Piping & Instrumentation Domain Model

Assignment for LWC 2012

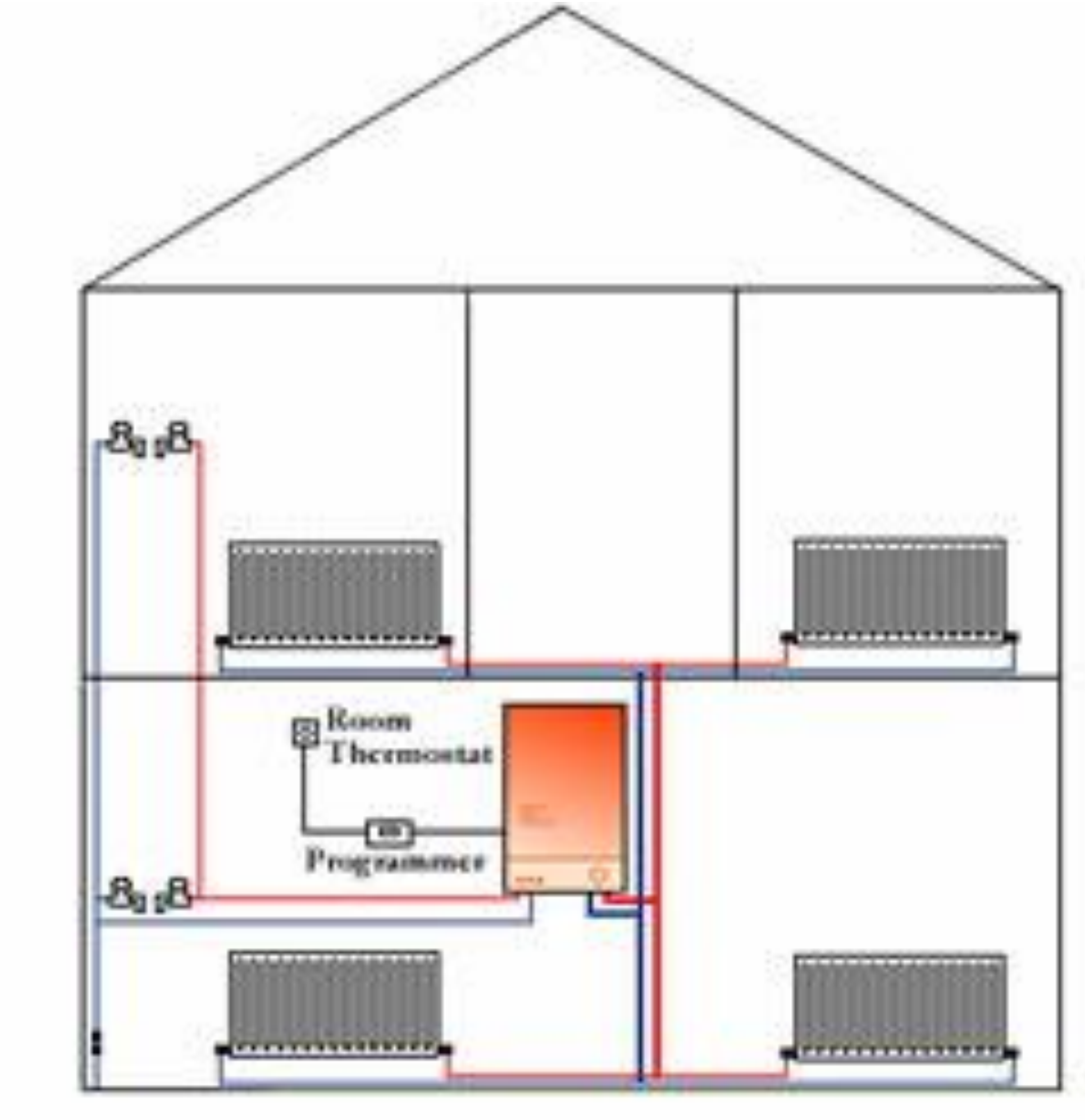
1-11-2011

Paul Zenden
(paul.zenden@sioux.eu)

