School of Electrical Engineering

Department of Electrical Engineering and Automation

ELEC 8102 Design of Distributed Intelligent Automation Systems

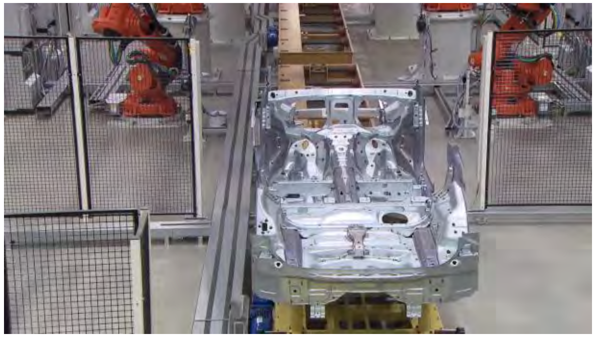
Lecture 2.
Software in Industrial Automation

Valeriy Vyatkin

Aalto University

2022

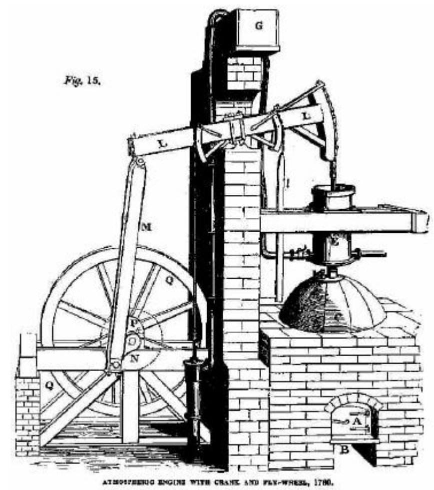
What is Industrial Automation?



Generations of Automation Systems

1. Relay based controllers (40s – 50s)
2. Microprocessor based PLCs (70s)
3. Multifunctional PLCs
4. Industrial Networks
5. Internet of Things

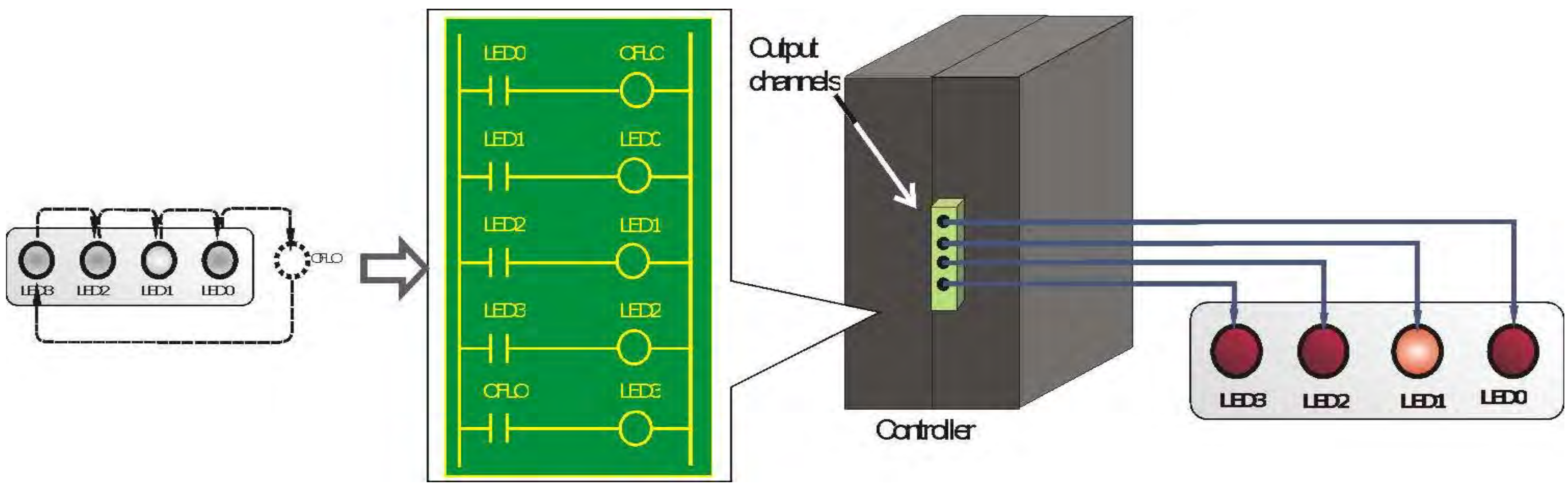
Generation 0. Mechanical regulators



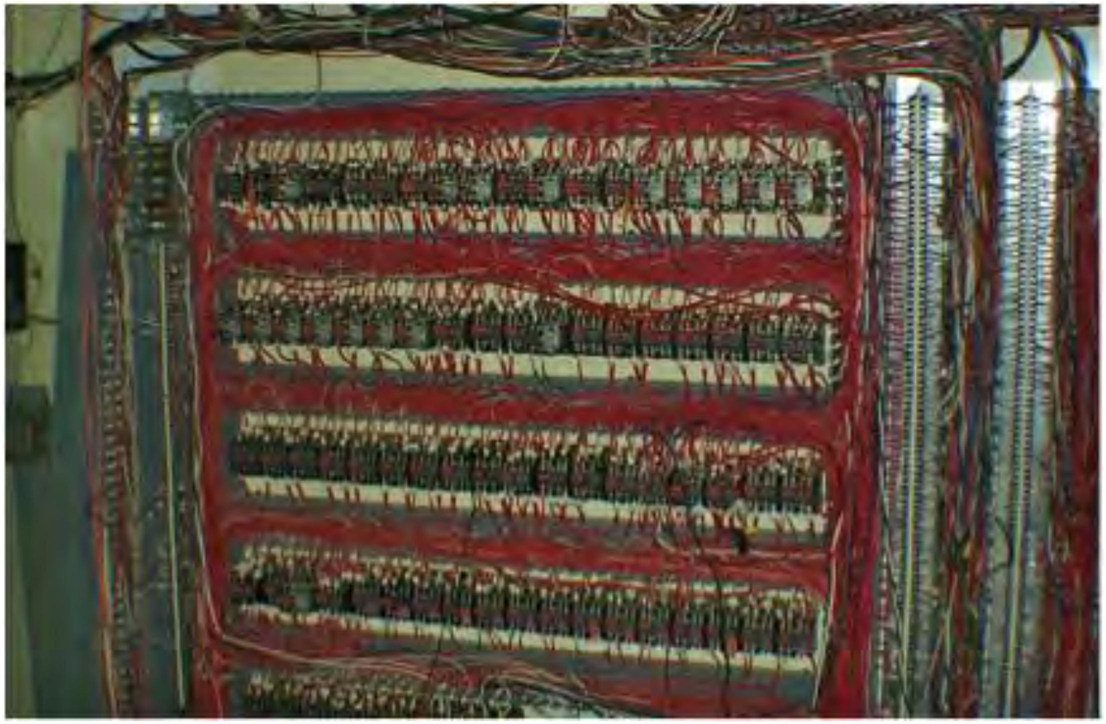
Mine in Cornwall equipped with first steam engine in 1790. First <u>industrial</u> regulator invented by James Watt to keep the wheel's speed constant.

Generation 1: Relay Ladder Circuits

Hard-wired ladder logic circuits were widely used to control industrial equipment. This explains current popularity of the Ladder Diagram language for programming industrial controllers

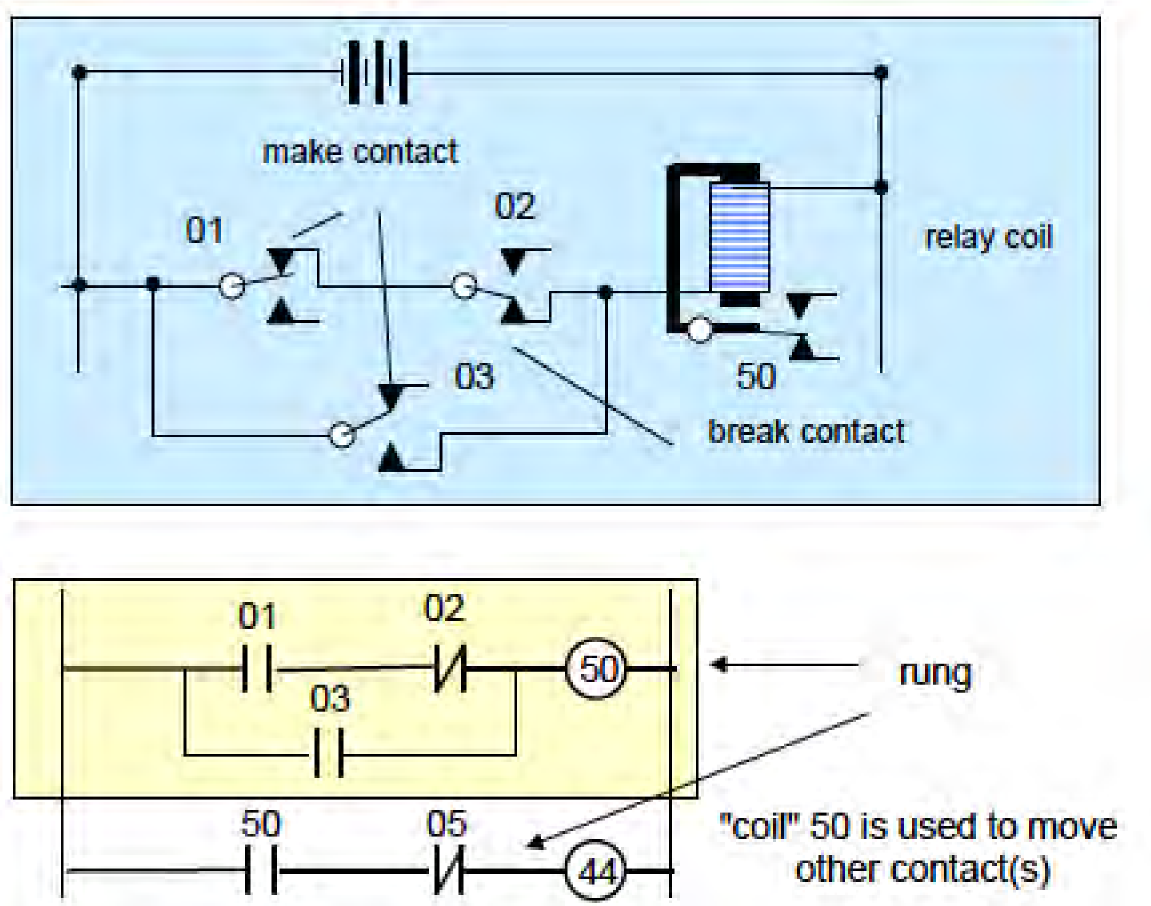


Relay Ladder Logic



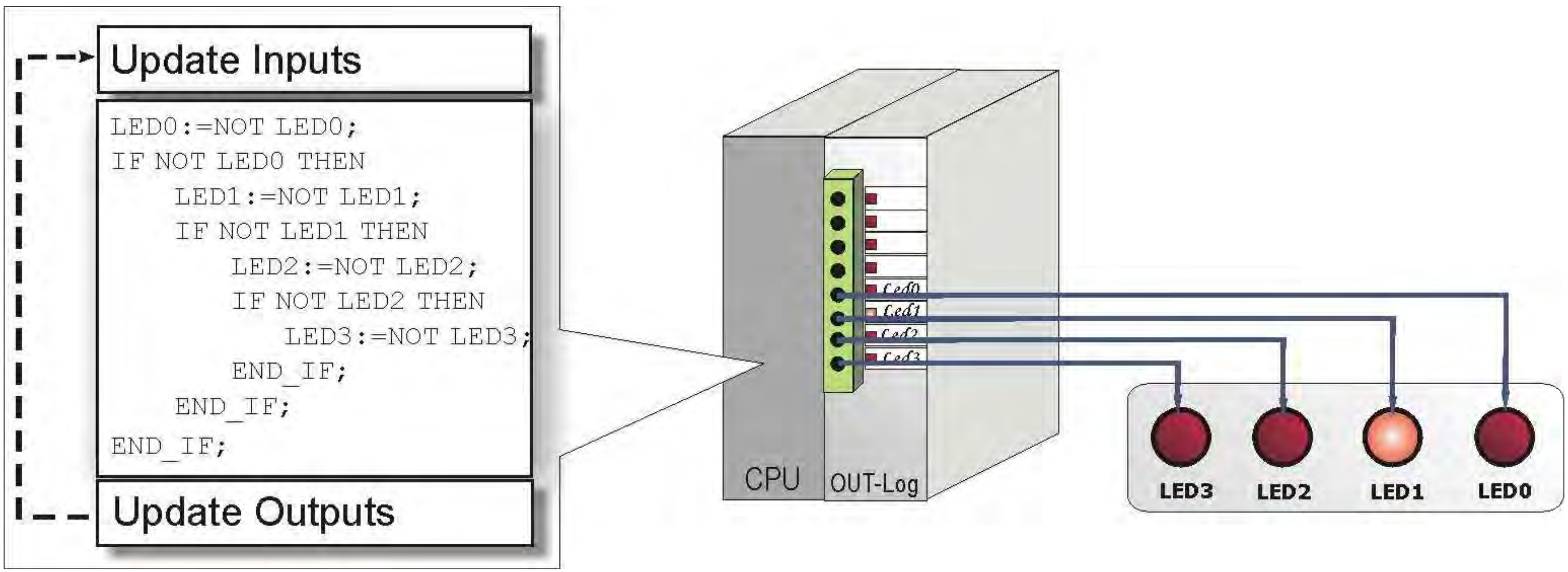
Control logic implemented as hard-wired **relays**

