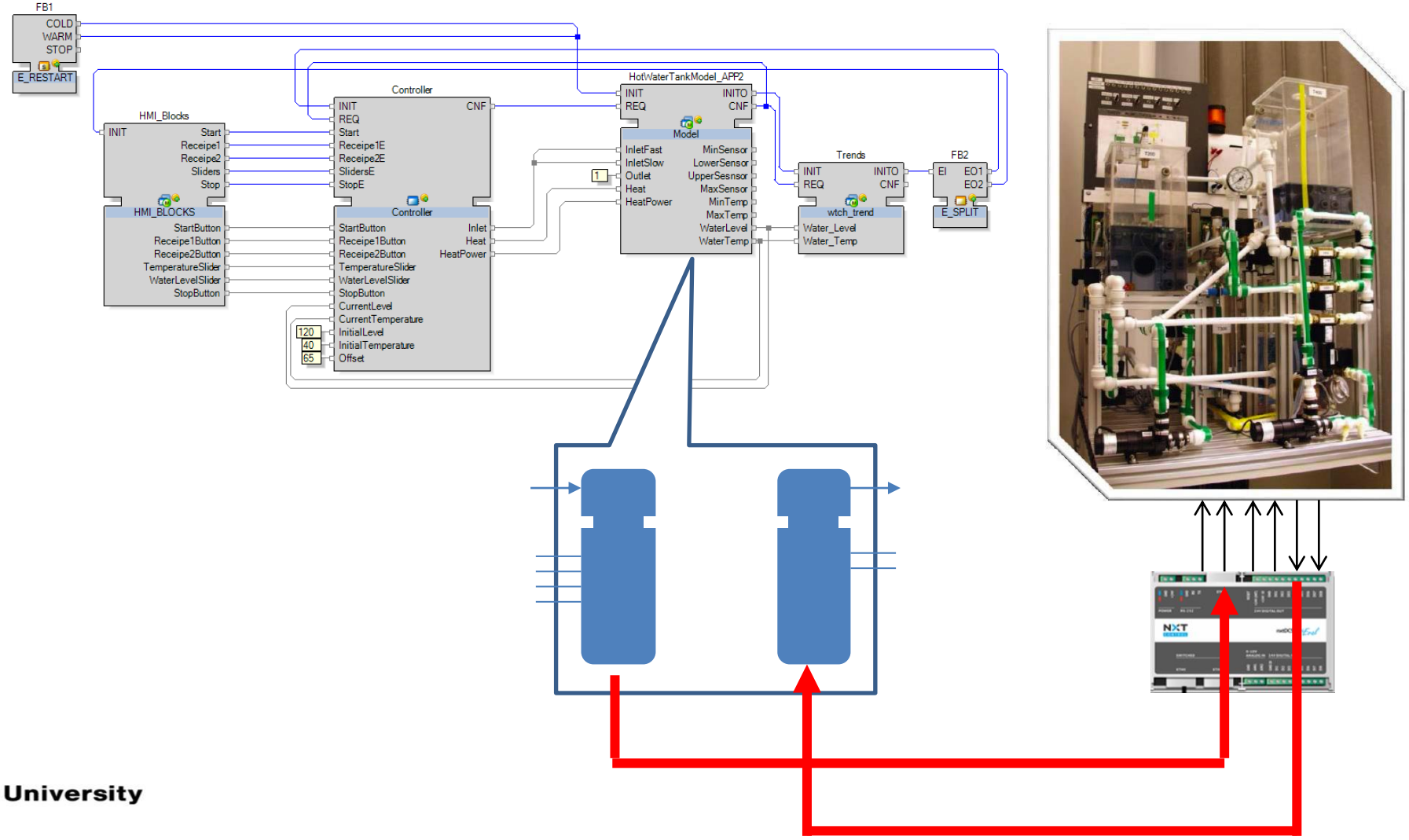
$$X_{OUT} = -(K_p \cdot e + \int \frac{e}{T_R} dt + T_d \frac{de}{dt})$$

e=PV-SP;

- The manipulated variable XOUT, the process variable PV and the proportionality constant KP are of type REAL, as are the internal terms ERROR, ITERM and ETERM.
- The inputs PV, SP and X0 are assumed to be limited to the range 0 to 100 per cent of full scale, and the output XOUT is limited to the same range.

How to deploy this code to control a real Tank?

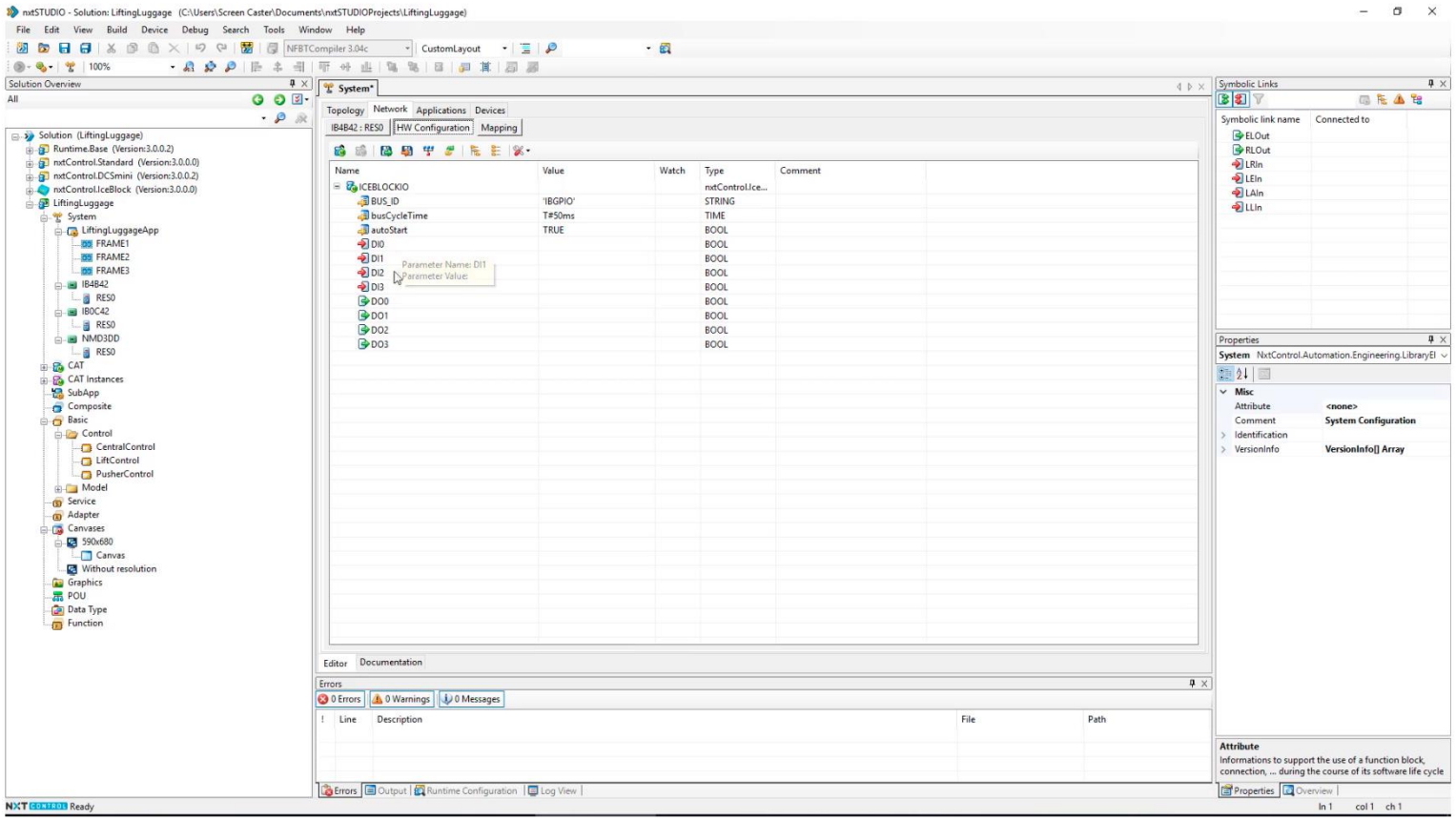
Substitute the plant model with interface to PLC I/O



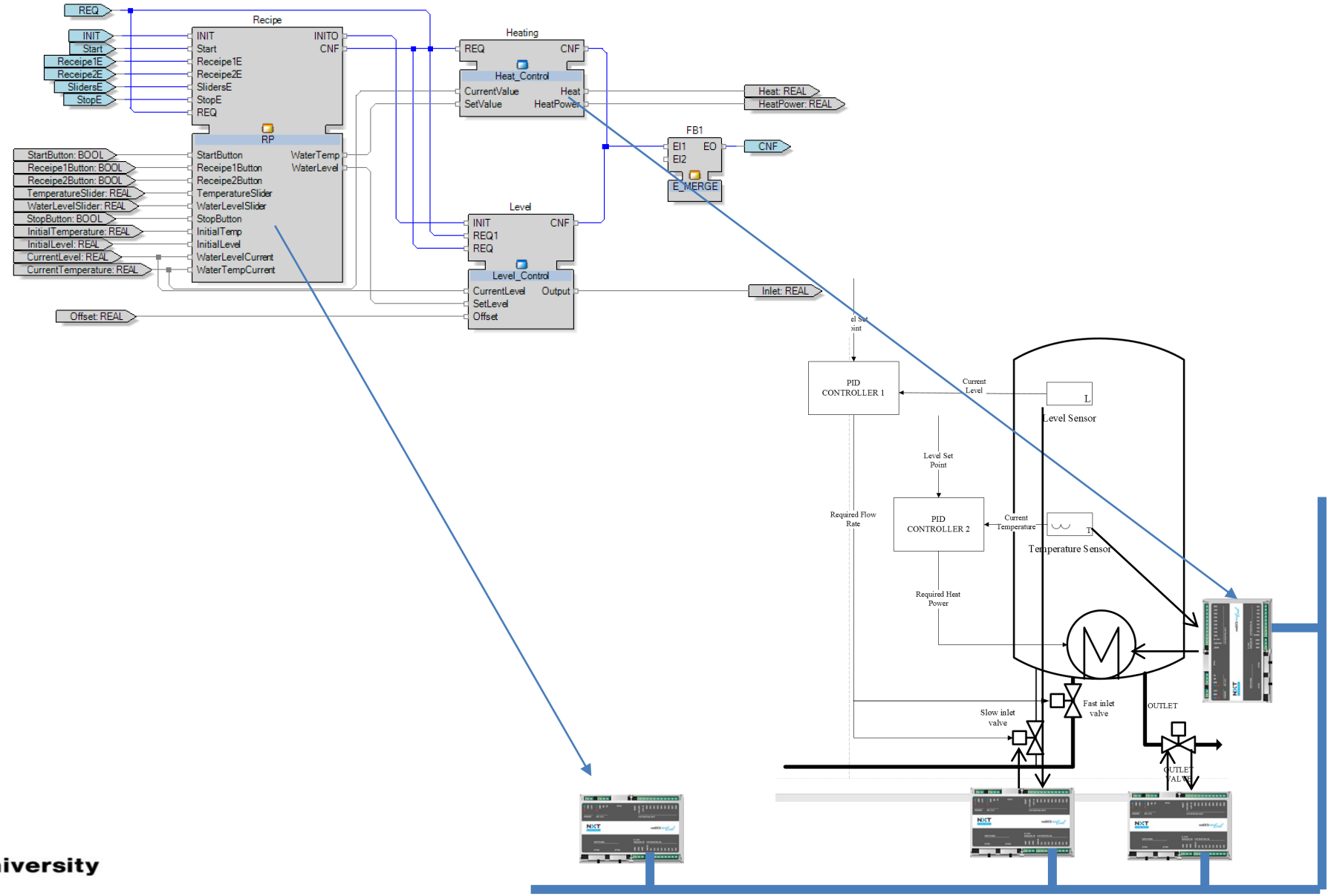
How to connect application with PLC I/Os?