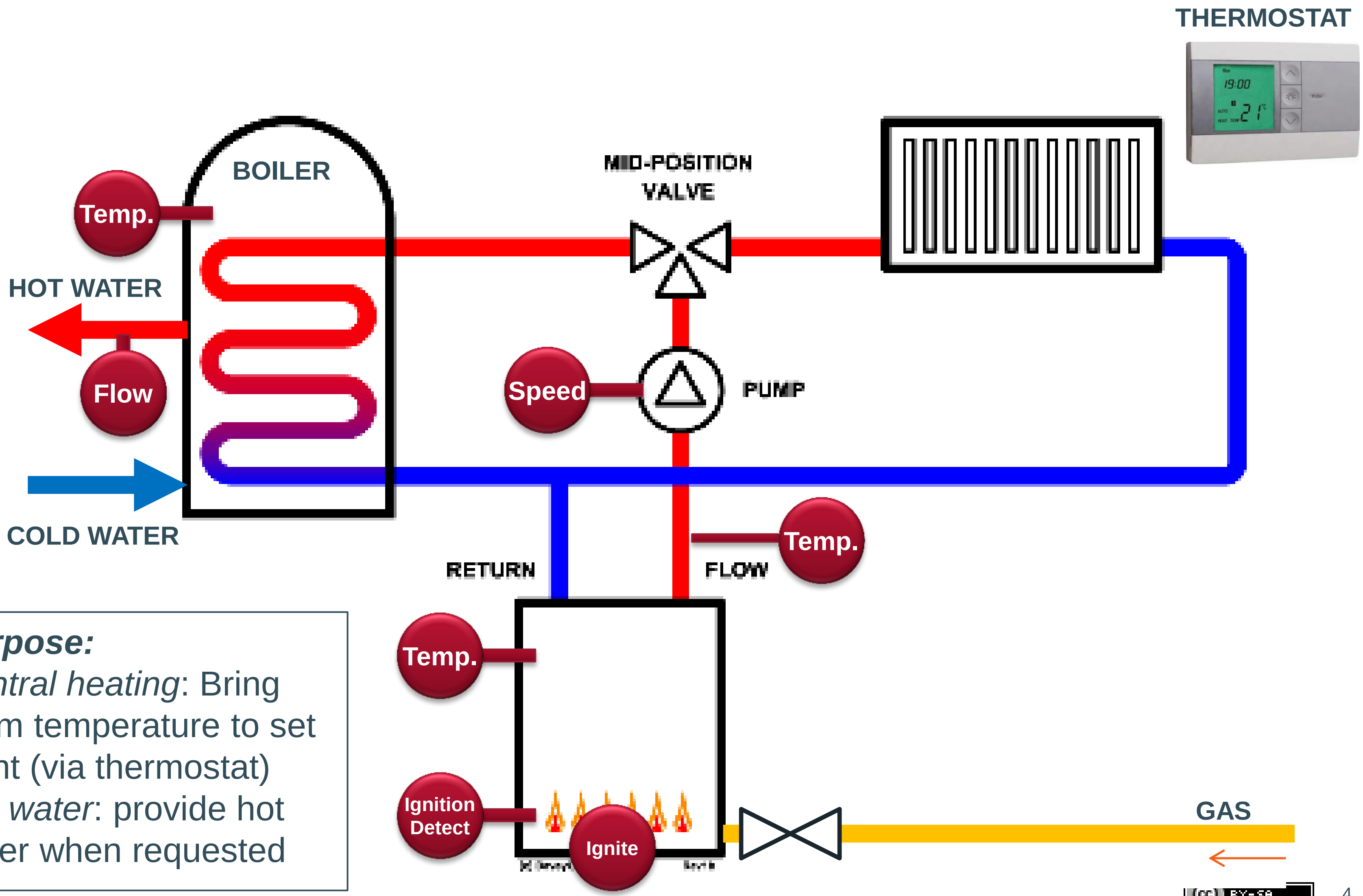
Goal

- Use/Combine models based on *multiple meta-models*
- Apply MDSD in a *non-software domain*
  - But controlled with software
- Focus is on *capabilities of tooling* not on the domain itself
  - Limited but useful subset of domain concepts

Case: Central Heating (1)



Case: Central Heating (2)



Purpose:
Central heating: Bring room temperature to set point (via thermostat)
Hot water: provide hot water when requested

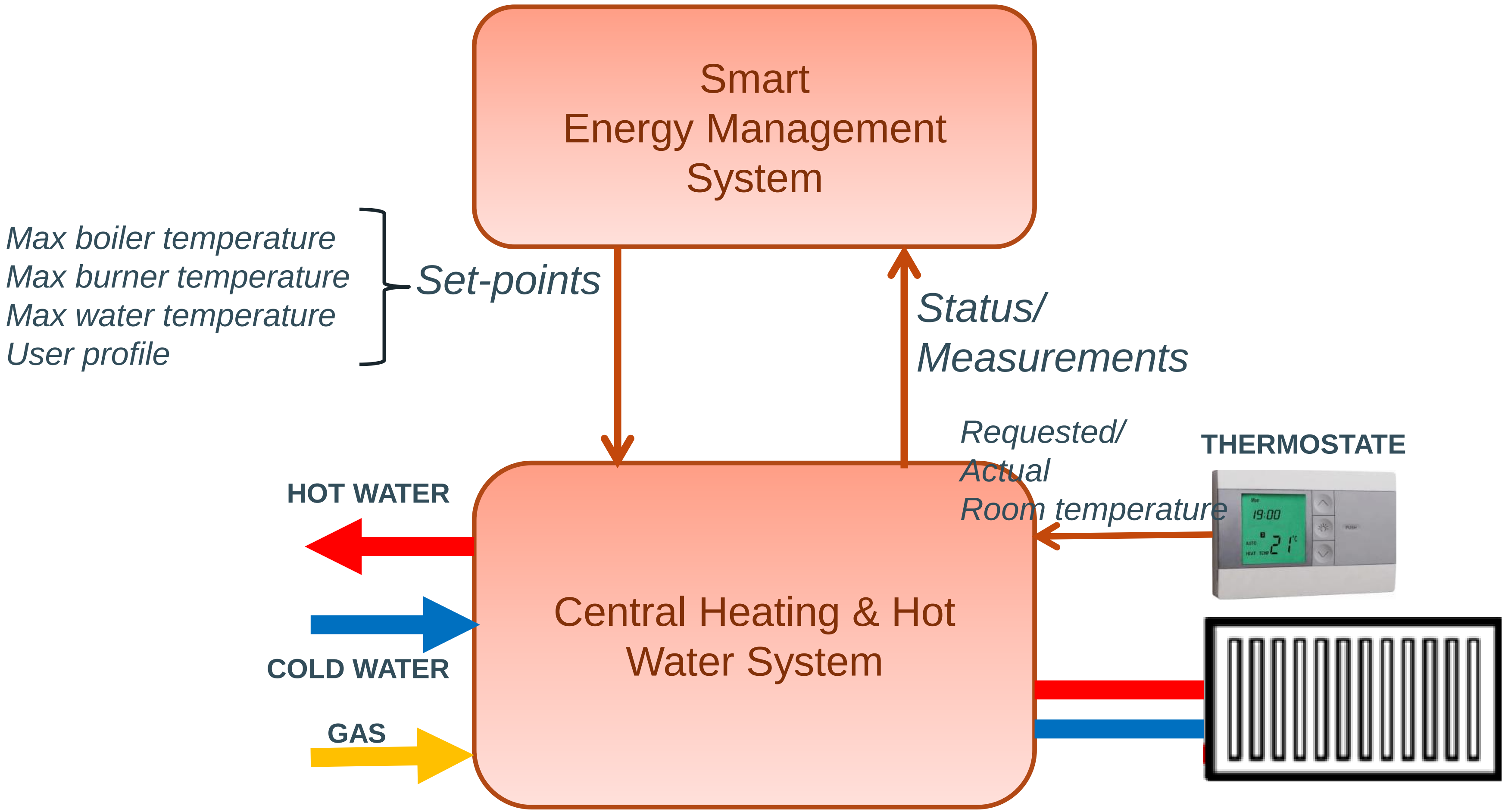
Case: Central Heating (3) Requirements/Constraints

- Heat up the radiators if the actual room temperature is below the requested temperature.
- Keep the hot water to configured set point (e.g. 95º C)
- To increase efficiency, when heating up, gradually increase the burner heat to its max over a certain time period.
- To increase efficiency, decrease the burner heat when the actual room temperature is in reach of the requested temperature
- The pump must run when the burner is on.
- The pump can be set to on (max speed) or off (no speed).
- The mid-position valve can be in three positions: all flow to boiler, all flow to radiators, or flow going to both boiler and radiators.
- Be able to connect to a ‘Smart Energy Management System’ for:
  - Accepting set-points of maximum temperature of water, burner, etc.
  - Accepting specific user settings (user profile), like rate of heating up, holiday/non-holiday setting, etc.
  - Visualization of the actual status of the central heating system (status of actuators, values of sensors).