Electric Circuit and Ladder Diagram

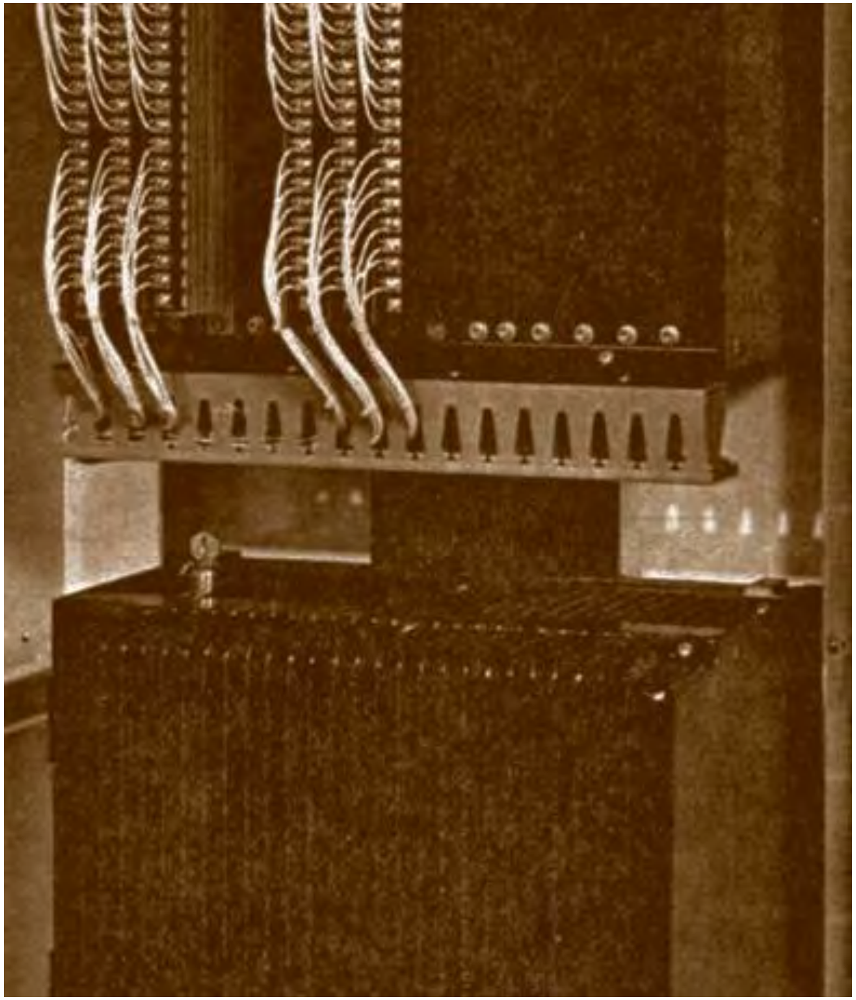


Generation 2: Programmable Logic Controllers (PLCs)

PLCs – specially hardened industrial computers - tremendously improved flexibility of automation systems



Programmable Logic Controllers

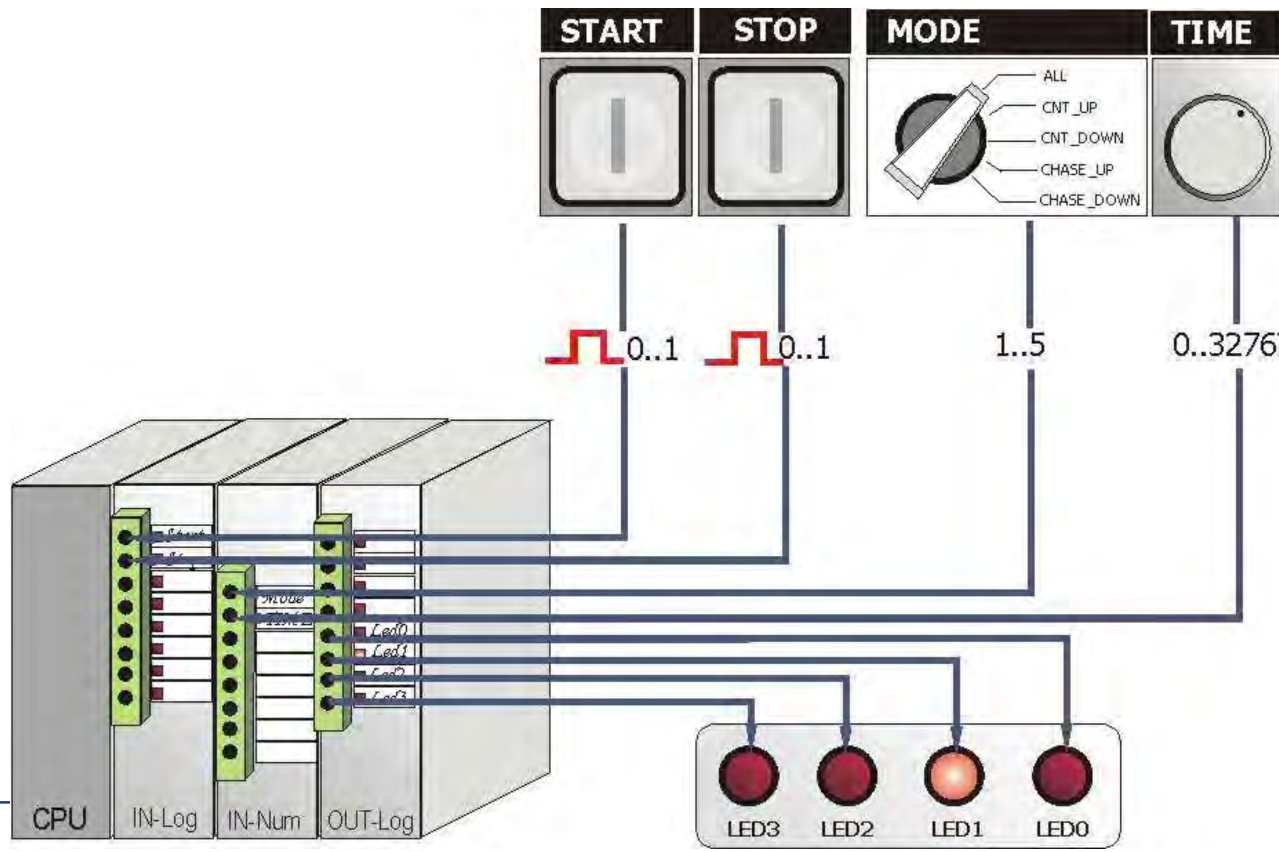


The Modicon 084 PLC

Source: C. C. Bissel, A history of Automatic Control

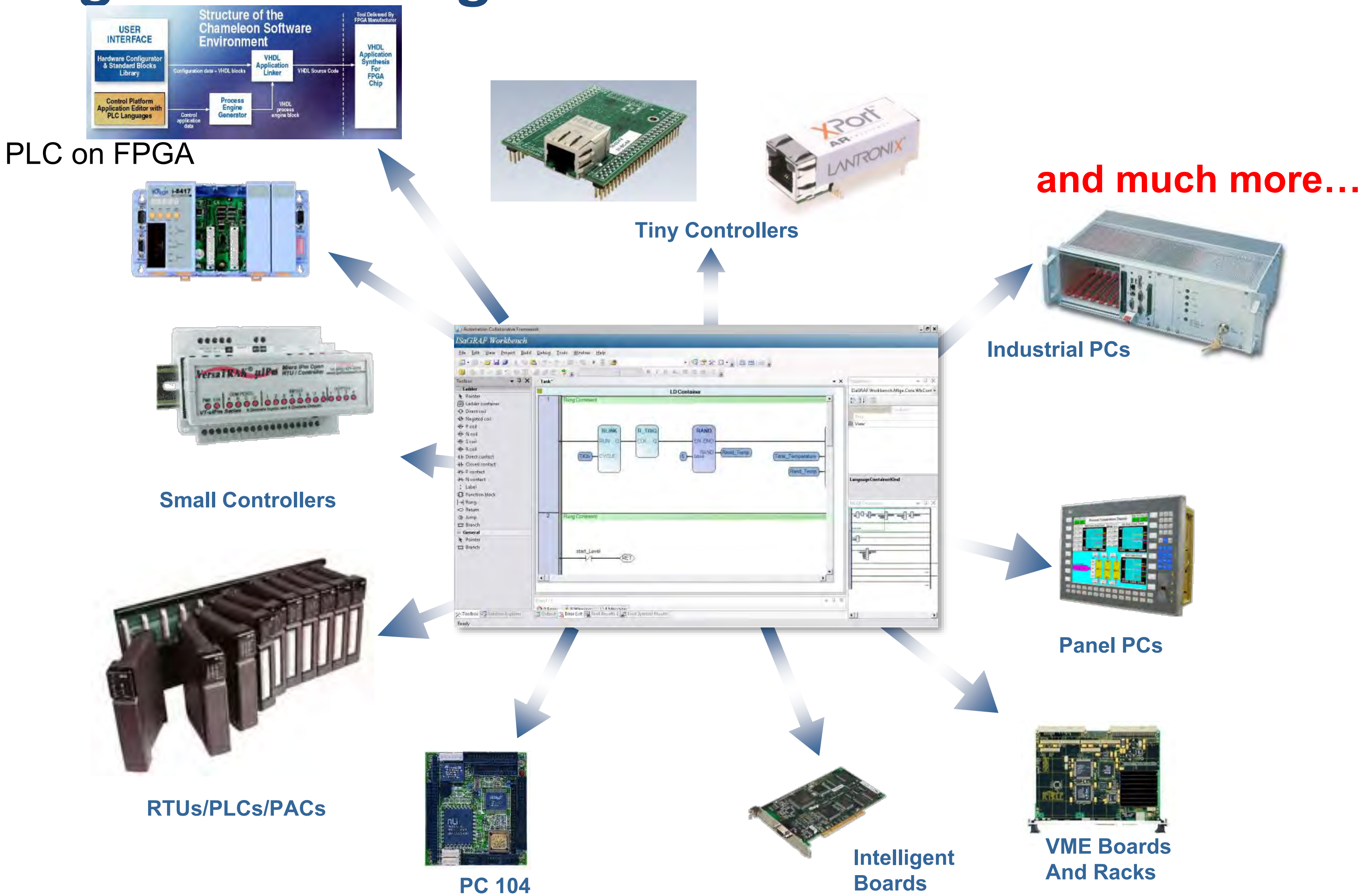
Generation 3: Multifunctional PLCs

Modern PLCs know many programming languages and have versatile and easily expandable architecture.