Watch the lesson:

Adding Symbolic links to controller IO's and IceBlock HW configuration



Distributed Control System



Aalto University

Object-based modular design: EnAS lab demo

Assembly system consists of

six conveyors

pallet

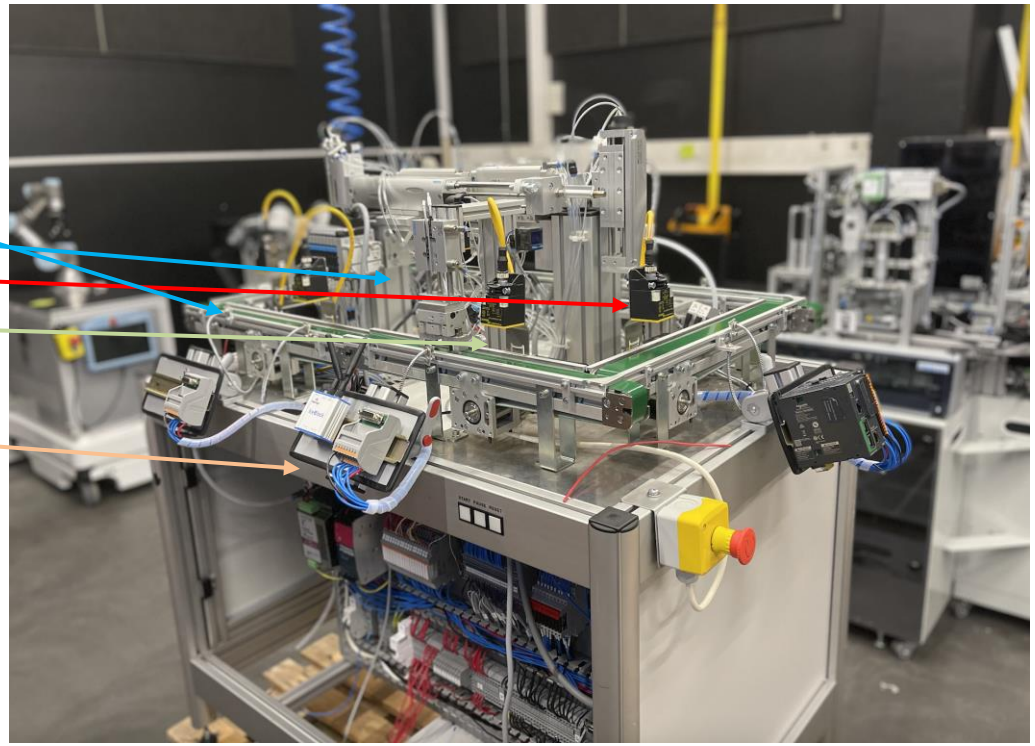
two identical jack stations

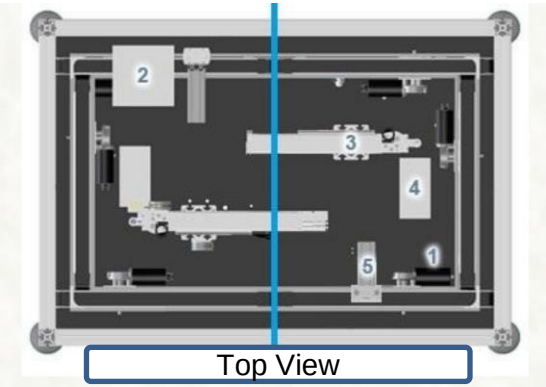
Two sledges

two grippers

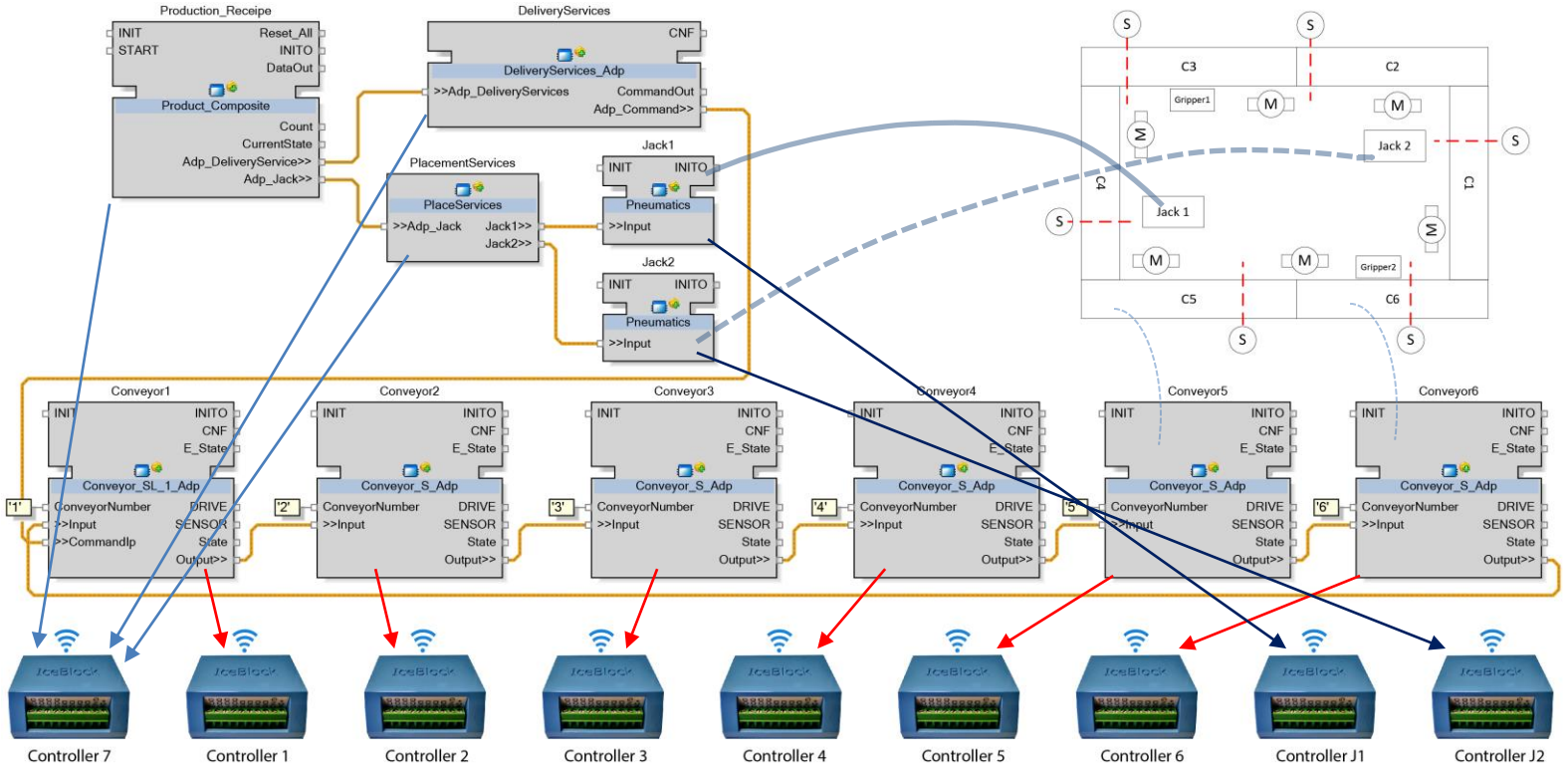
ten sensors.