Challenge (1)

- Develop a DSL and generator that allows:
  - describing the central heating system as a Piping & Instrumentation system by a domain expert
  - describing the required functionality:
    - behavior
    - interlocks/constraints
  - generating as much as possible of the control software
  - generating a simulation of the central heating system:
    - the control system can be tested without the need of actual plumping.

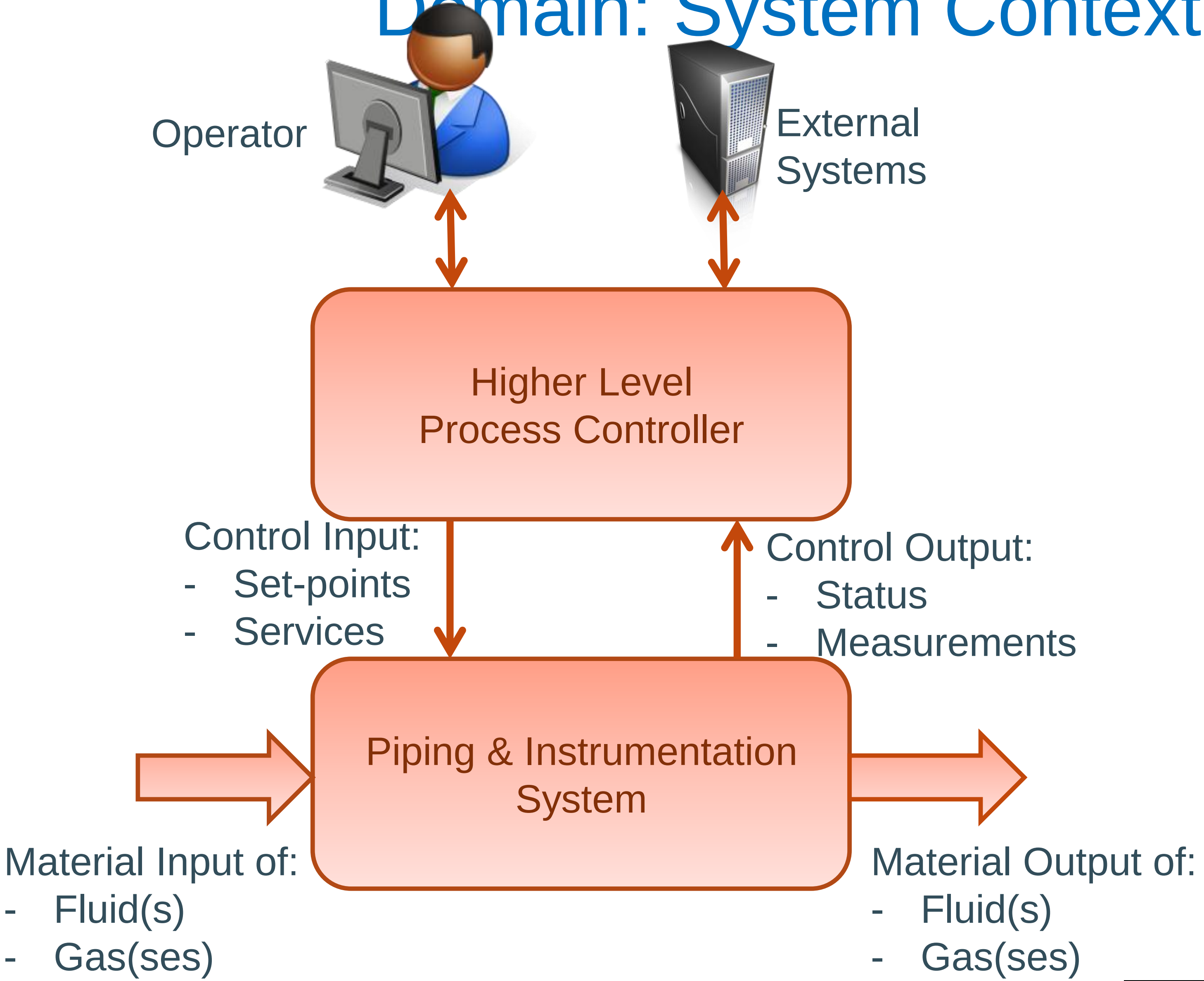
Challenge (2)
System Context