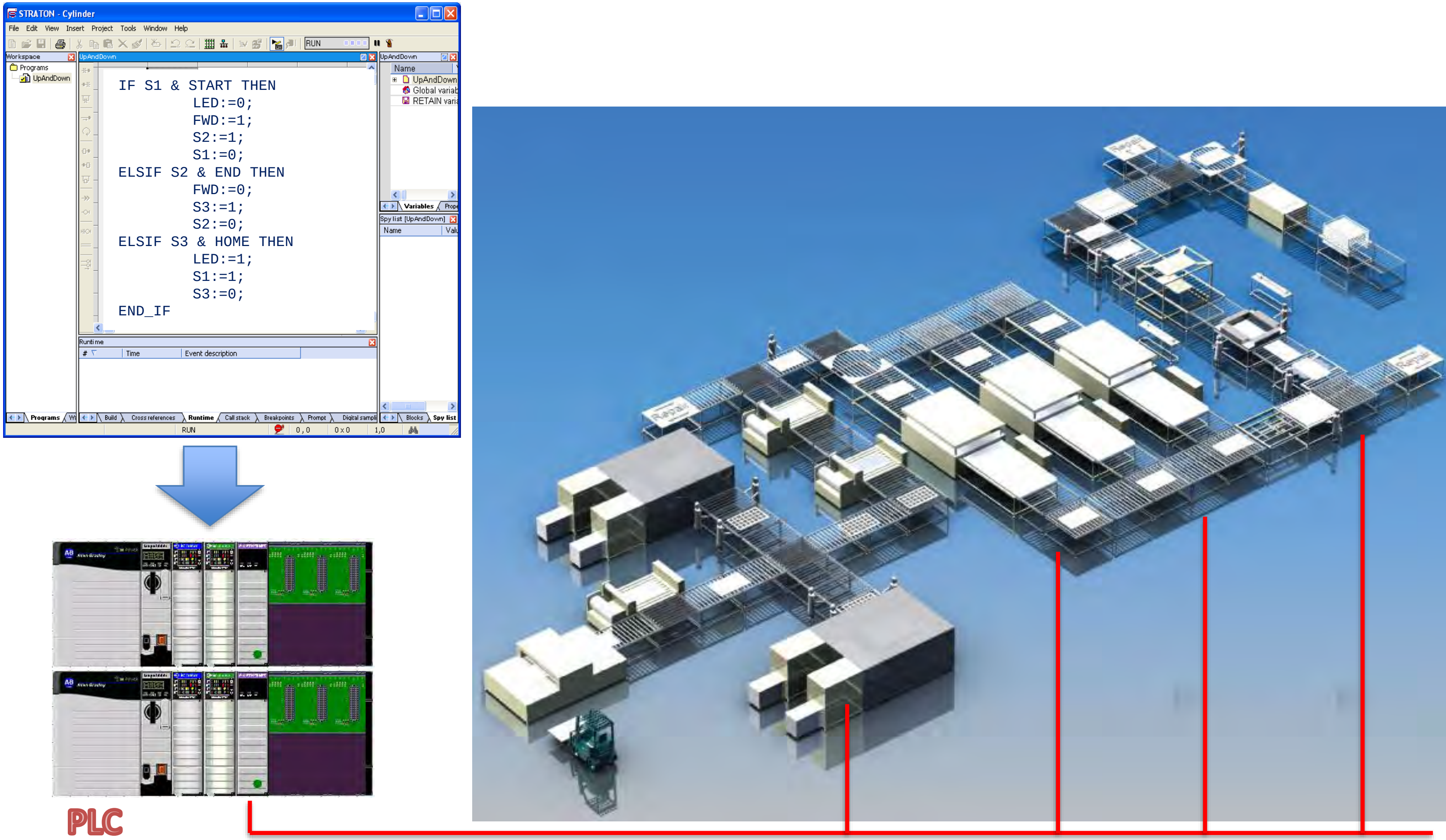


Generation 3: Multifunctional PLCs

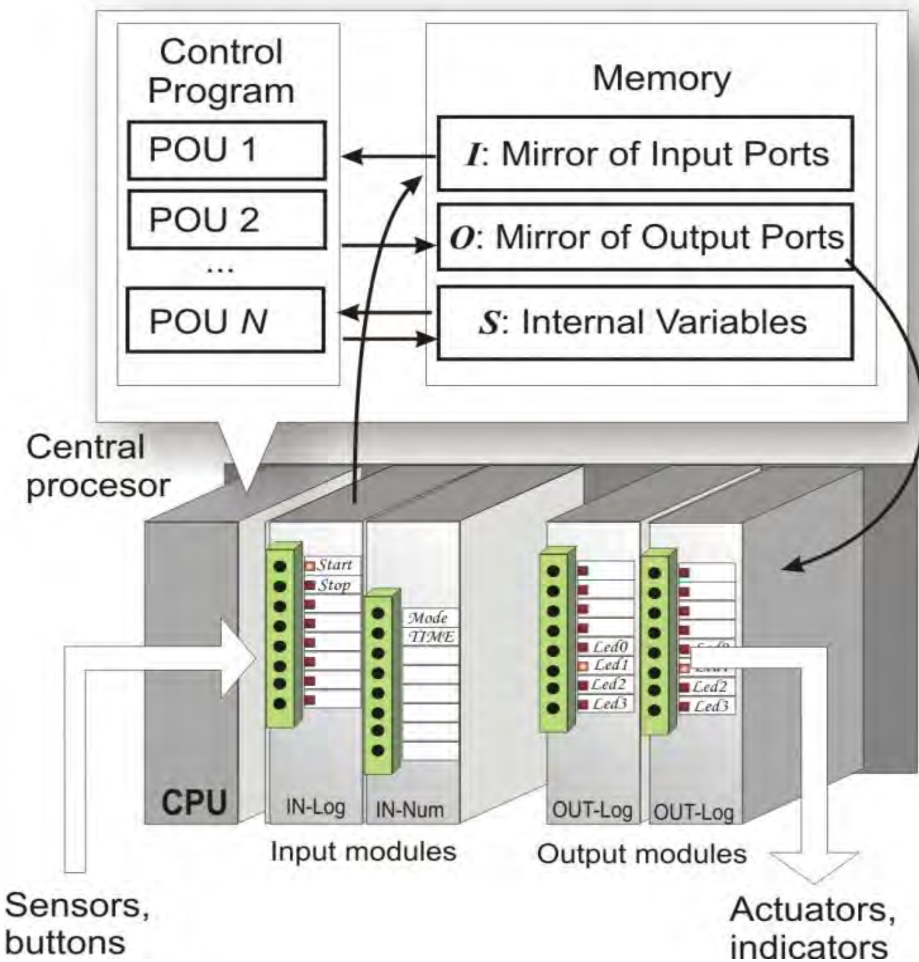
Programmable Logic Controllers: Form Factors



Generation 4: Industrial Networks



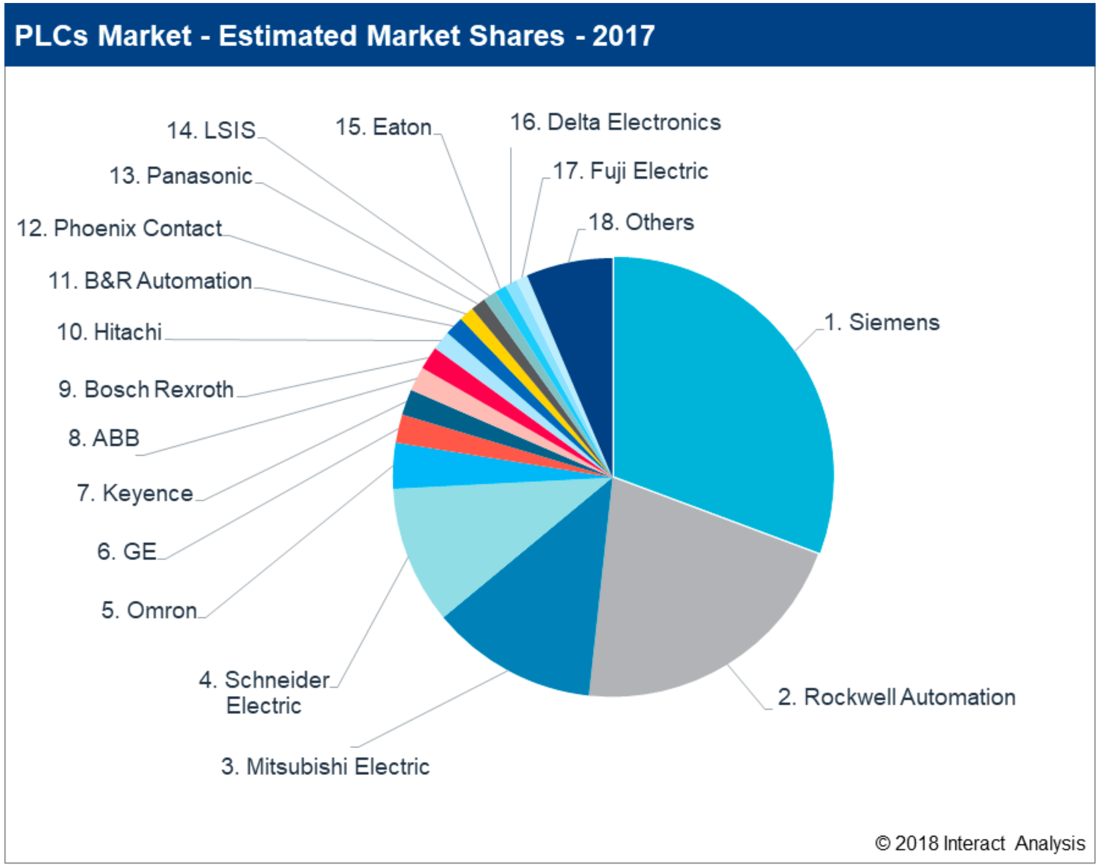
PLC as Integration Platform



PLC Market

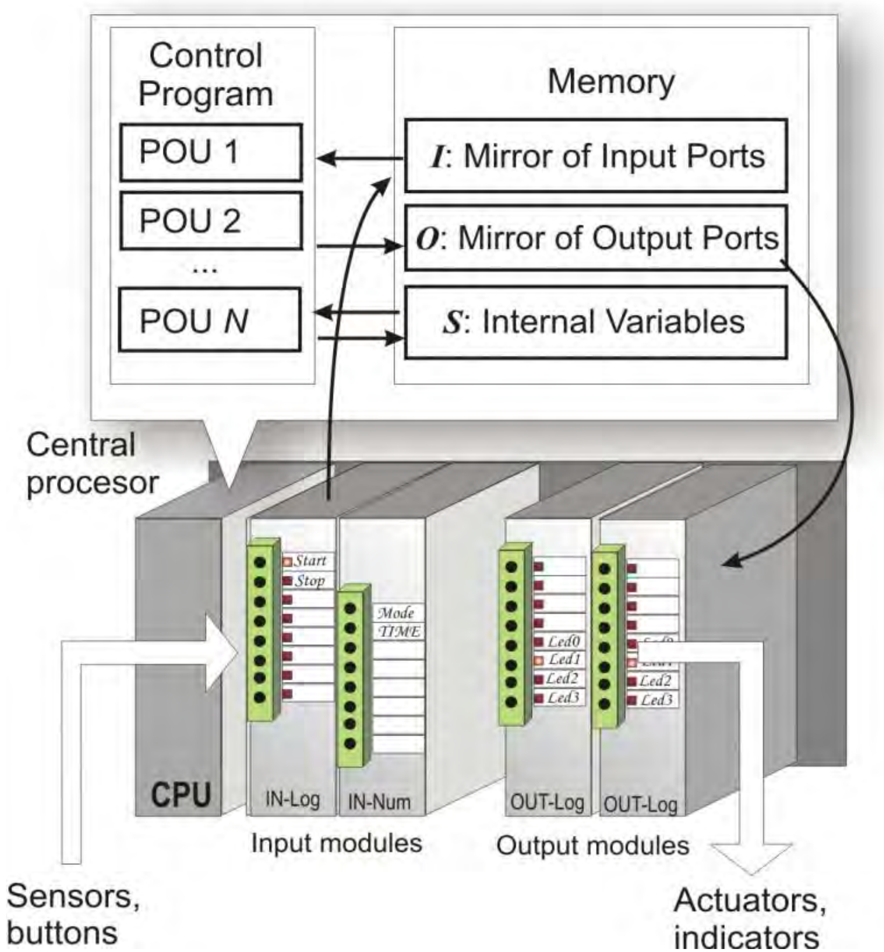
The programmable logic controller (**PLC**) **market** was valued at USD 3897.36 million in 2019 and is expected to reach USD 4292.66 million by 2025.