Controlled using seven PLCs



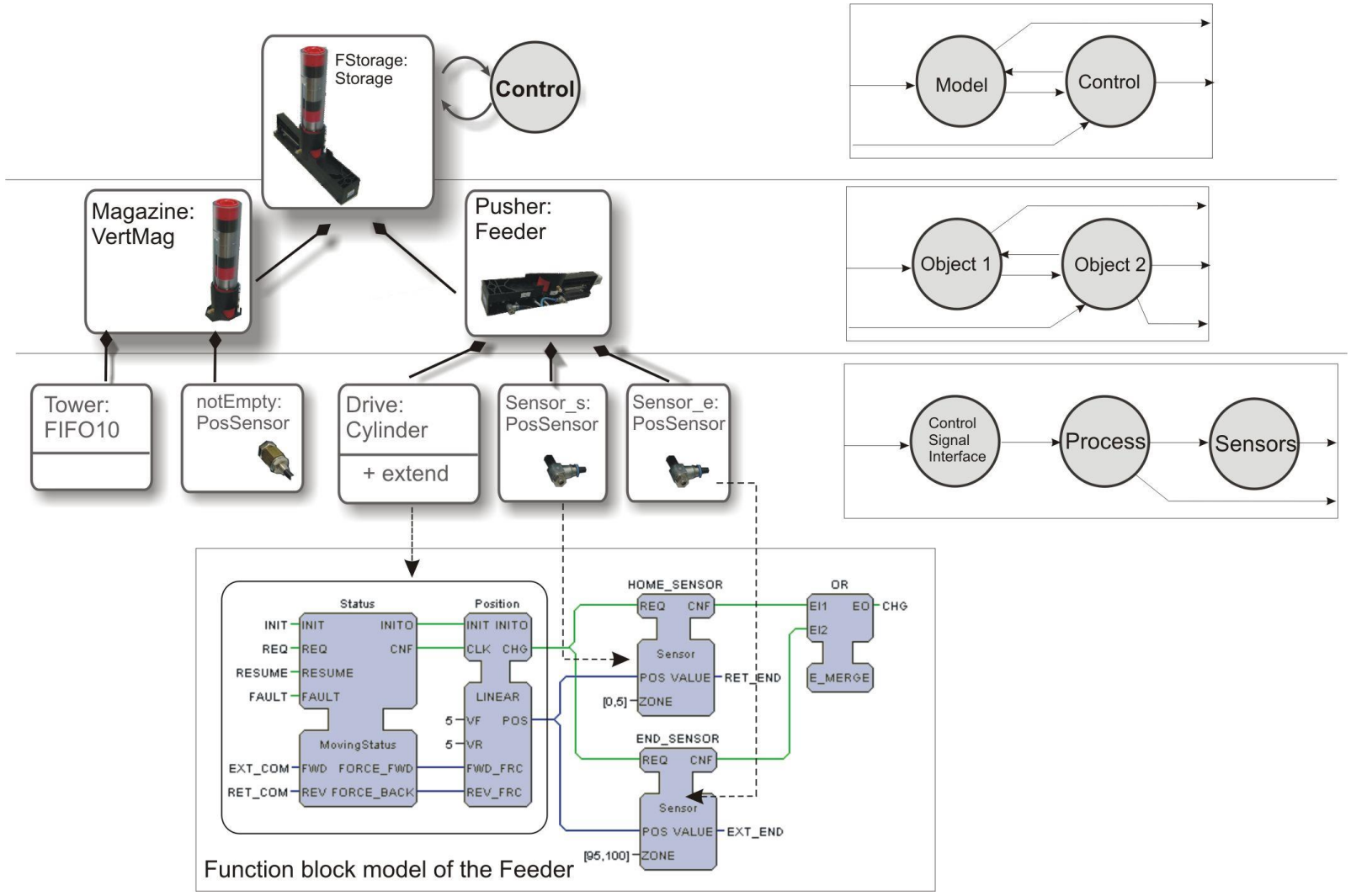


EnAS control application



Aalto University
School of Electrical
Engineering

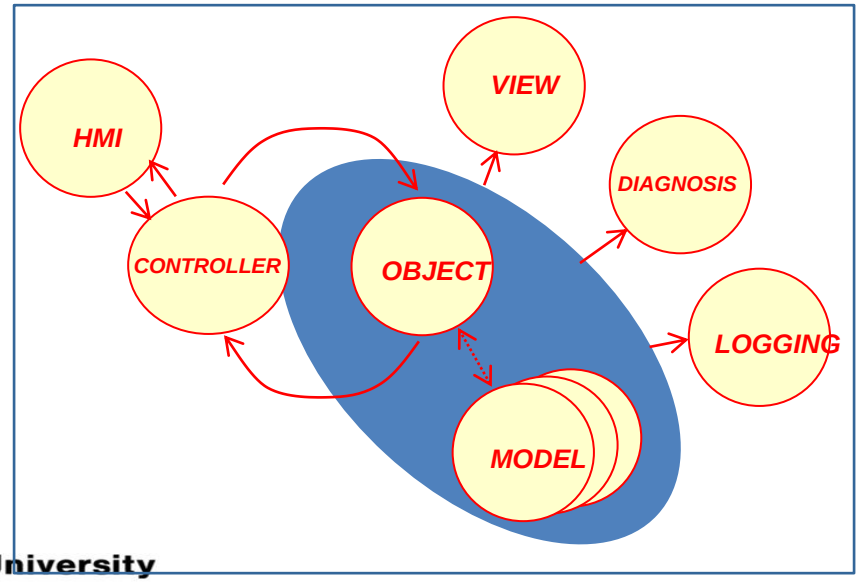
Hierarchical Composition

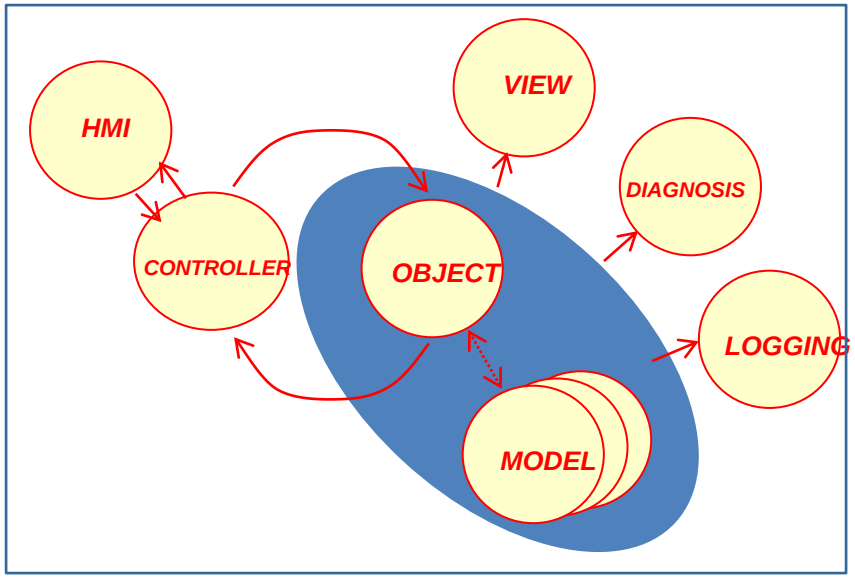


Function block model of the Feeder

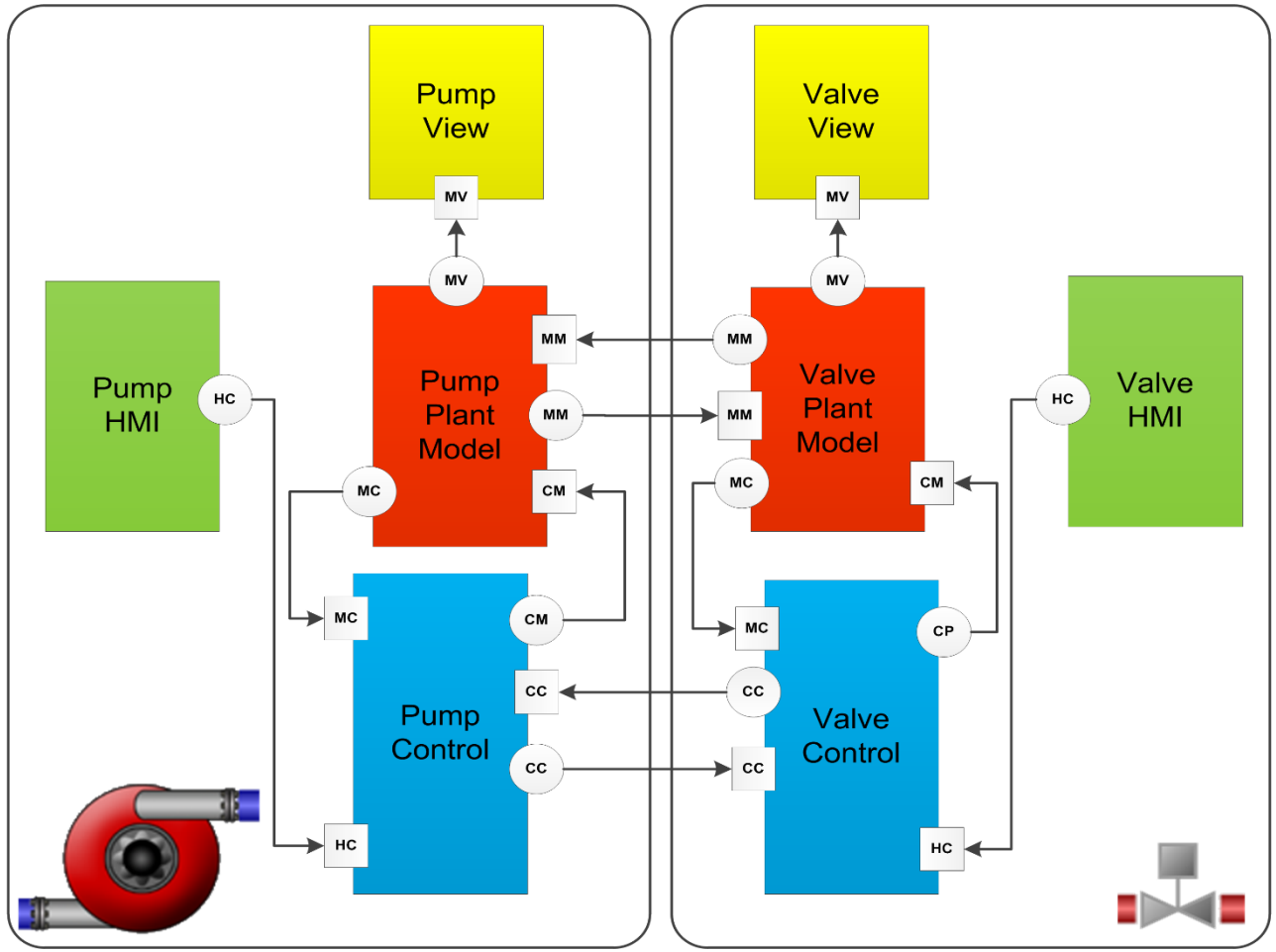
How to apply Model-View-Control Pattern when integrating objects ?

?