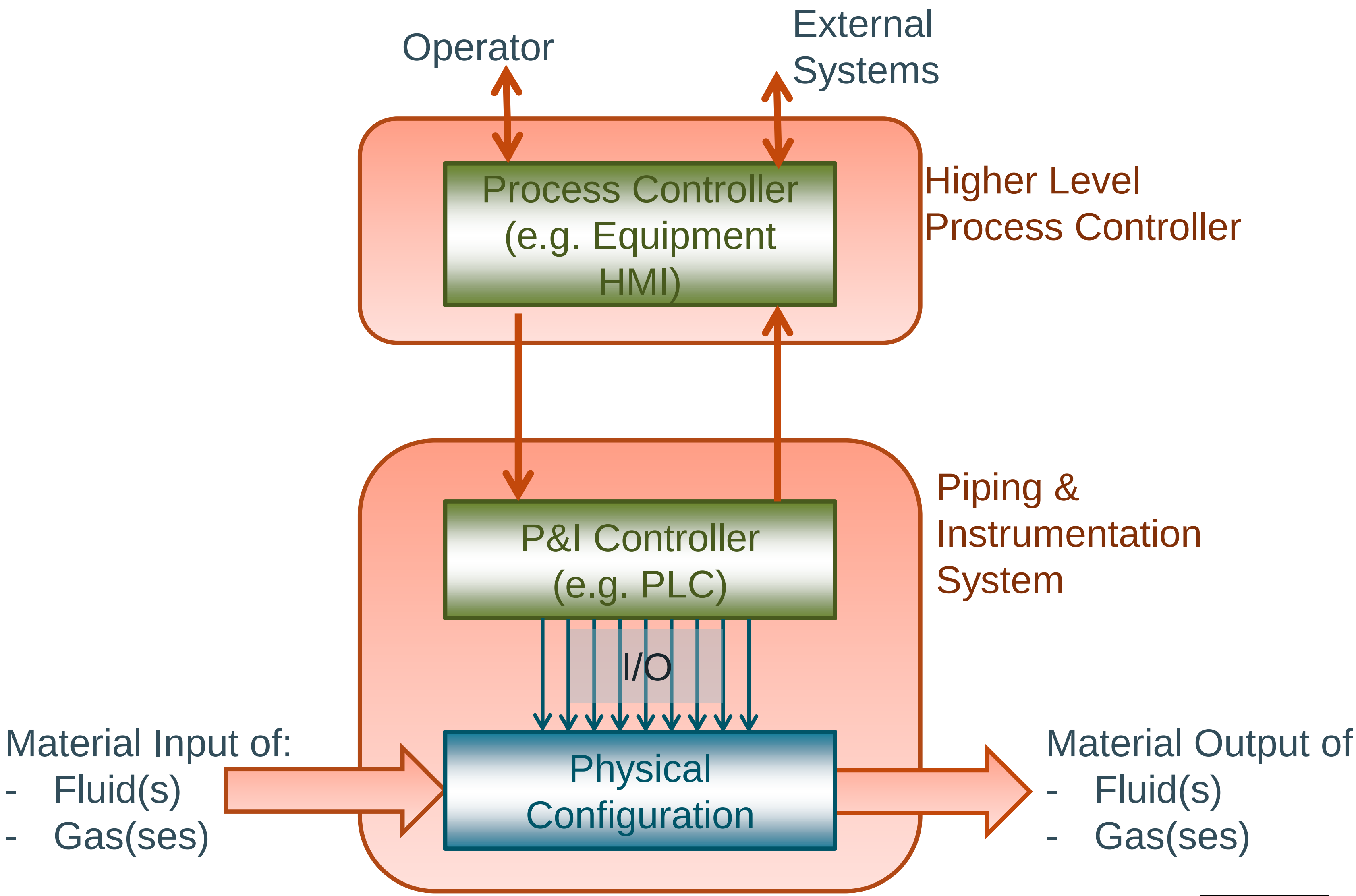
Piping & Instrumentation Domain

- Physical equipment and pipes that handle fluid and/or gas flows
- Subset of element groups:
  - Pipes, Joints, System ends
  - Valves
  - Pumps
  - Vessels
  - Measurement instruments (Sensors)
- Purpose:
  - Define a network of connected domain elements such that it fulfills a dedicated purpose.
- Reference:
  - http://en.wikipedia.org/wiki/Piping_and_instrumentation_diagram

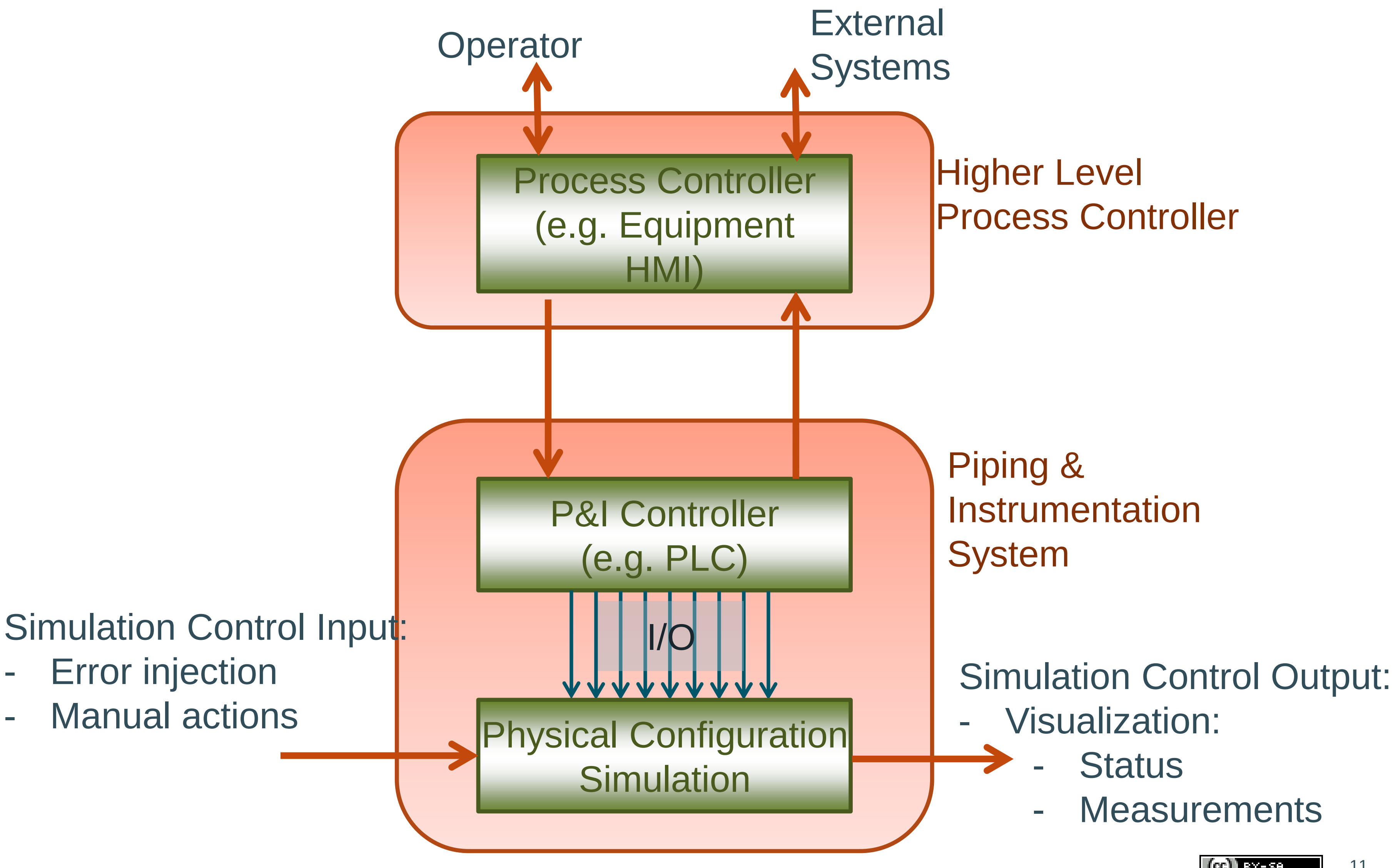
Piping & Instrumentation Domain: System Context (1)