https://www.mordorintelligence.com/industry-reports/programmable-logic-controller-plc-market



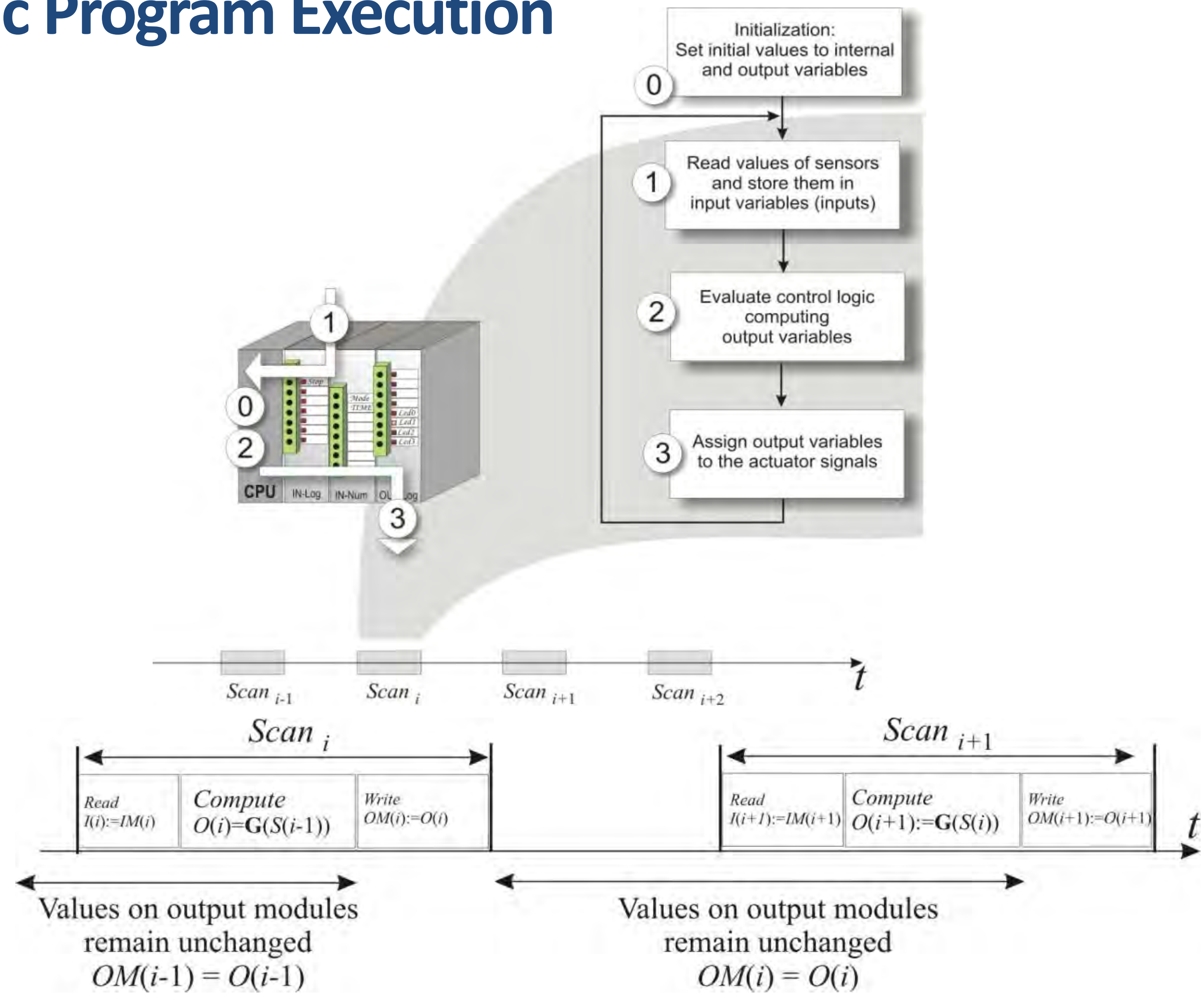
Computations in Controllers

PLC



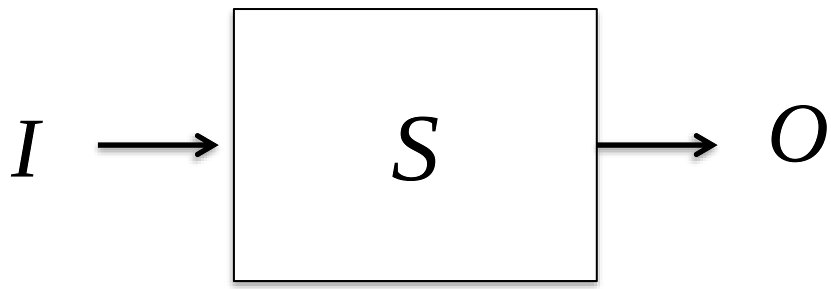
Control Program
POU 1
POU 2
...
POU N
Memory
I: Mirror of Input Ports
O: Mirror of Output Ports
S: Internal Variables
Central procesor
CPU
Start
Stop
Mode
TIME
Led0
Led1
Led2
Led3
Led2
Led3
IN-Log
IN-Num
OUT-Log
OUT-Log
Input modules
Output modules
Sensors, buttons
Actuators, indicators

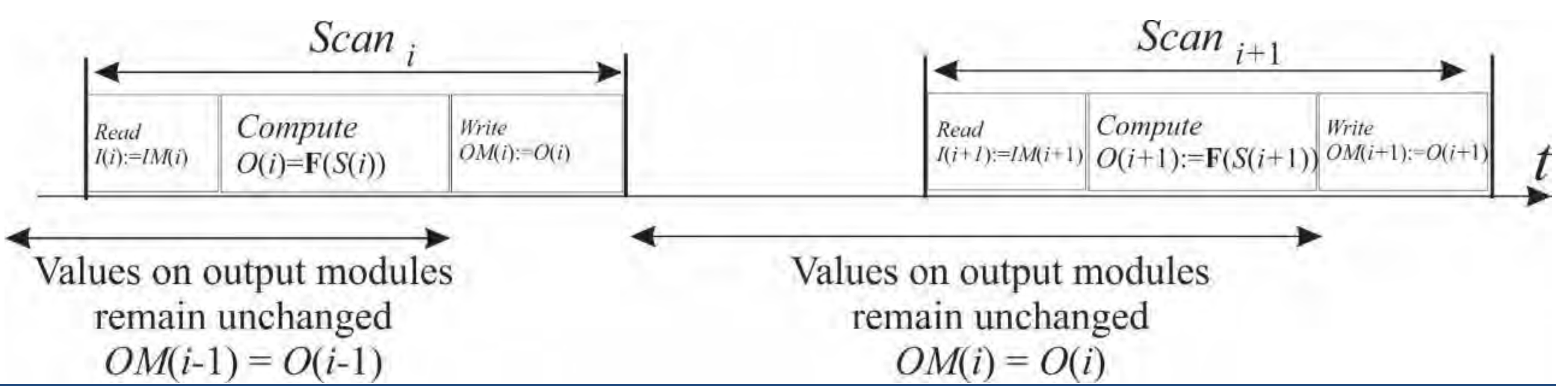
Cyclic Program Execution



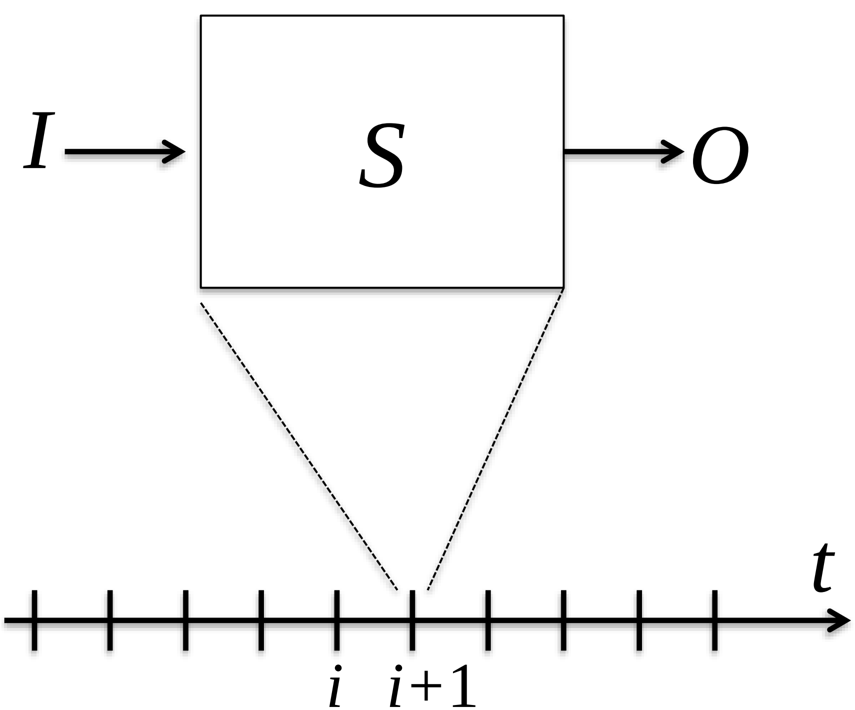
Formal Models of Automation Logic

Let us denote by I and O Boolean vectors of inputs and outputs, and S is Boolean vector of state variables.
Then the semantics of the controller can be described by the following system of Boolean assignments:





General Models of PLC execution



Combinatorial:

$O(i)=\mathrm{F}(I(i))$

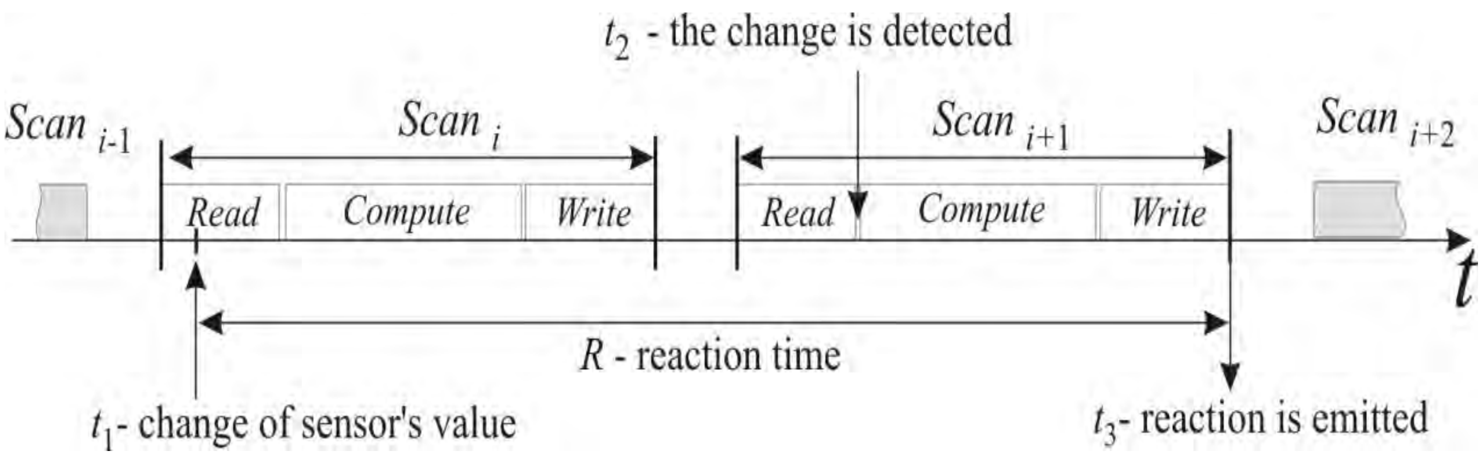
Moore type state machine:

$$\begin{cases} S(i+1) := \mathrm{T}(I(i+1), S(i)); \\ O(i+1) := \mathrm{F}(S(i+1)), \end{cases}$$