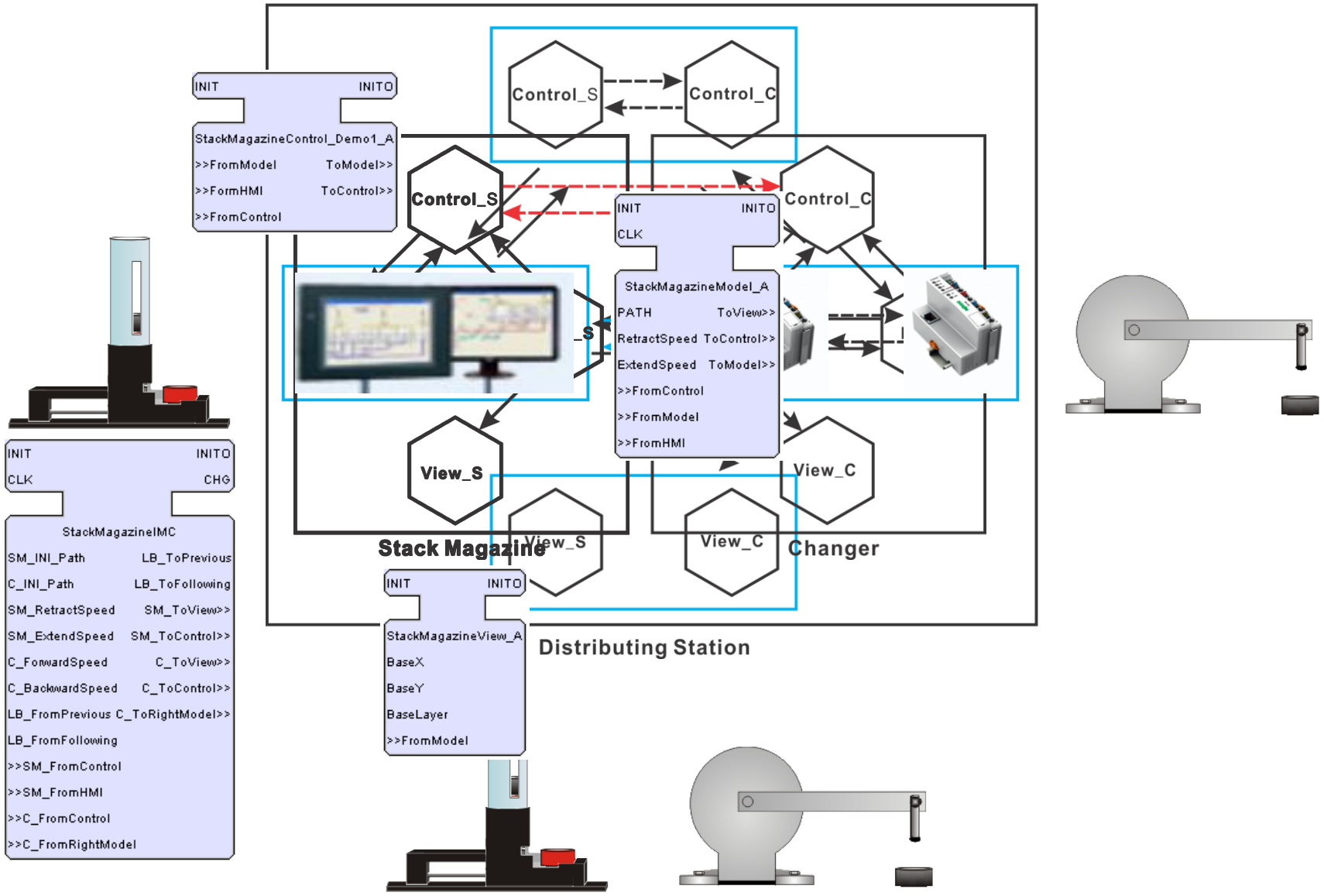




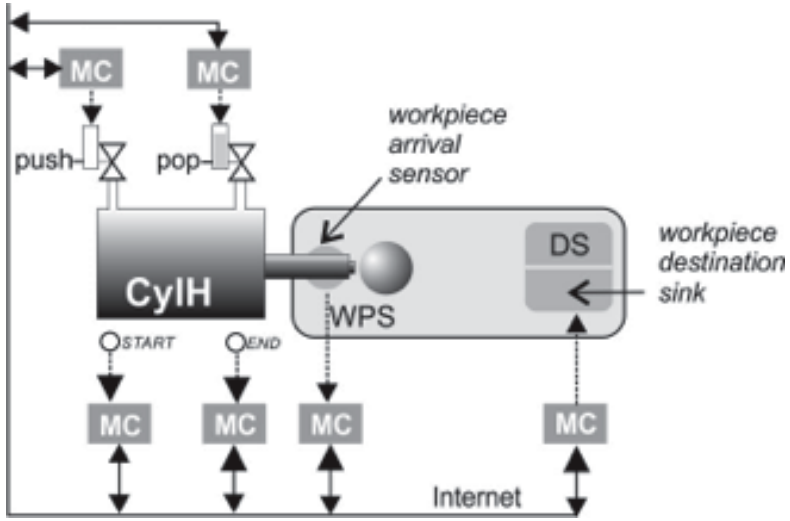
Block Diagram Modelling: Composition

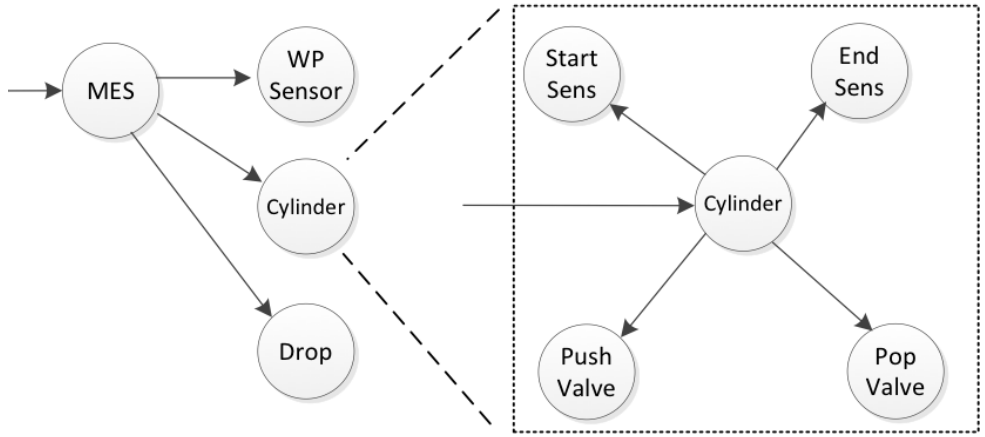


Model-View-Control-HMI Design Pattern

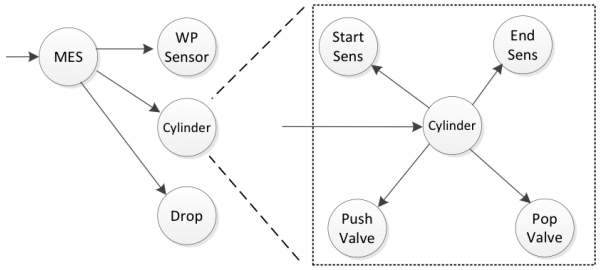


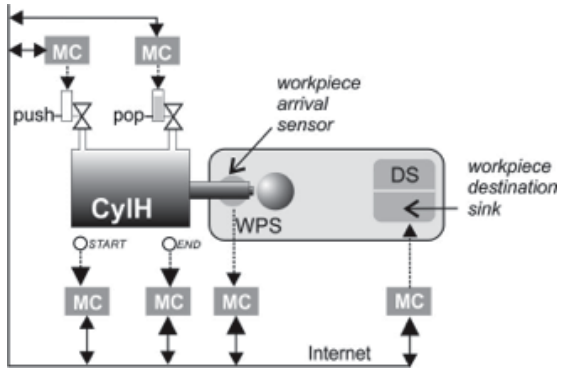
Service-oriented Architecture in Automation