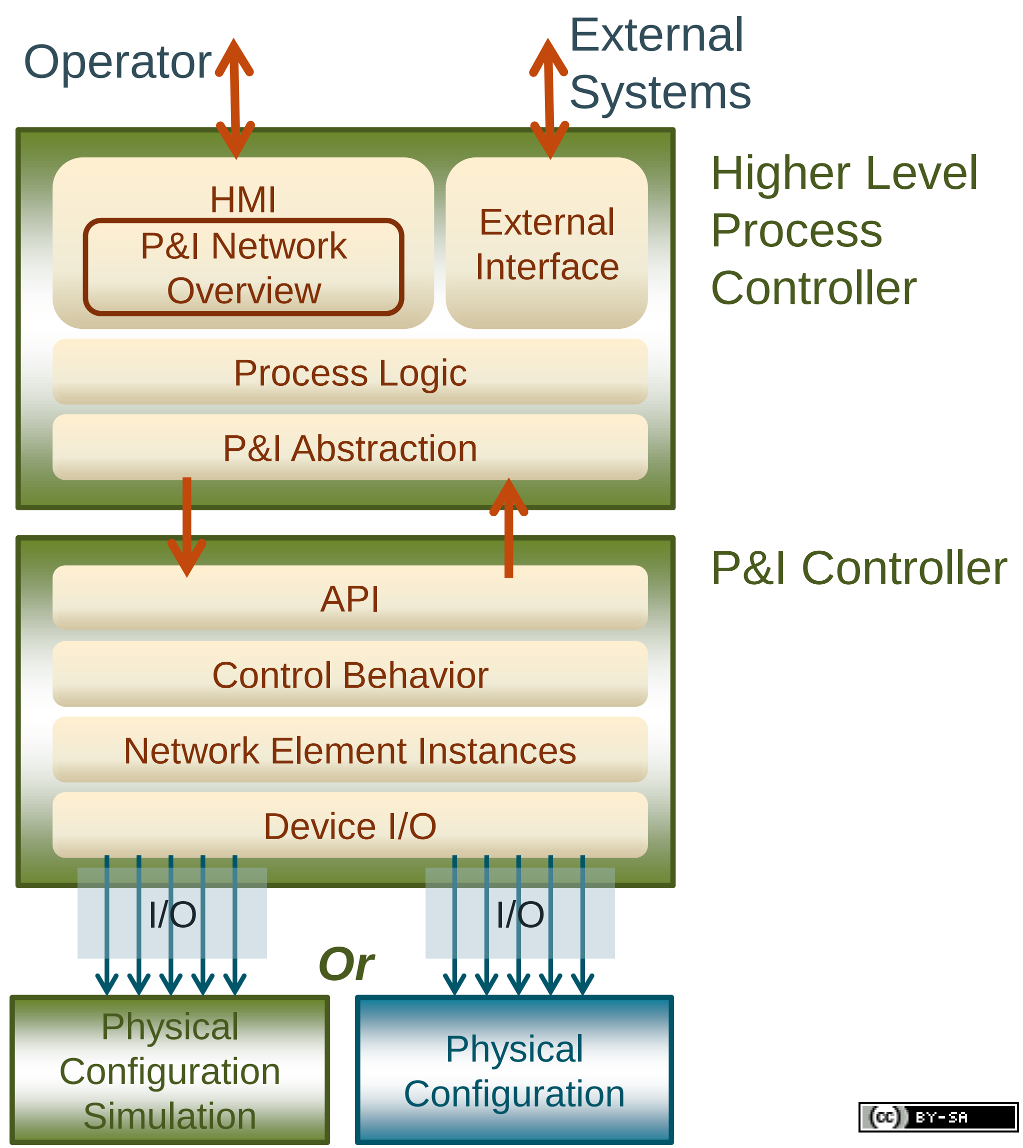
Piping & Instrumentation Domain: System Context (2)



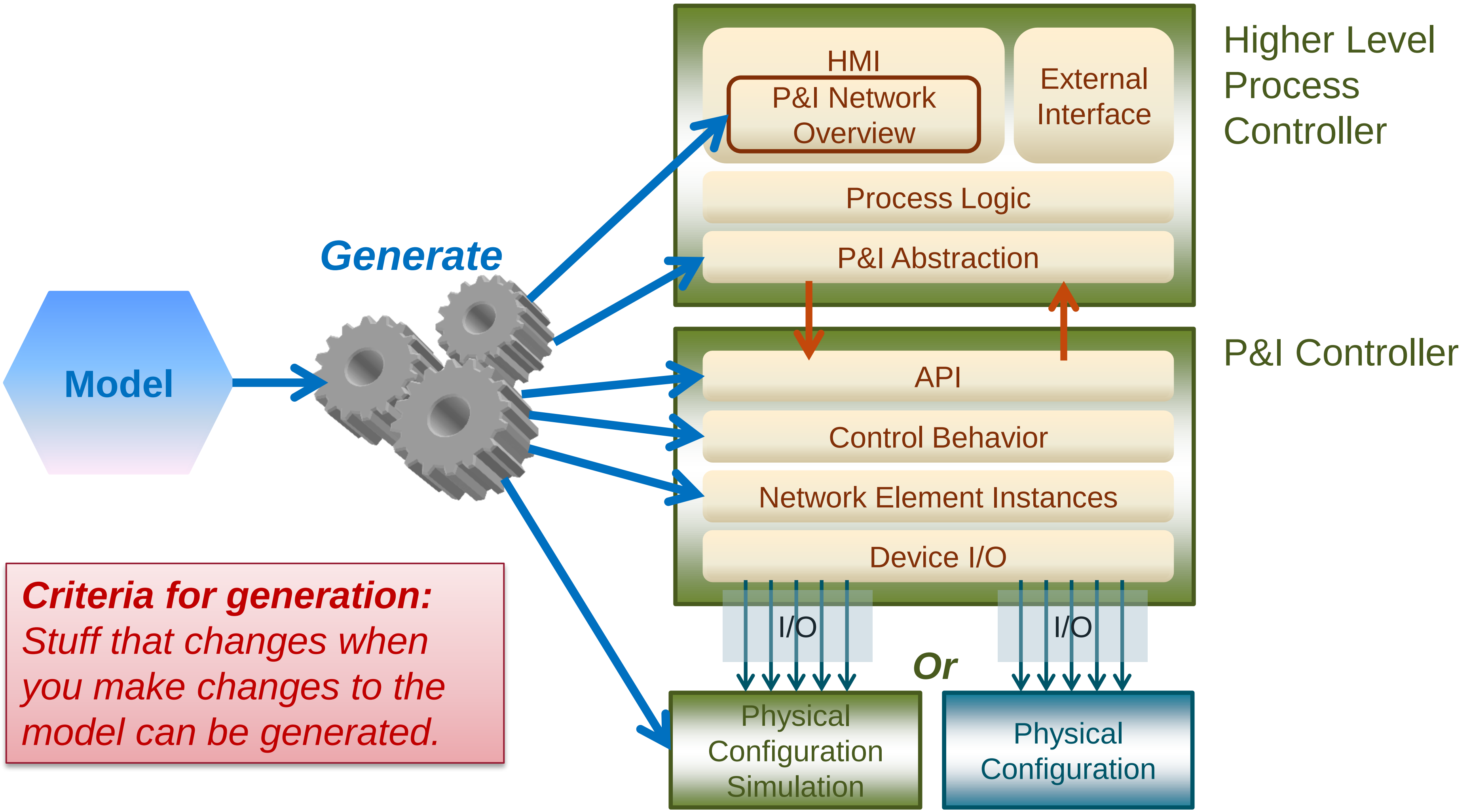
Piping & Instrumentation Domain: System Context (3)