Mealy type state machine:

$$\begin{cases} S(i+1) := \mathrm{T}(I(i+1), S(i)); \\ O(i+1) := \mathrm{F}(I(i+1), S(i)); \end{cases}$$

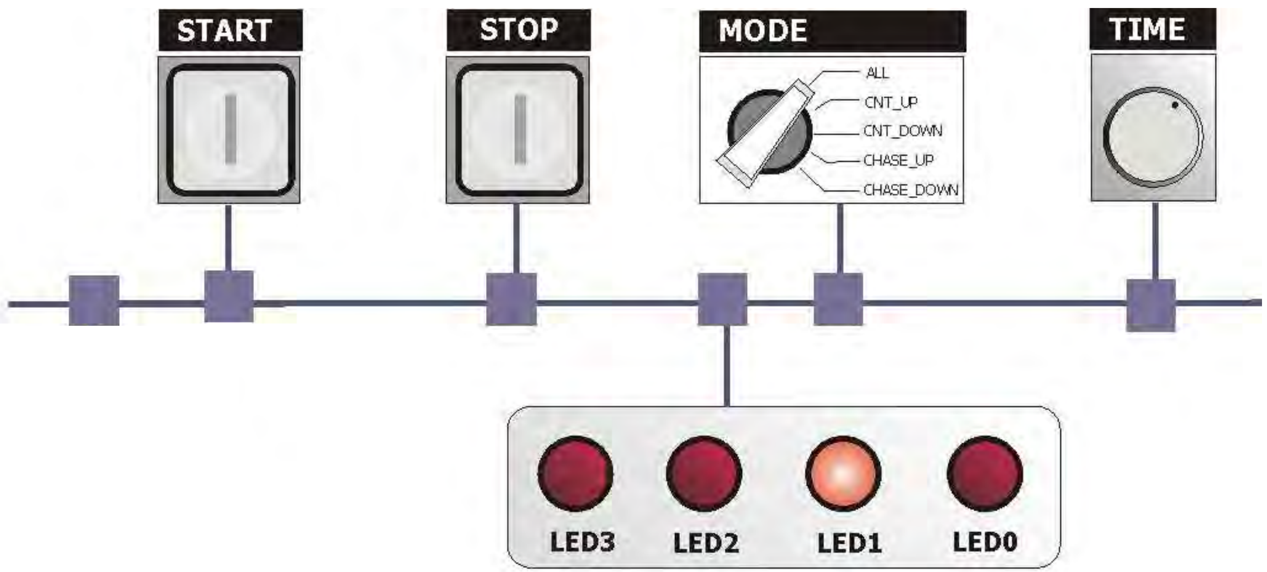
Reaction of PLC



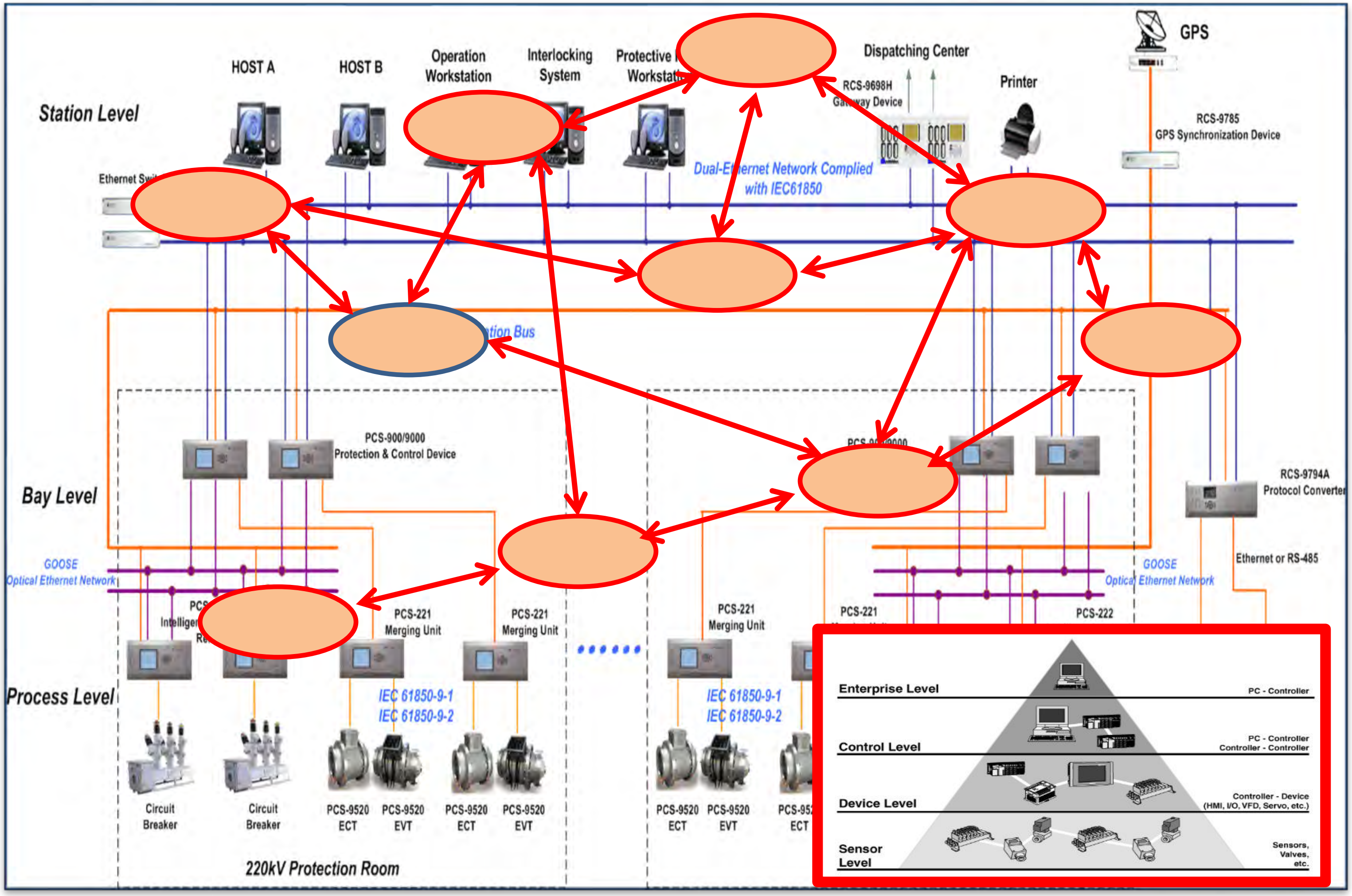
Why PLCs are around?

- Reactivity
- Timeliness
- Reliability: if input is read with an error, the error will be corrected in the next scan.

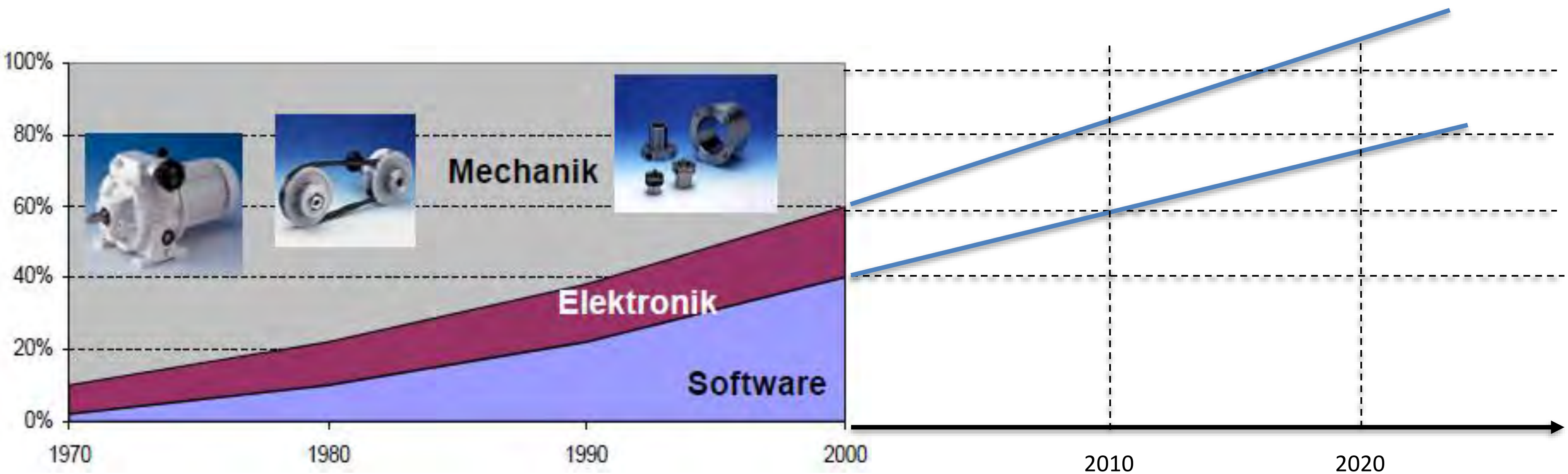
Generation 5: From Networking to Internet of Things



Machine To Machine Communication, Internet of Things



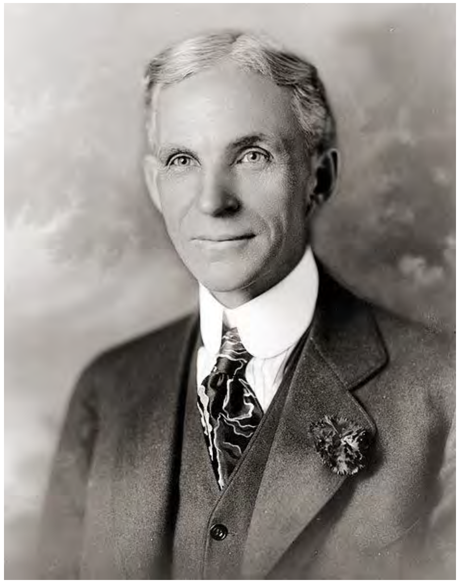
Software Ratio in Industrial Automation



According to Verband Deutscher Maschinen- und Anlagenbau e.V. (VDMA)