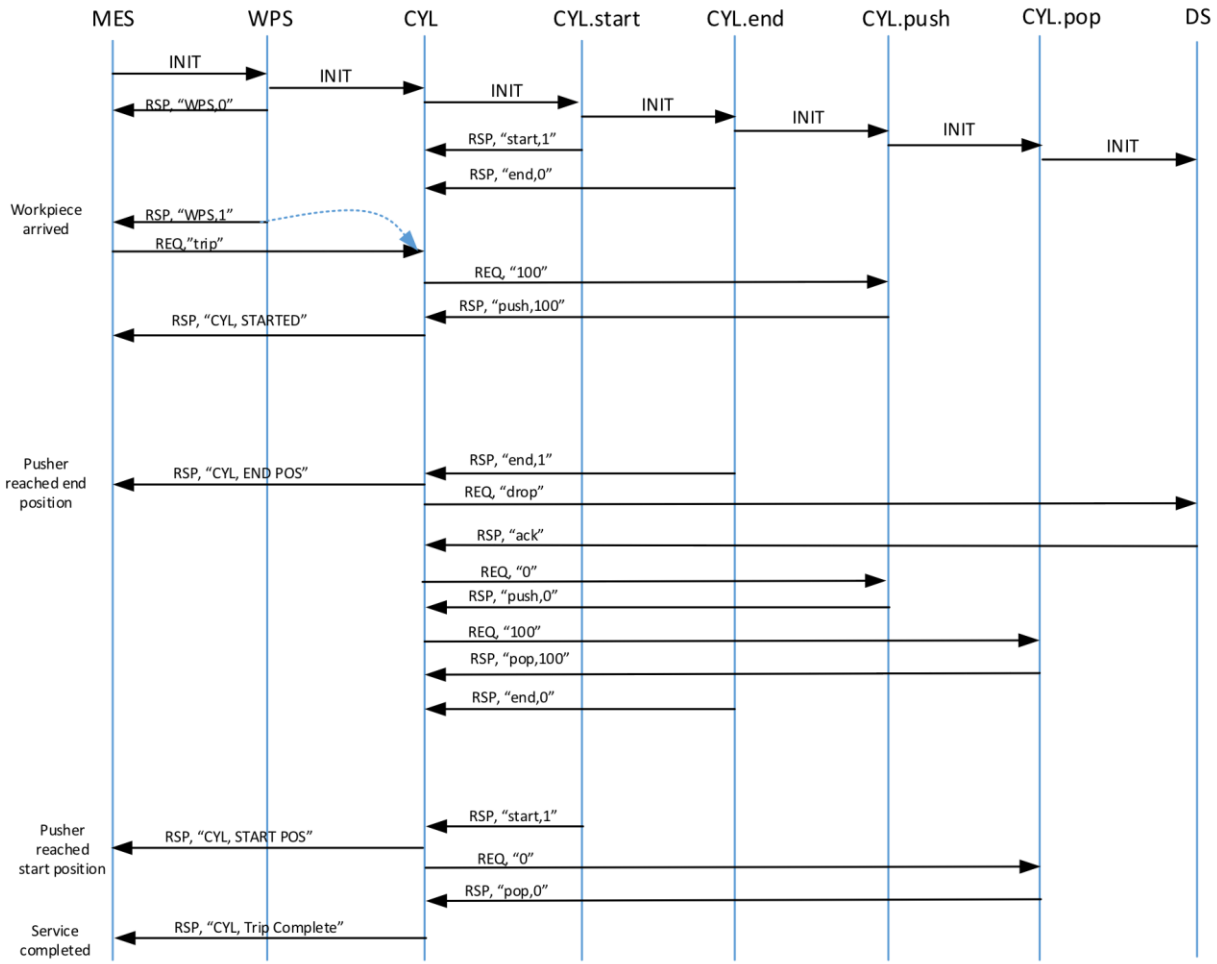




Message exchange between services

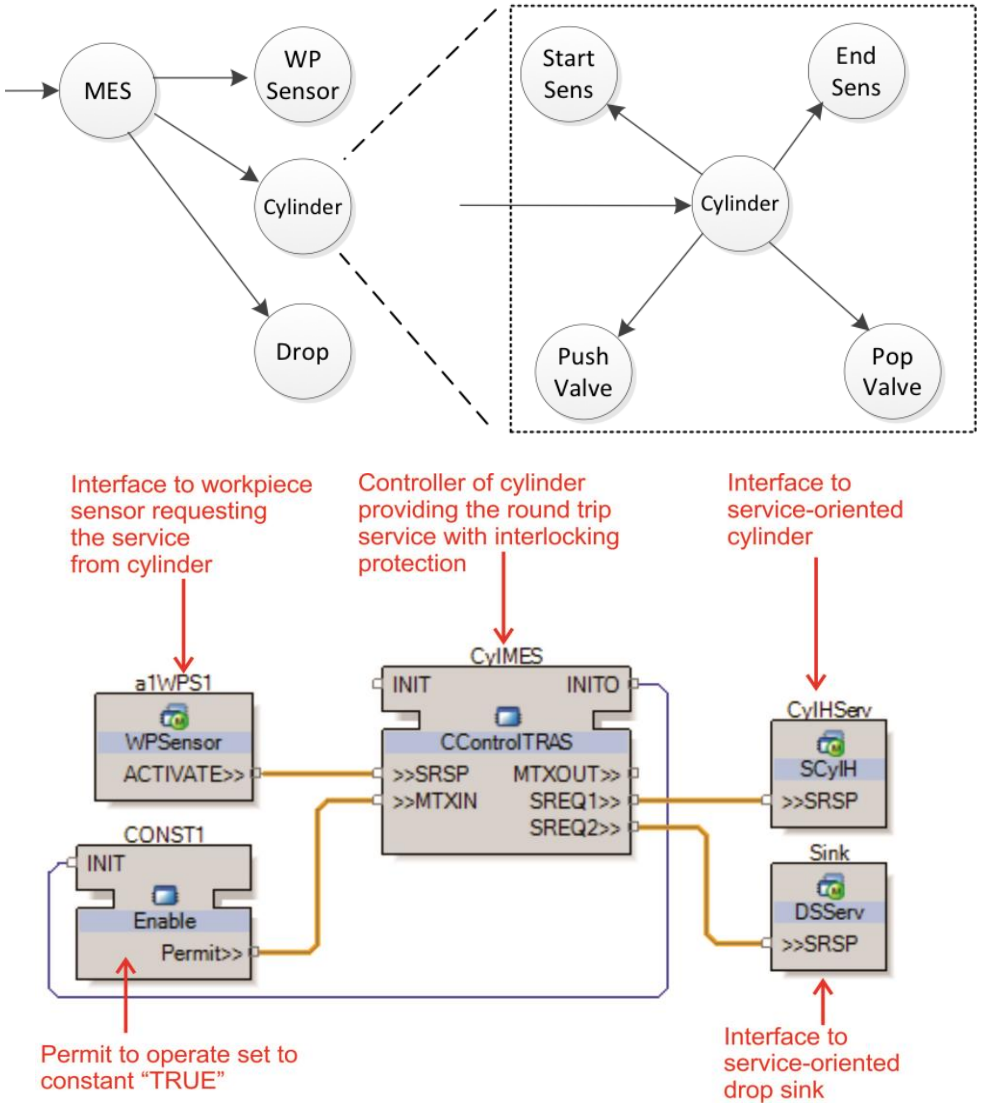




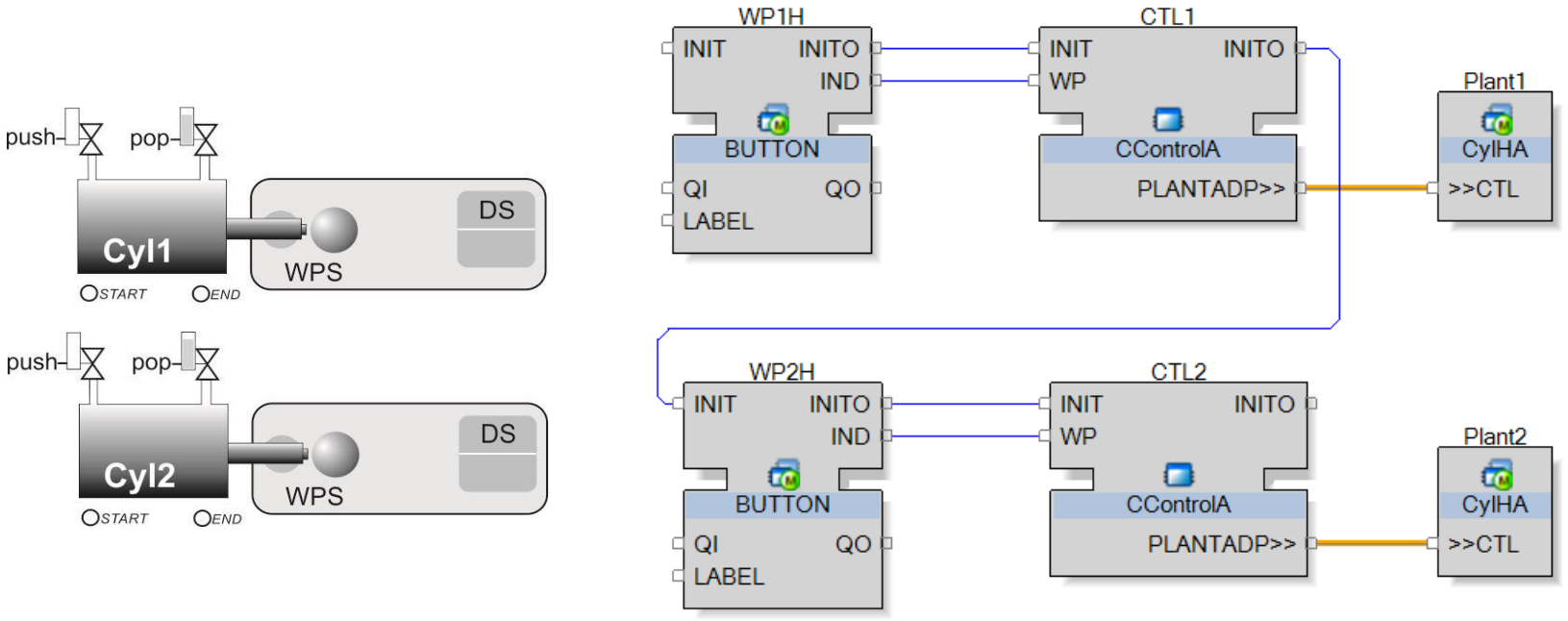


Aalto University

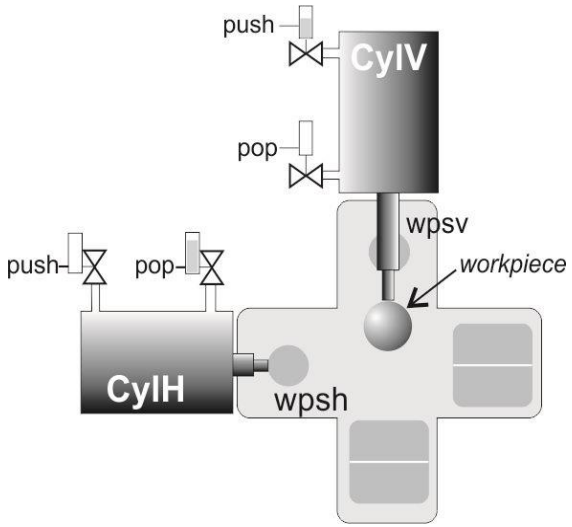
Implementation in function blocks

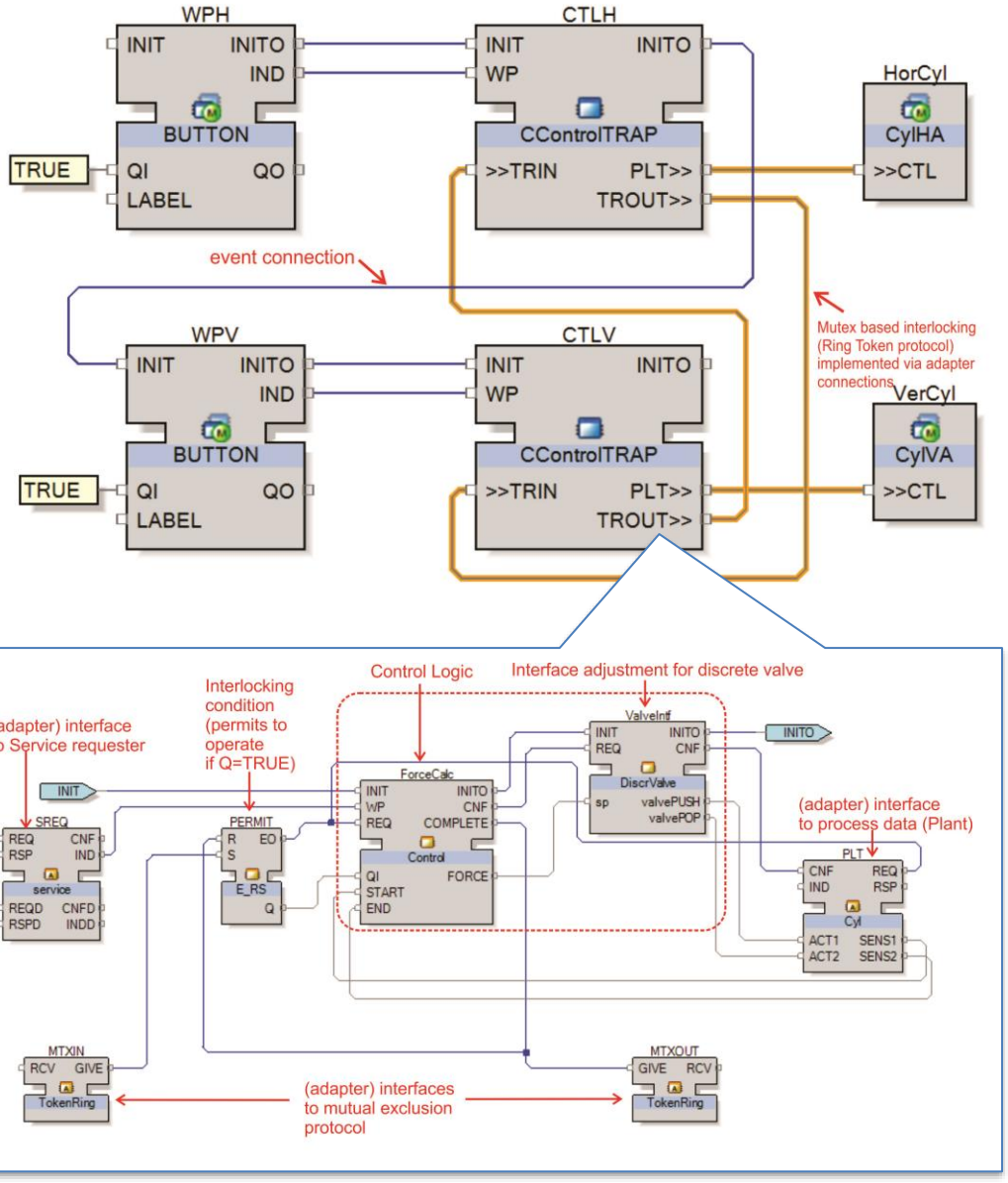


Two independent processes