Piping & Instrumentation Domain: Software Layers

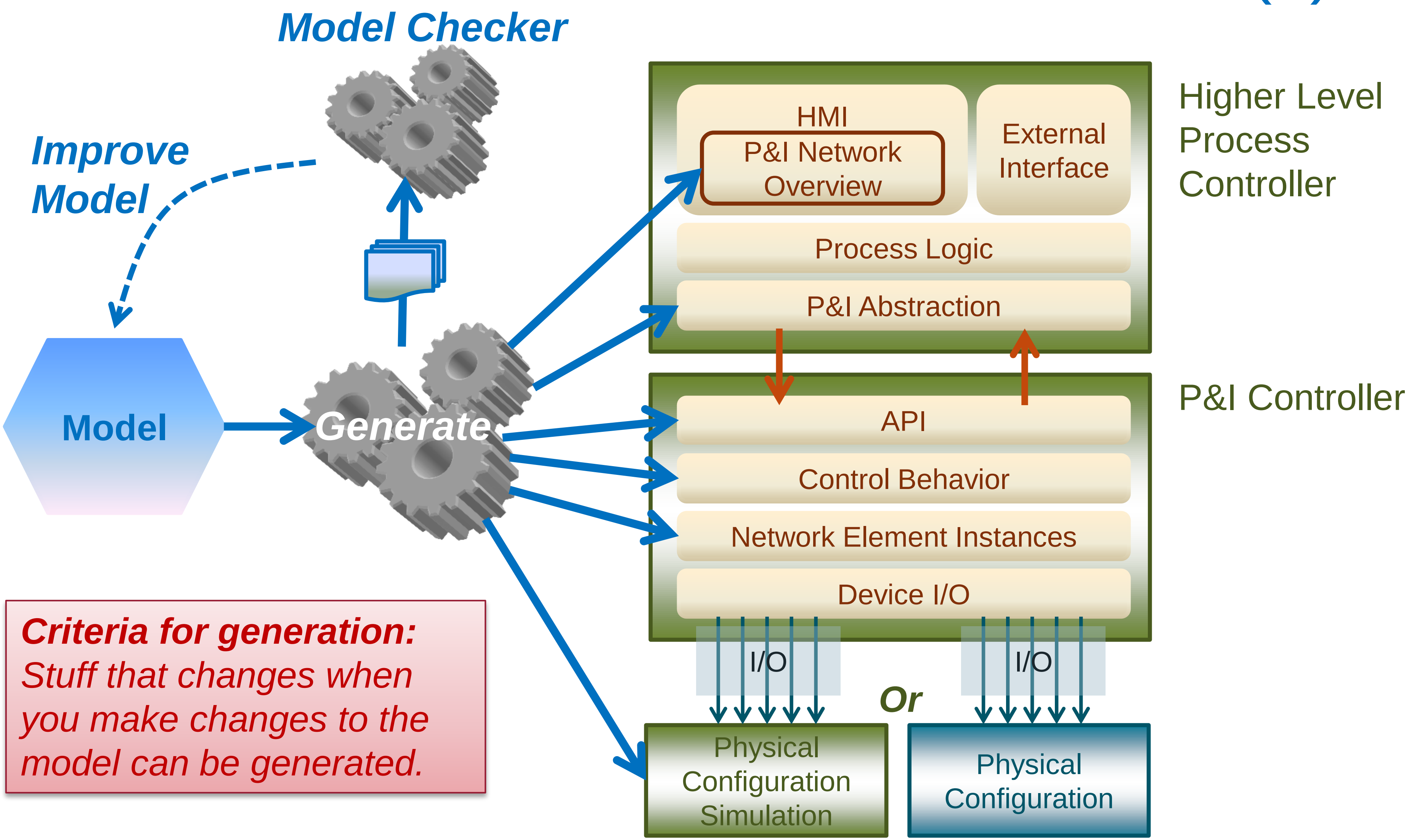


Appliance of MDSD
The vision



Criteria for generation:
Stuff that changes when you make changes to the model can be generated.

Appliance of MDSD
The vision (2)



Criteria for generation:
Stuff that changes when you make changes to the model can be generated.