Production in the past…

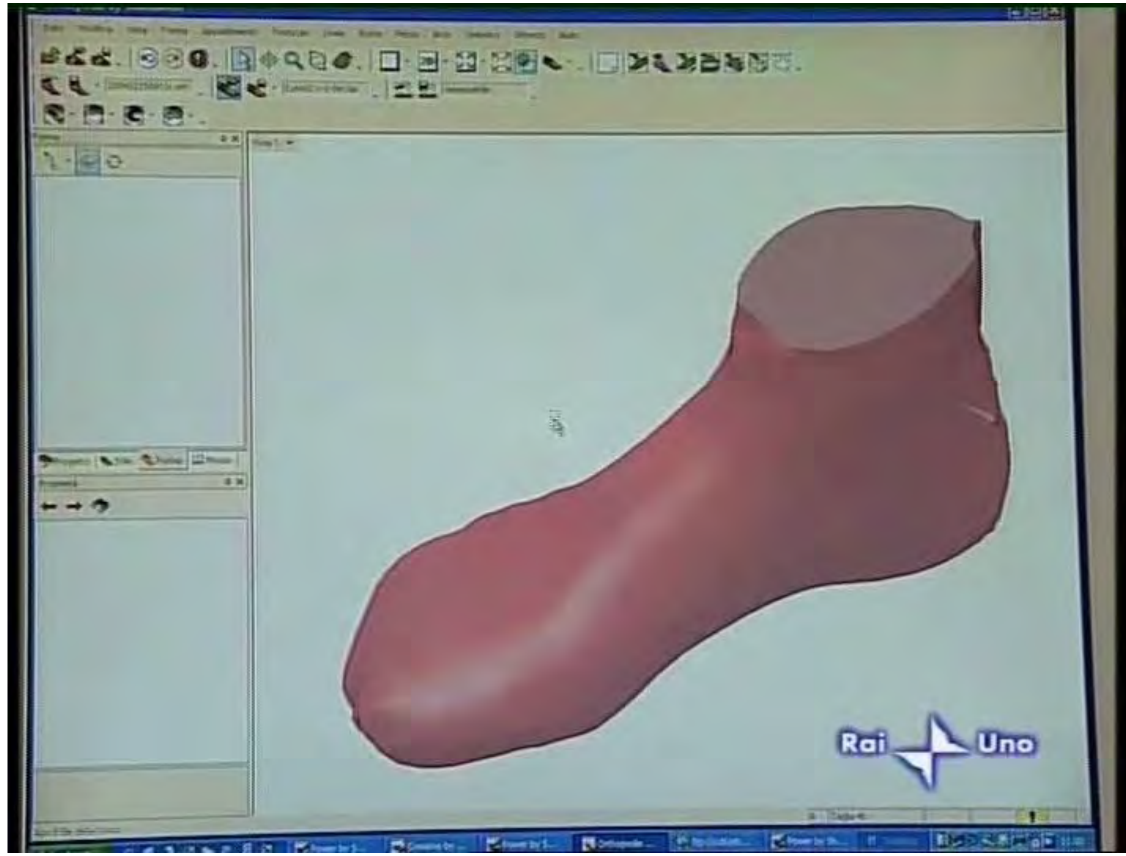
"Any customer can have a car painted any colour that he wants so long as it is black!"



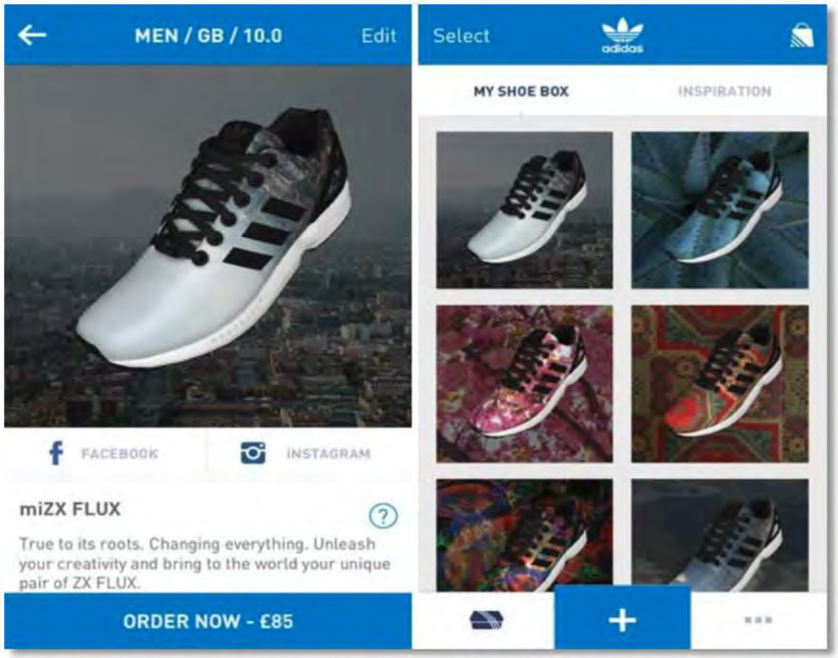
Henry Ford (1863-1947)
Source of photo: Wikipedia

1926 Ford Model T
Source of photo: Boldride

XXI Century: Manufacturing to order!

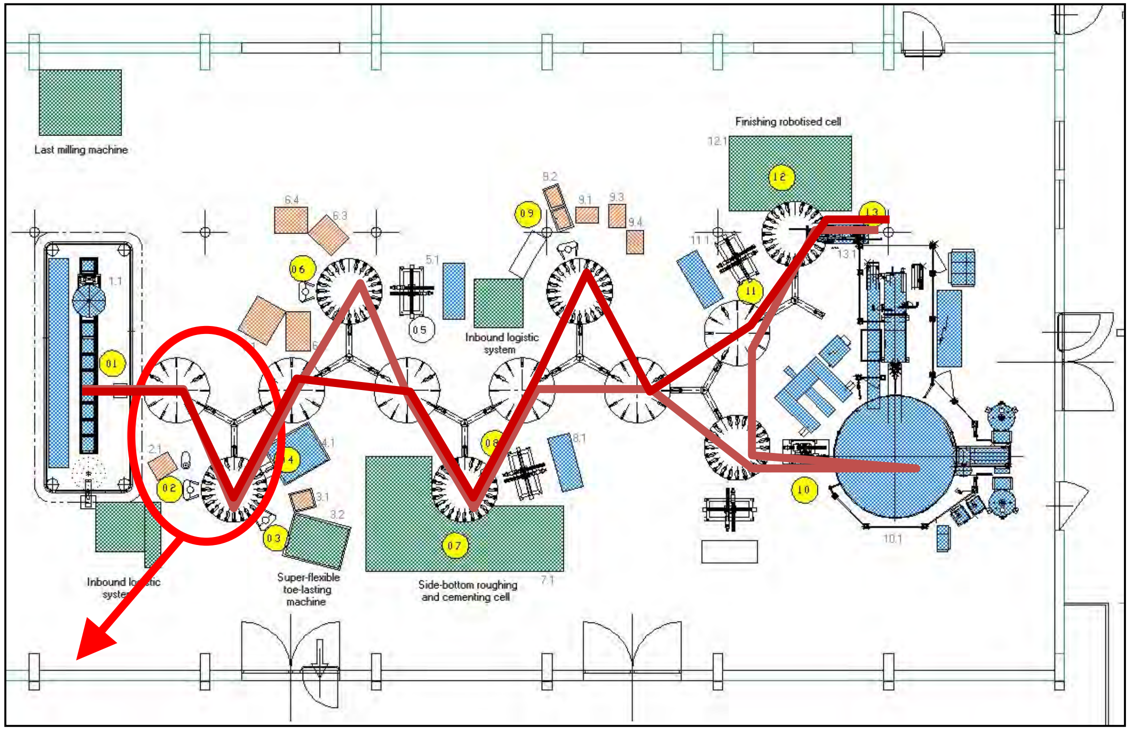


The Adidas Speedfactory: Bringing Sports Shoes Production back to Germany by Industrie 4.0 for Mass Customization

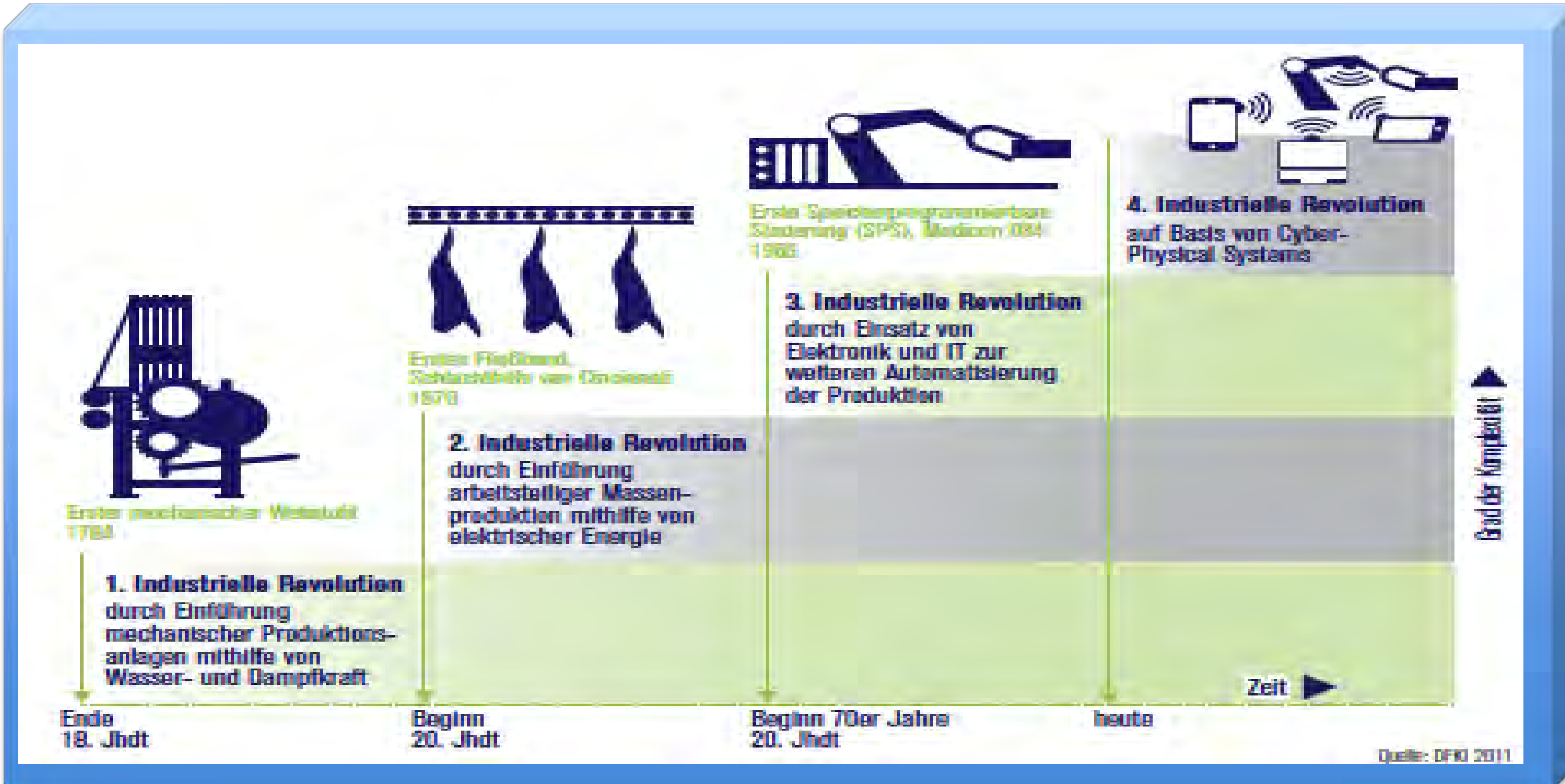


- The costumers can design their own short shoes using an App.
- Since the customer wants to receive his personalized product on the next day or faster, long logistic chains from low-wage countries are no longer acceptable in the era of mass customization.
- Thus, adidas decided to open various "speedfactories" for personlized shoes in Germany close to the customer, using Cyber-physical production systems (CPPS).

Flexible Manufacturing



German Industrie 4.0



Internet of Things

Cyber-Physical Systems

Industry 4.0: New Factory Floor

Industrial robots

3D printers

Mobile machines

Cyber-physical system

Cyber-physical system is a concept, assuming tight integration of computational resources to physical processes

Example: a flock of drones

Industrial CPS

Cyber-Physical Systems (CPS) is an emergent approach that focuses on the integration of computational applications with physical devices, being designed as a network of interacting cyber and physical elements.

CPS control and monitor real-world physical infrastructures and thus is starting having a high impact in industrial automation.