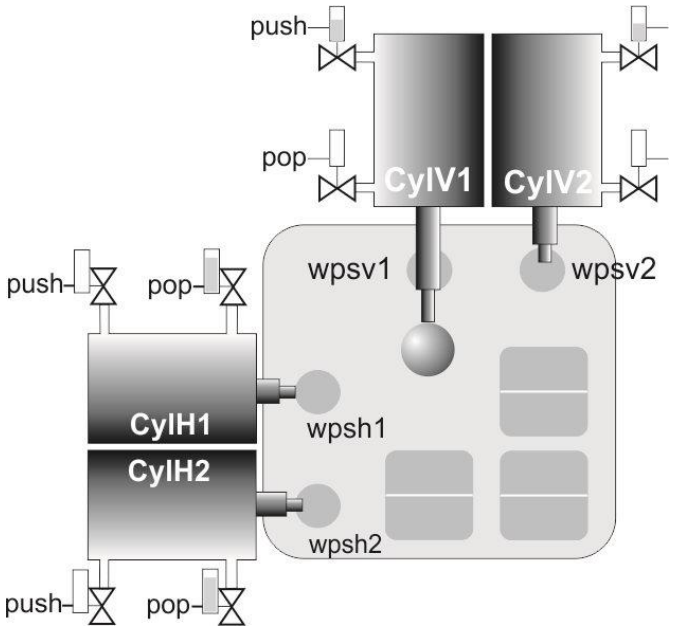


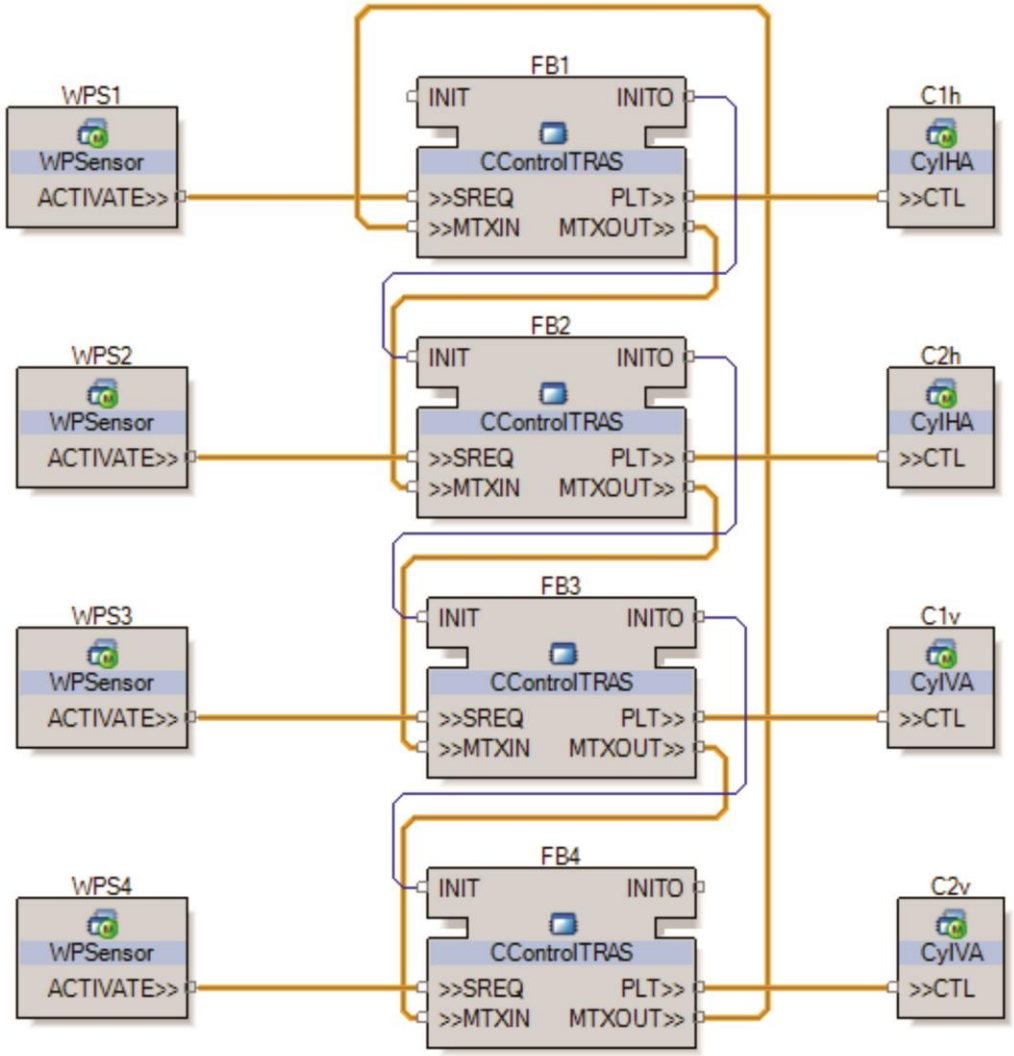
Mutual Exclusion



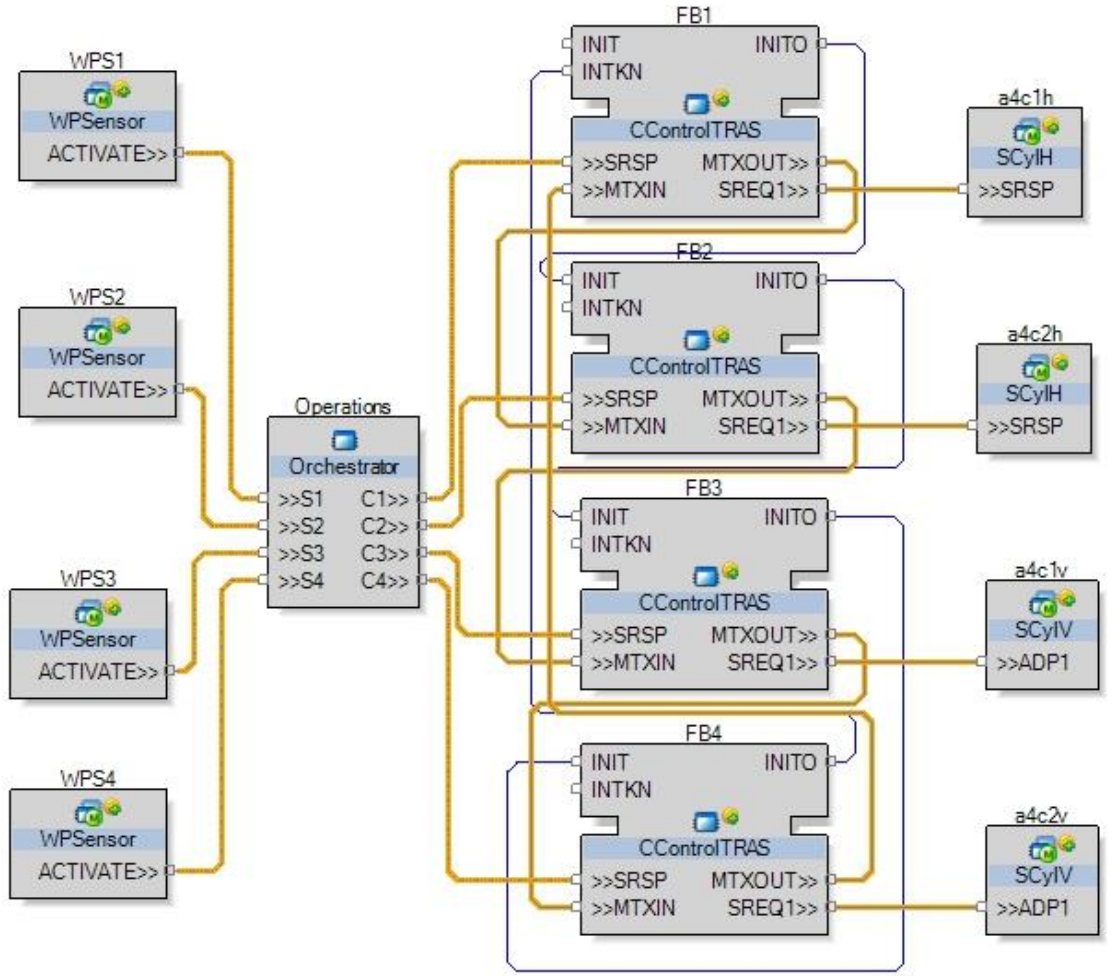


Four processes





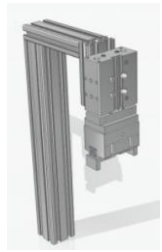
Orchestration

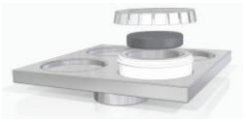


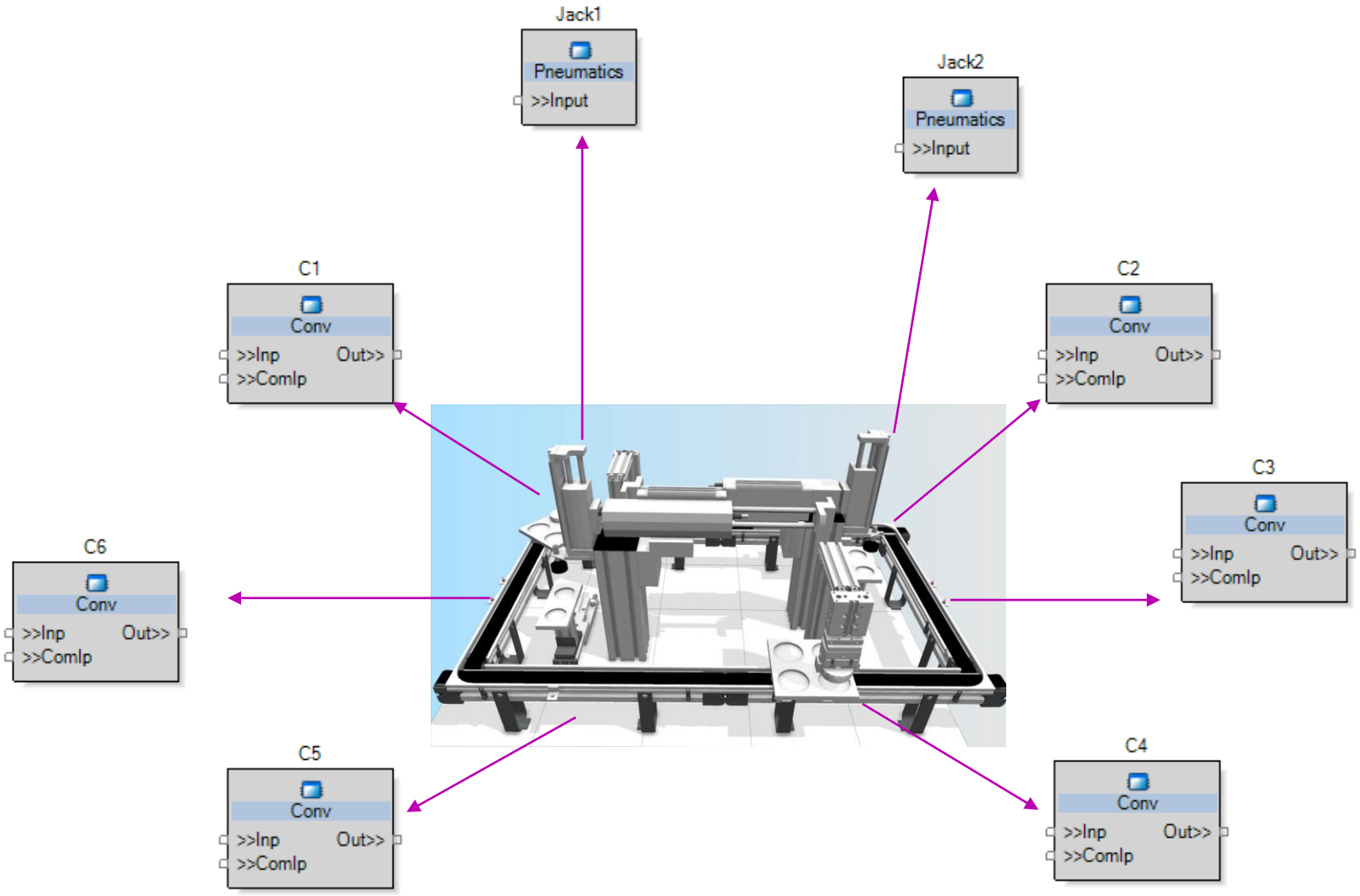