P&I Structure Meta-model (1)
Domain concepts

- *Pipe*
  - Has 2 connection points
  - Each connection point may connect to all other elements, except other *pipe*
  - Transports gas/fluid from one connection point to the other connection point
    - Either due to gravity or to difference in pressure between the two connection points
  - Attributes:
    - Diameter (in mm), Length (in mm)
- *Joint*
  - Has multiple connection points
  - Only *pipes* can connect to *joints*
  - Allows splitting or merging of gas/fluid streams
  - Attributes:
    - None: Assume zero volume
- *System end*
  - Has 1 connection point
  - Only *pipes* can connect to *system ends*
  - Provides the connection with the 'outside' world. 2 kinds:
    - *Source*: Allows bringing gas/fluid into the system
    - *Exhaust*: Allows moving gas/fluid out of the system

P&I Structure Meta-model (2) Domain concepts

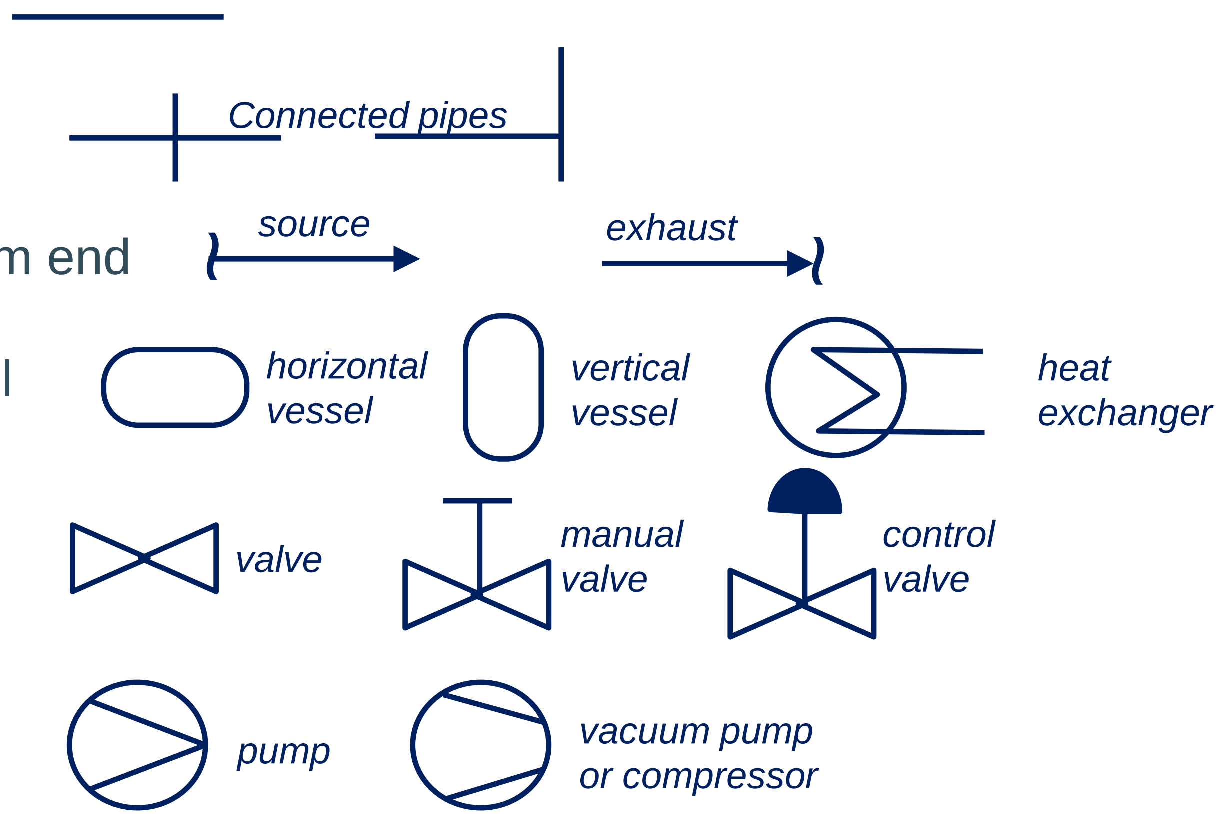
- *Vessel*
  - Has 1 or more connection points
  - Only *pipes* can connect to *vessels*
  - Varity of functions possible. E.g.:
    - Allows (temporary) storage of gas/fluid
    - Mixing of substances
    - Exchange of physical characteristics (e.g. heat exchanger)
  - E.g. Attributes storage vessel:
    - Capacity (in liter)
    - Variable number of connection points
- *Valve*
  - Has 2 or more ends
  - Only *pipes* can connect to *valves*
  - Blocks and enables the flow of gas/fluid streams
    - Discrete: Open/Close
    - Pin-valve: Allow an certain amount of flow per time period
  - Assume manual and electronic controllable valves only:
    - Open/close valve with either default state open or close
    - Analogue valve with settable amount of flow, with settable min, max and default flow

P&I Structure Meta-model (3) Domain concepts

- *Pump*
  - Has 2 or more connection points
  - Only *pipes* can connect to *pumps*
  - Builds up pressure or vacuum that forces the gas/fluids to flow through the system
  - Assume 3 kind of pumps:
    - Regular pump: transports substance from input to output connection point
    - Vacuum pump: with a suck and a blow connection point
    - Venturi pump: with a suck, input and output connection point
- *Measurement Instrument*
  - Has 1 or 2 connection points, depending on what is measured
  - Only *pipes* can connect to *measurement instruments*
  - Measure physical characteristics of the gas/fluid stream. (E.g. pressure, temperature, flow)
  - Assume instruments can be read out electronically by the controller:
    - Pressure sensor with one connection point
    - Temperature sensor with one connection point
    - Flow sensor with two connection points
- *Interlock*
  - Invariant constraint of the system – *must always be true*
  - Defines combination of states of elements in the system that never may happen or may always happen
  - If a constraint is violated, the system should immediately execute an emergency shutdown
    - Assume the interlock function is realised in the control software
  - The way interlocks are defined is not prescribed. Could think of decision table.

P&I Structure Meta-model (4a)
A well know domain notation (1)

- Examples of symbols domain experts are used to:
  - Pipe
  - Joint
  - System end
  - Vessel
  - Valve
  - Pump

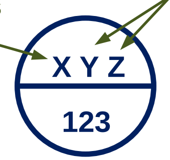


  - Measurement instrument – *see next sheet*

P&I Structure Meta-model (4b)
A well know domain notation (2)

- Measurement instrument

The first letter designates the measured variable



Succeeding letter(s) designate the function of the component, or modify the meaning of the first letter.

E.g.:
Pressure
Level
Flow
Temperature

E.g.:
Indicator
Recorder
Controller
Transmitter

Location

No Line
The instrument is mounted in the field near the process (close to the operator)

Solid Line
The instrument is mounted in the control room (accessible to the operator)

Dashed Line
The instrument is mounted out of sight (not accessible to the operator)

P&I Control Meta-model

Possible solution:

Behavior can be modeled as (UML) State Machines,
possibly augmented with Activity Diagrams for complex activities in a single state.