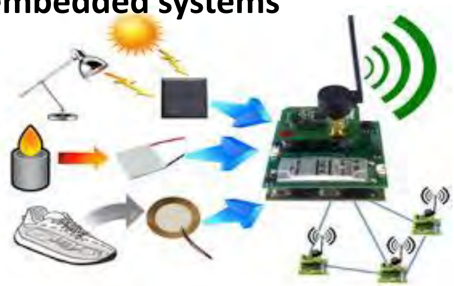
Leitão, Paulo, Armando Walter Colombo, and Stamatis Karnouskos. "Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges." *Computers in Industry* 81 (2016): 11-25.

Is this modular production line a cyber-physical system? What made it such?

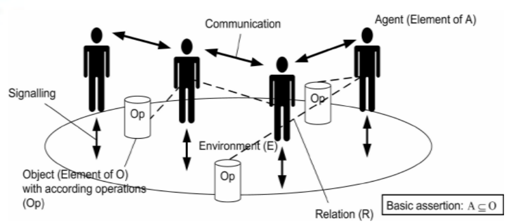
Cyber-Physical Systems

Supporting technologies

Communicating embedded systems

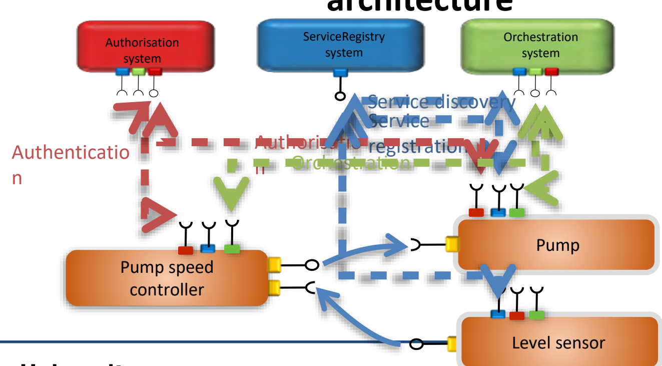


Agents



Cloud

Service-oriented architecture

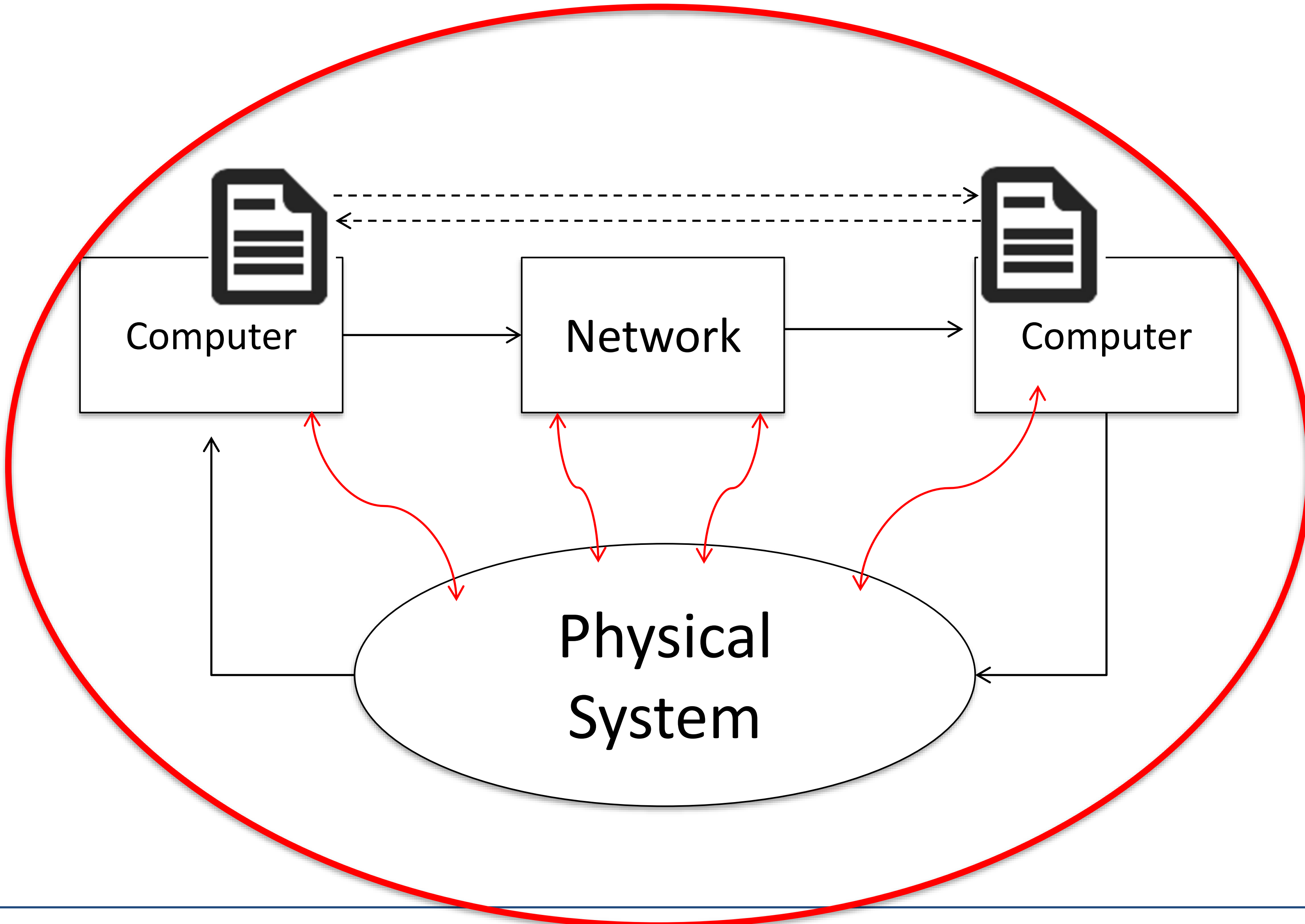


CPS science

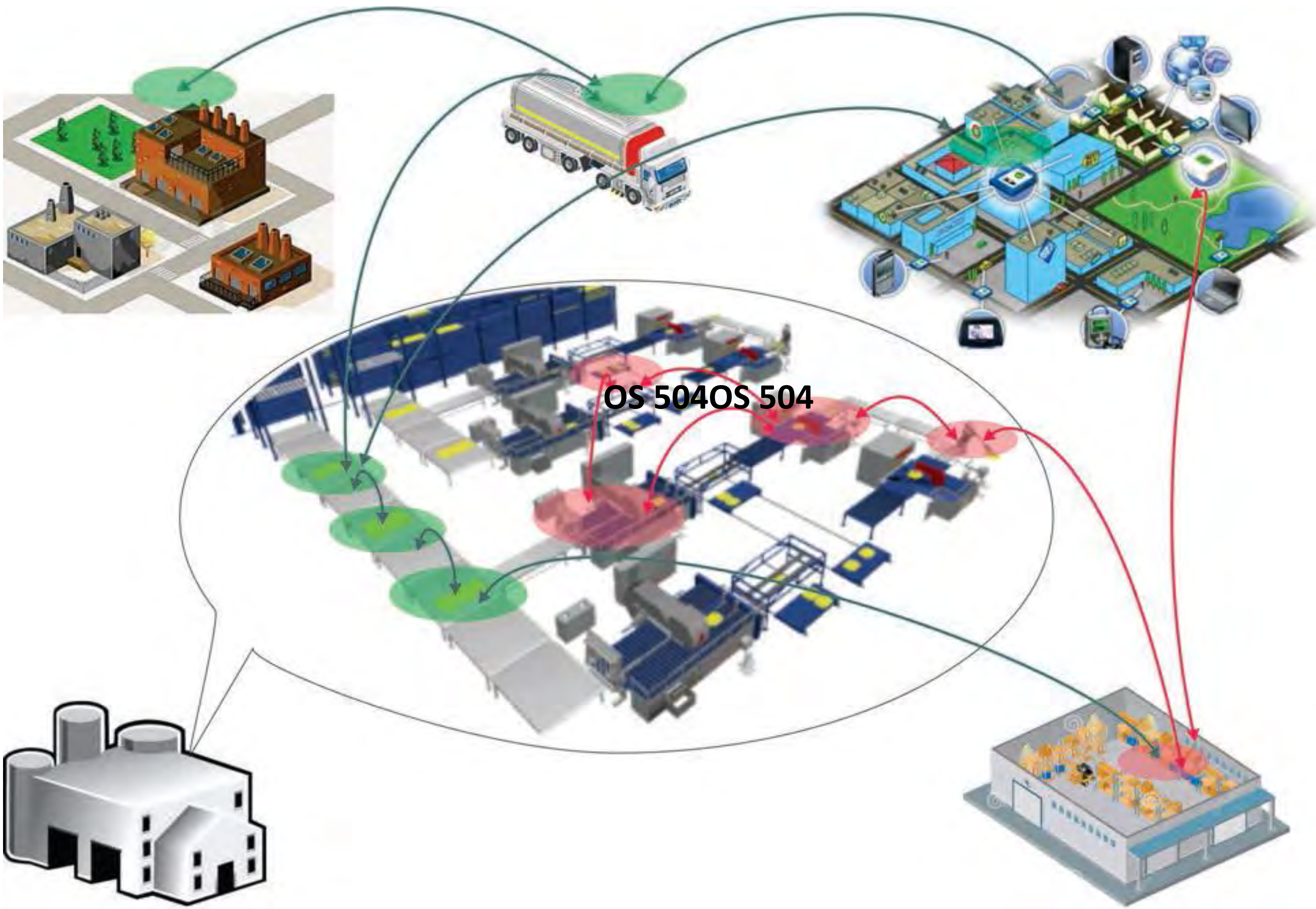
How to make them working in a predictable and robust manner ?

Cyber-Physical Engineering !

Cyber-Physical phenomena: cross influences



Internet of Machines:
Internet Lessons and business models



Aalto Factory of the Future at Scanautomatic Fair 2018

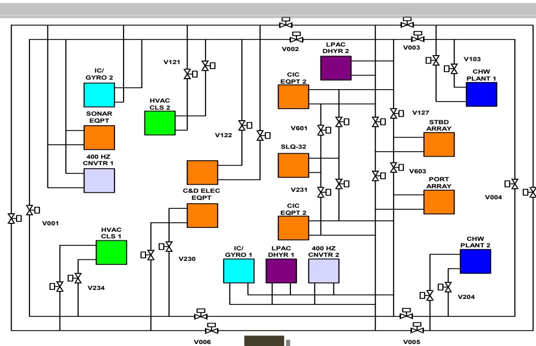
Dream: Bio-inspired Self Assembly


2

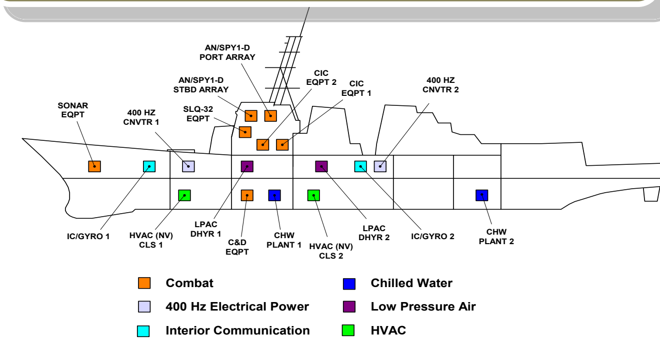
Autonomous Control Systems Application:

Shipboard Automation

Agent Based Solution Developed



Navy Requires Highly Survivable System



Agent based prototype developed

- Test and debug automatically generated agents
- Test Agent Behavior for different configurations