Example of SOA application: EnAS

Modular Mechatronic Software Engineering

Each function block type corresponds to a mechatronic component type.

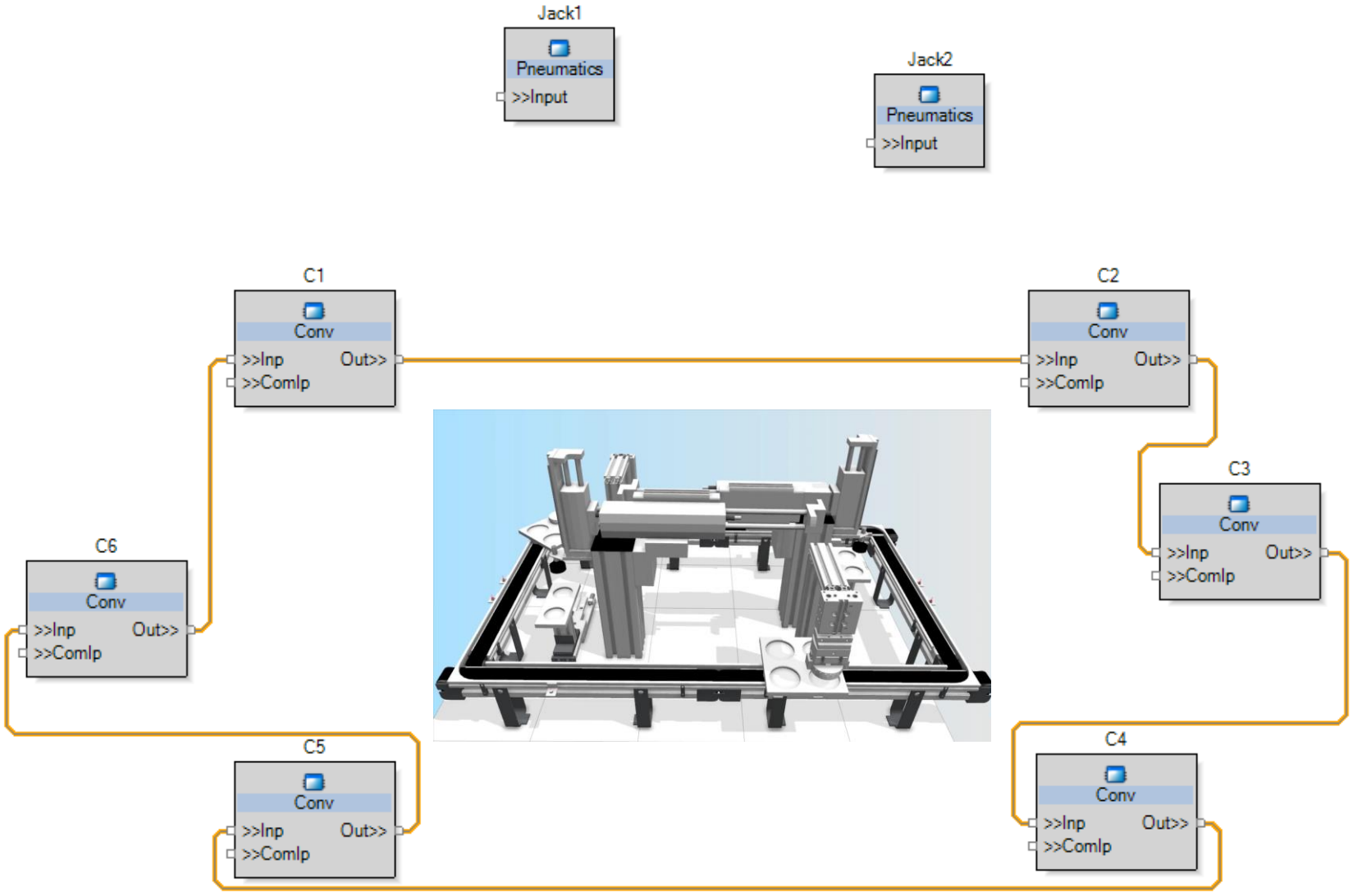




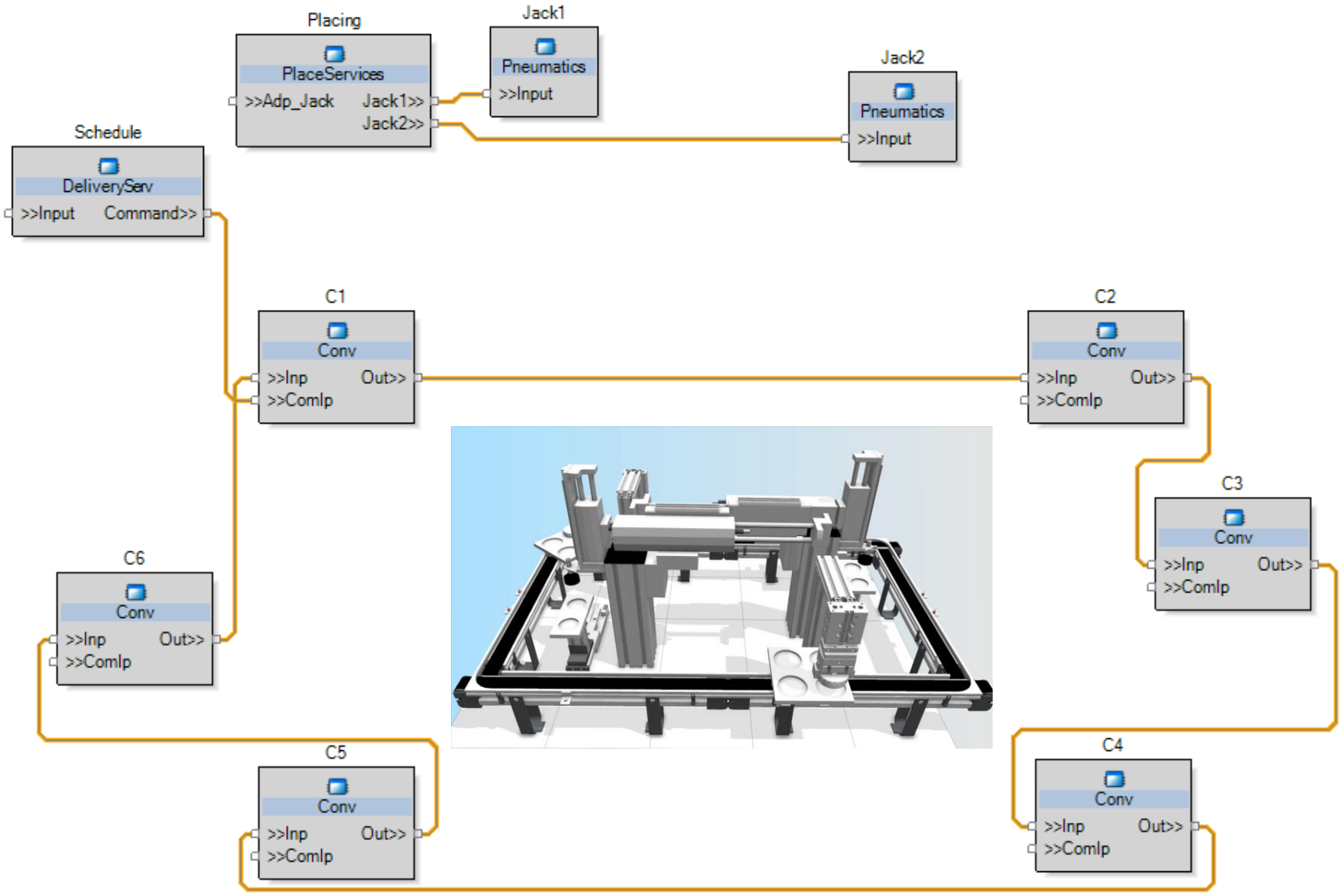


Each function block typeimplements basic control services for the mechatronic component.