Control input/output can be modeled as (UML) Interfaces

The Challenge: In more details (1)

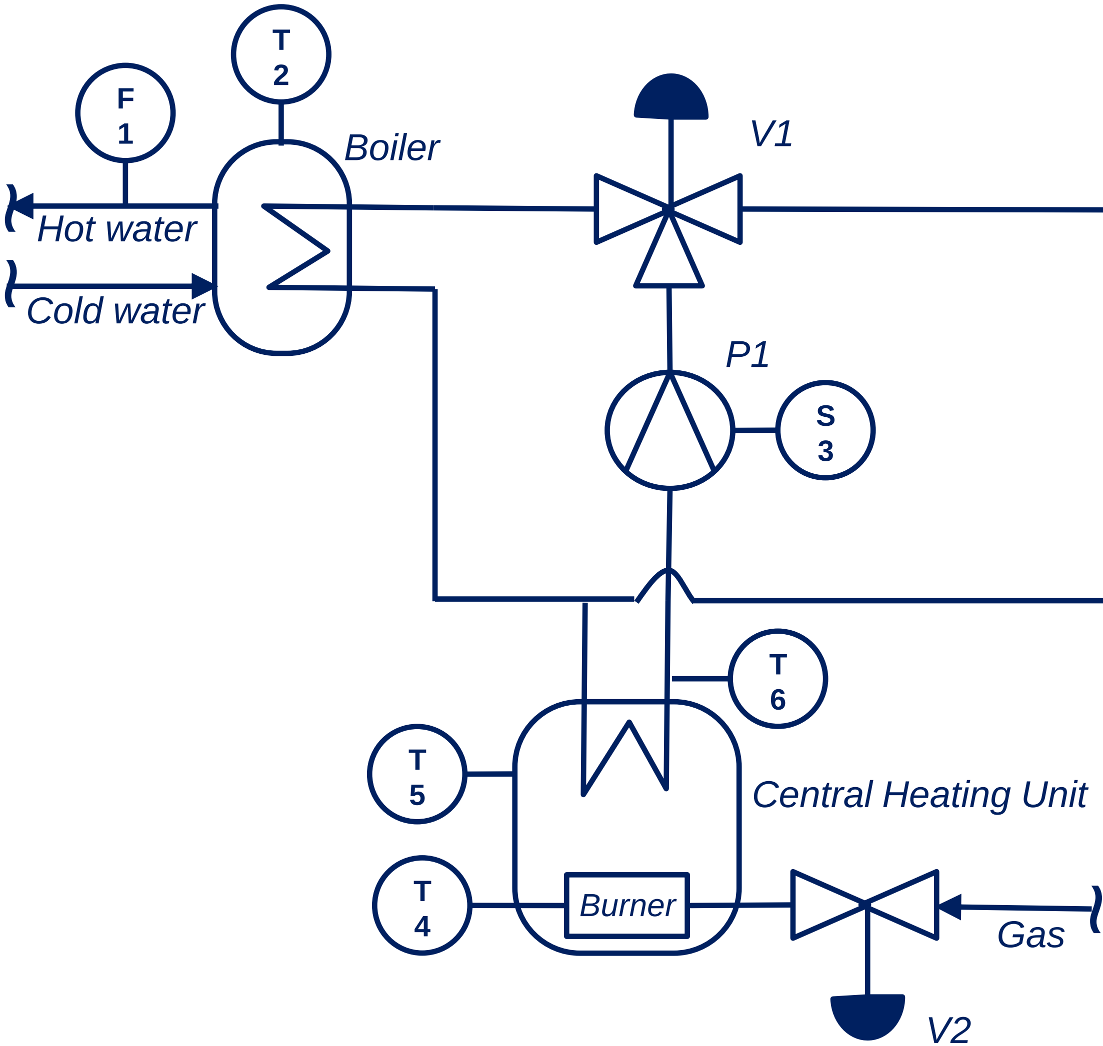
1) Be able to define the DSL elements: different elements, inheritance if elements are specializations, some characteristics of the elements.
2) Be able to define a P&I network of the central heating system
   - be able to instantiate and connect elements
   - be able to verify constraints like: max number of connected pipes to other non-pipe elements; non-pipe elements may only be connected to pipe-elements, etc.
   - be able to assign values to characteristics of elements
   - be able to define initial state of elements (e.g. open, close)
3) Be able to define the P&I network in such a way that it is intuitive to the domain expert (either diagram or text).
4) Be able to define the control behavior of the central heating system
   - as state machine or activity flow (or otherwise)
   - be able to define external operations provided by the structural network (e.g. administer gas x for certain amount, certain time, with certain flow …)
   - be able to define the states/flow required to fulfill these operations
   - be able to use references or names of actual element instances in the defined P&I structure network in your state/activity model. E.g. check for values or state, set values or state

The Challenge:
In more details (2)

5) Be able to add interlock/constraint definitions the define invariants of the central heating system. These invariants must always be valid. (The generated code should check on these).
6) Be able to generate structural definitions and stubs for a target.
7) Be able to generate control code for a target.
8) Be able to generate a mock representation (simulation) of the defined central heating system that behaves like it and that can be used to test the other generated stuff against.
9) Generate operations interface for higher level software that enables the use of the generated functionality through that API.
10) Very nice to have: Generate visualization software that can connect to the target to visualize the dynamic behavior.

The Challenge: In more details (3)

Example of P&I diagram for central heating system.



The Challenge: In more details (4)

- Provided input
  - This description
  - An example of the model
  - Behavior written down in requirements/constraints
    - Expect to receive additional requirements at the latest moment
  - Use TwinCAT as the target PLC (Structured Language)

- Domain references:
  - http://en.wikipedia.org/wiki/Piping_and_instrumentation_diagram
  - http://commons.wikimedia.org/wiki/Category:Chemical_engineering_symbols
  - http://materias.fi.uba.ar/7699/S_5.1.pdf
  - http://www.isa.org/Content/Microsites165/SP18,_Instrument_Signals_and_Alarms/Home163/ISA_Standards_for_Committee_Use/S_55.pdf

Twincat

- Soft PLC – Windows based
- Documentation:
  - http://www.beckhoff.de/english.asp?twincat/default.htm
    - InfoSys.exe in preferred language (~435 MB)
  - Twincat quickstart in the help
    - Example: Machine.pro
- Software:
  - http://www.beckhoff.de/english.asp?twincat/default.htm
  - 30 day trial version – can be reinstalled several times
    - TwinCAT 2.11 R2, build 2038 (~48 MB) (32 bit only)
    - TwinCAT 2.11 R2 X64 (~28 MB) (64 bit only)