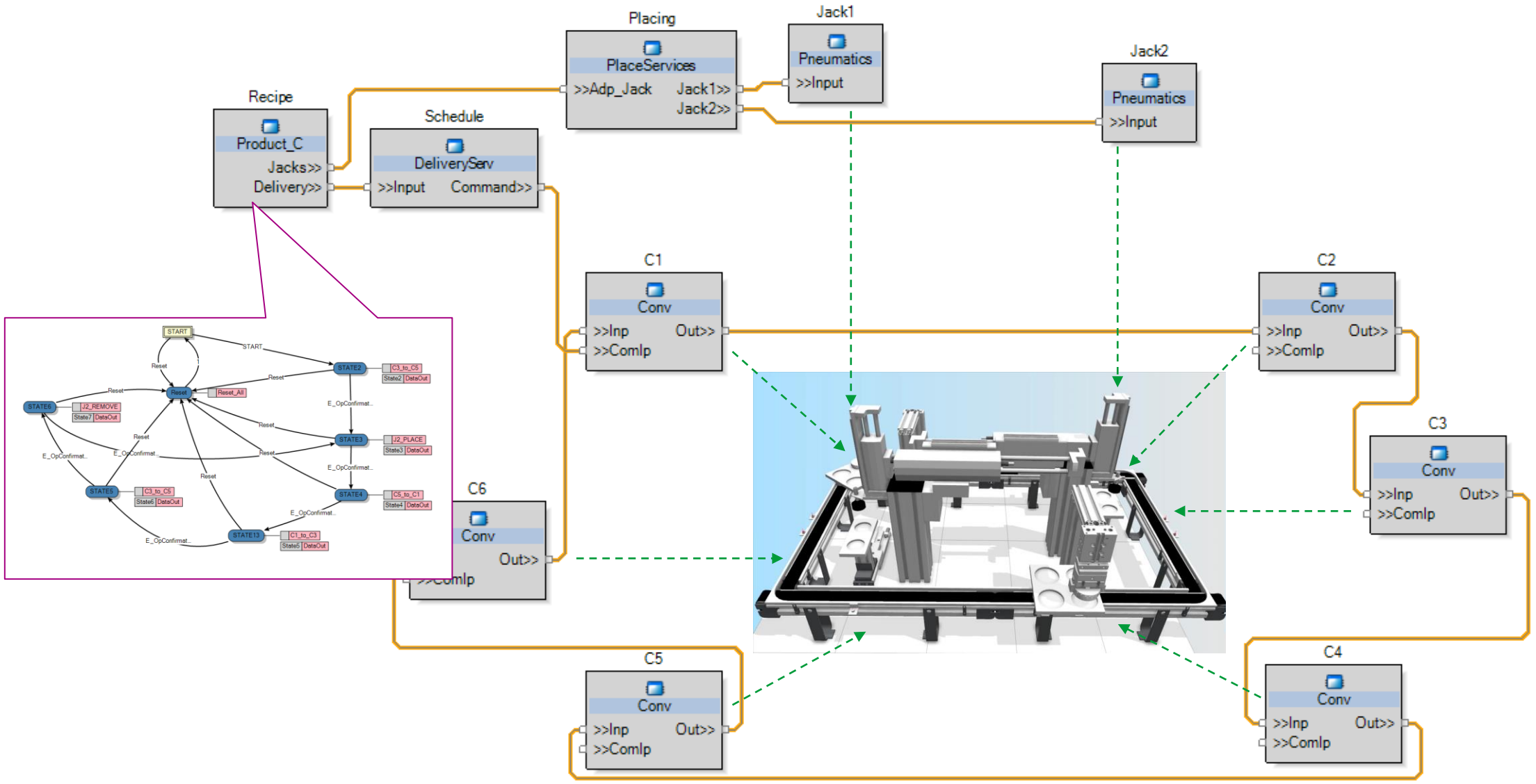
Programming with Function Blocks



Programming with Function Blocks



Programming with Function Blocks

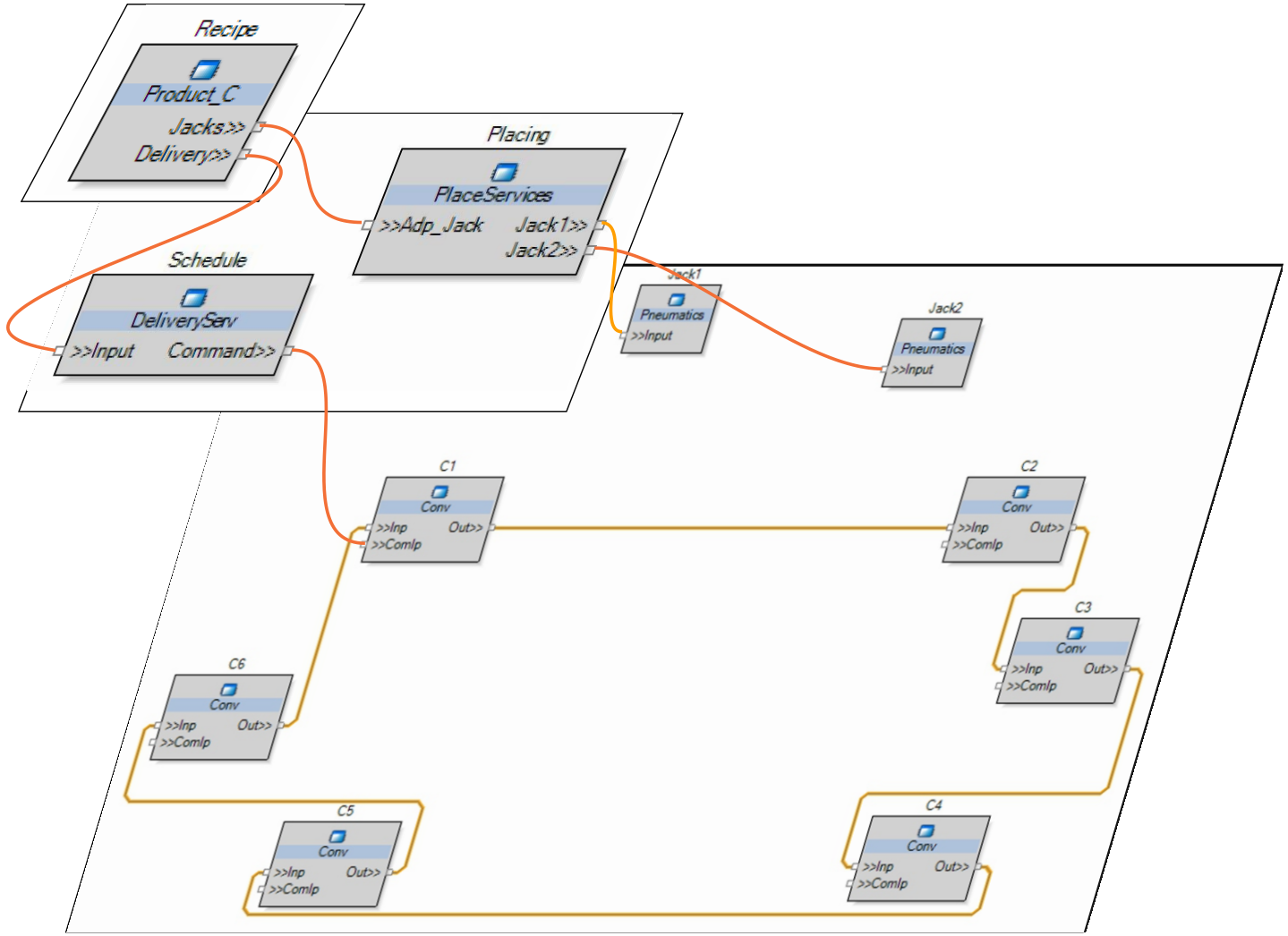


Layered services architecture

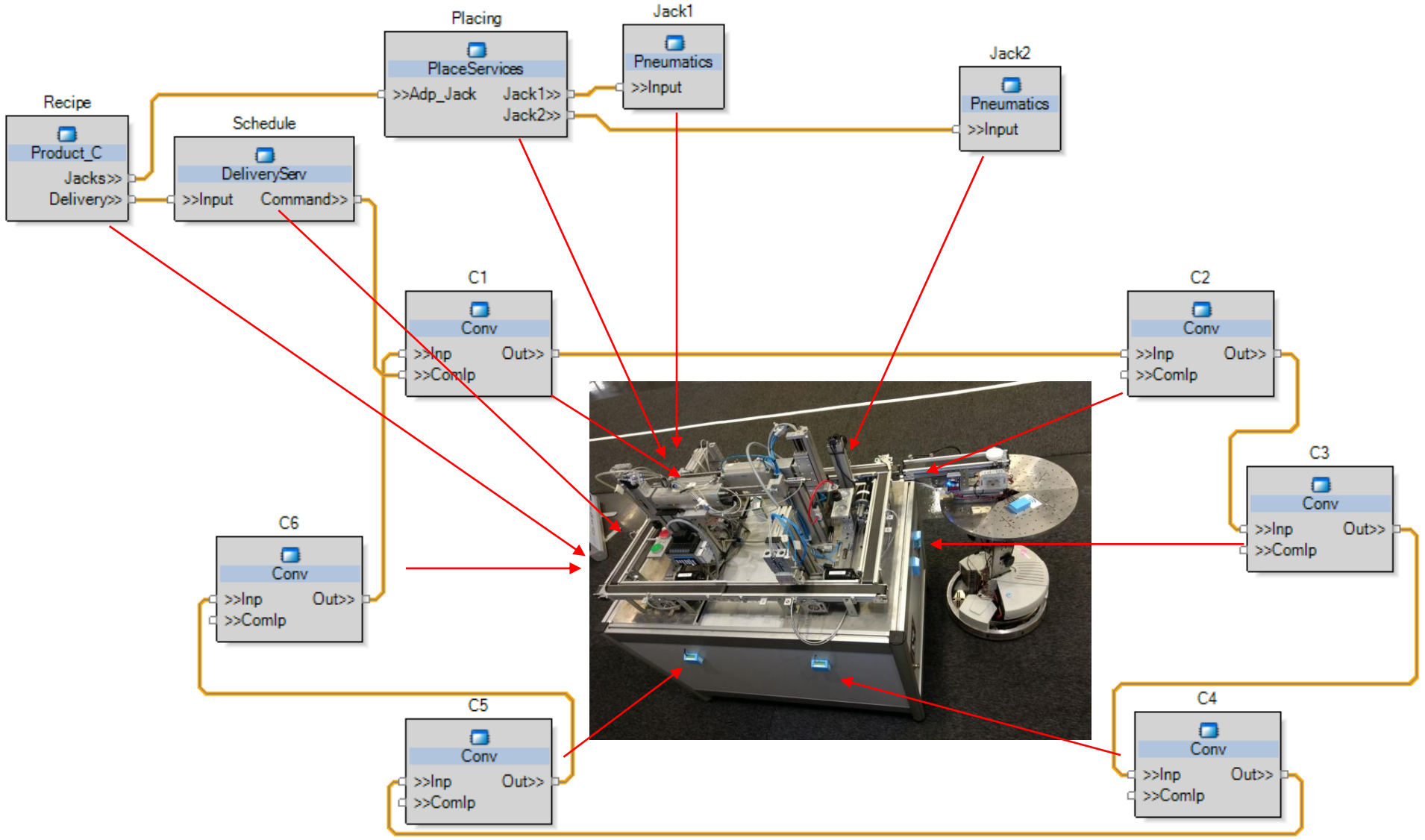
Product description layer

Planning services layer

Execution services layer



Distributed deployment magic



What to remember?

- How to control continuous process with PLC?
- Batch process: combination of discrete steps (recipe) and continuous processes.
- Object-oriented architecture
  - Benefits: re-use of code, flexibility
- Service-oriented architecture
  - Benefits: re-use of services, flexibility of loose coupling, relience on Cloud and Fog services