Demonstrate on the Navy's Land based Simulator

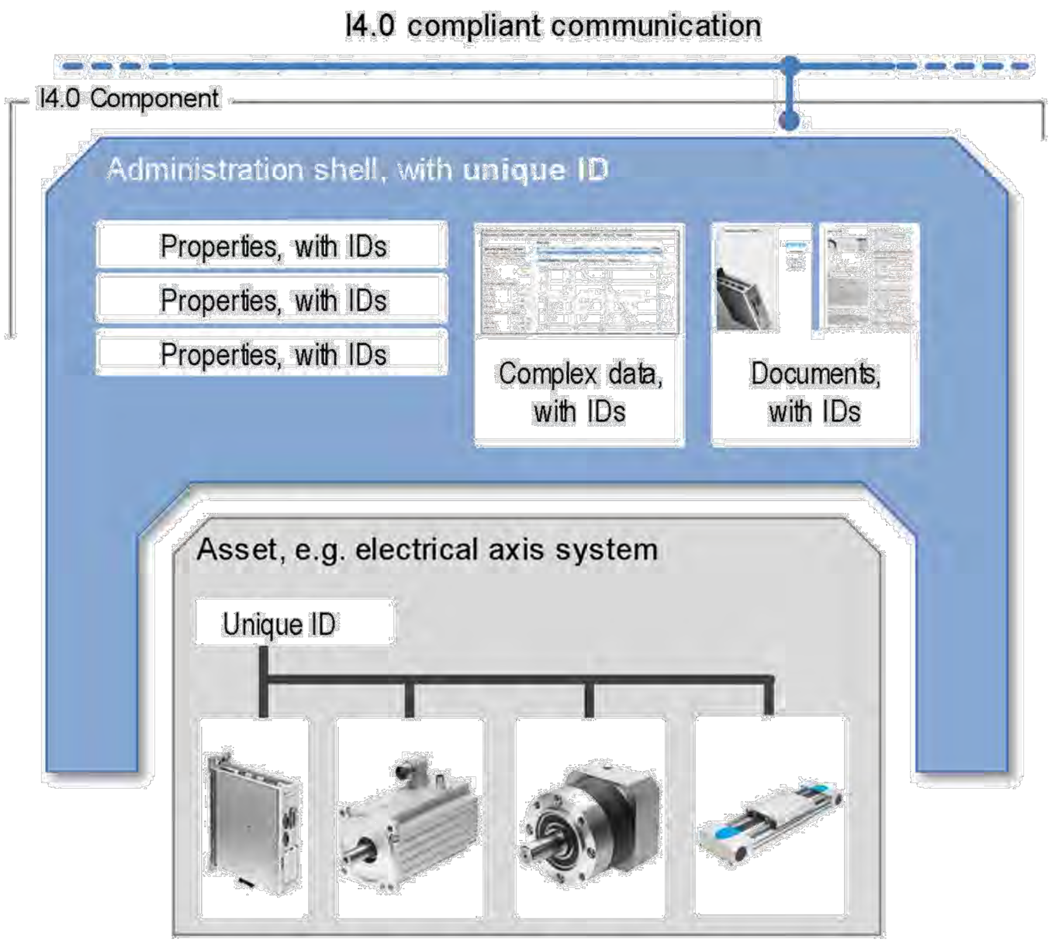
- Test Reconfigurable Shipboard Automation Architecture

Potential future applications: Automated, self-adjusting manufacturing lines to optimize throughput

Platform Industrie 4.0: the **Administration Shell**

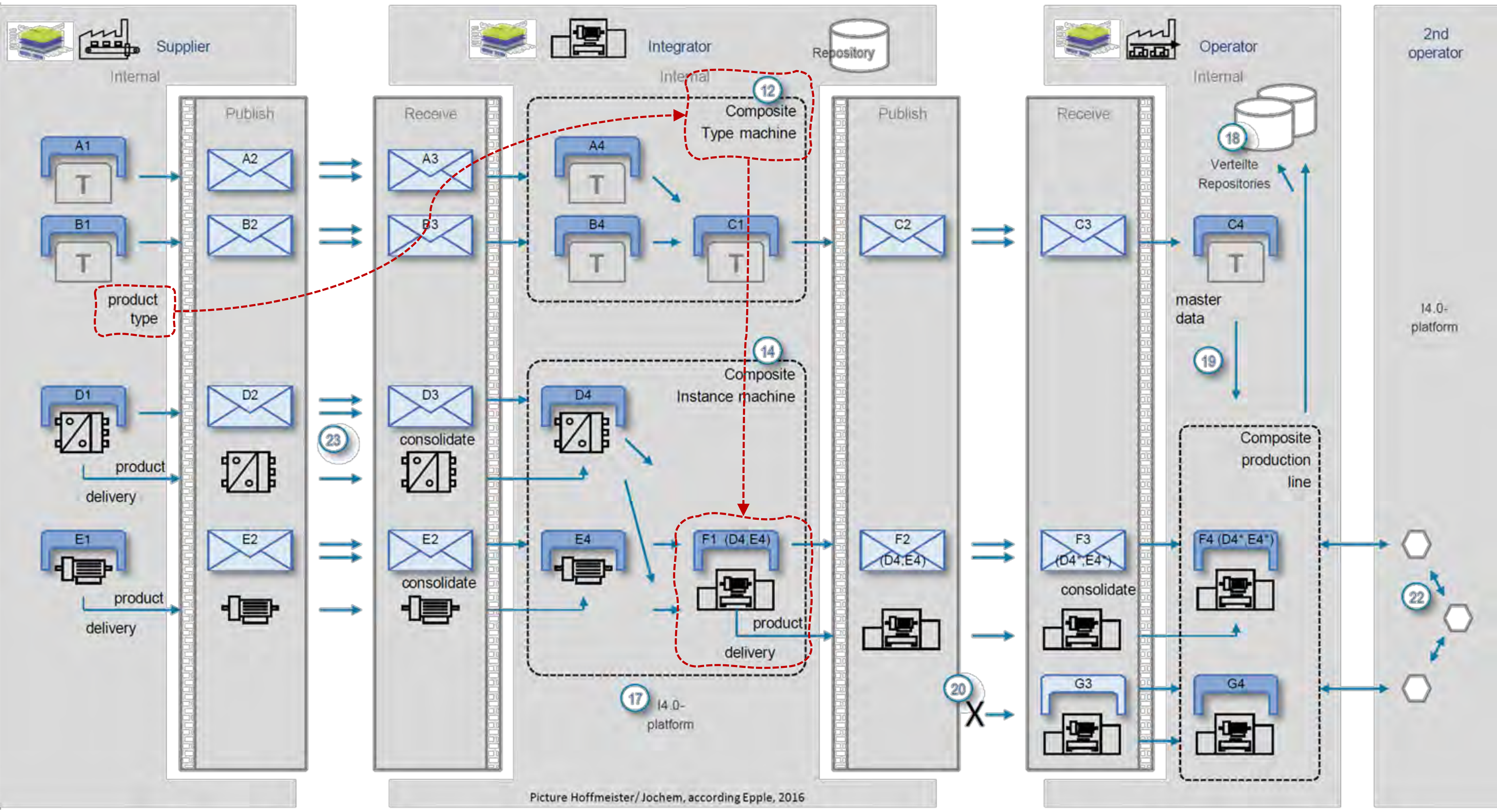
The **Administration Shell**…

- integrates the *asset into Industrie 4.0 communication*.
- provides a *controlled access to all information* of the asset.
- is the standardised and secure *communication interface*.
- can also integrate *„passive“ assets* without a communication interface, e.g. via bar codes or QR codes.
- is *addressable* in the network and *identifies the asset* unambiguously.



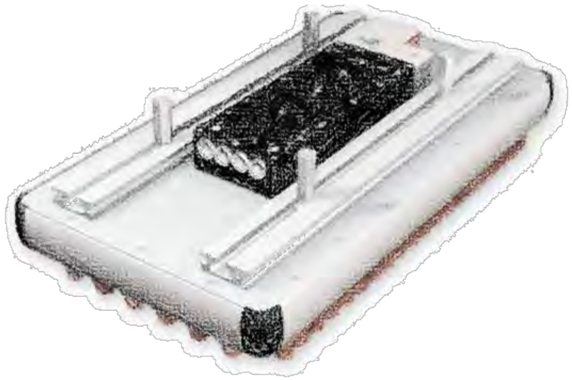
Source: Platform Industrie 4.0

Platform Industrie 4.0: the **Administration Shell**



Picture Hoffmeister/ Jochem, according Epple, 2016

A “Control Engineer” perspective



«Asset-1»: **Vacuum gripping system**

Control SW requirements:

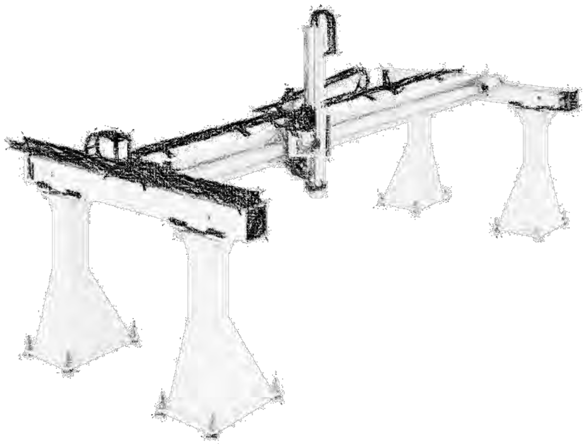
- ***I/Os***: *50x analogue and digital signals to command/monitor vacuum cups*
- ***Internal control***: *real-time adjustment of single cups to maximize grip effectiveness*
- ***Internal optimization***: *selection of the best cups to use*
- ***External configuration***: *rough shape of the object to grasp*
- ***Data analysis***: *compressed air consumption profiles*

Control HW requirements:

- ***Real-time connectivity to I/O devices (i.e. EtherCAT)***
- ***Low computation power (ARM-level)***
- ***Low cost***
- ***[OPTIONAL] Wireless connectivity***

A “Control Engineer” perspective

«Asset-2»: **Cartesian Gantry Robot**

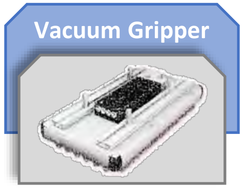


Control SW requirements:

- ***Drives***: *multi-axis for brushless motors (with encoder)*
- ***I/Os***: *lubrication, cameras, vibration sensors, safety switches, etc.*
- ***Internal control***: *trajectory generation, axis synchronization, high-level motion tasks*
- ***Internal optimization***: *dynamic obstacle avoidance*
- ***External configuration***: *gripper specifications, dynamic interactions with other robots, etc.*
- ***Data analysis***: *multi-axis/multi-point vibrations*

Control HW requirements:

- ***Real-time connectivity to I/O devices and drives (i.e. EtherCAT)***
- ***Low-jitter real-time environment***
- ***Mid-tier computation power (x86 level)***

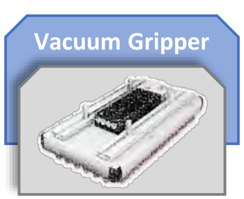


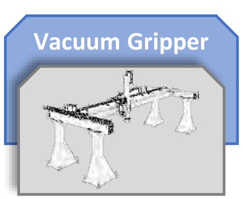
A “Control Engineer” perspective

«Asset-3a»: **Robotic Manipulator**

«Asset-3b»: **Conveying system**

«Asset-3c»: **Robotic Manipulator**





A “Control Engineer” perspective

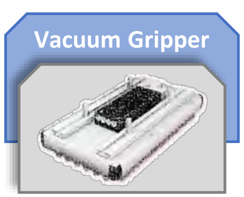
«Asset-3»: **Multi-robot material handling cell**

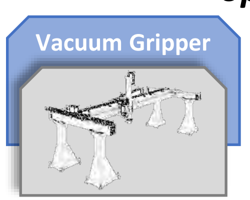
Control SW requirements:

- ***Sub-application interfacing***
- ***I/Os****: cameras, safety sensors, end-effectors, etc.*
- ***Internal control****: sub-systems orchestration*
- ***Internal optimization****: coordinated trajectory optimization under energy/performance constraints*
- ***External configuration****: material streams data, interactions with other systems, performance configuration, etc.*
- ***Data analysis****: power consumption,*

Control HW requirements:

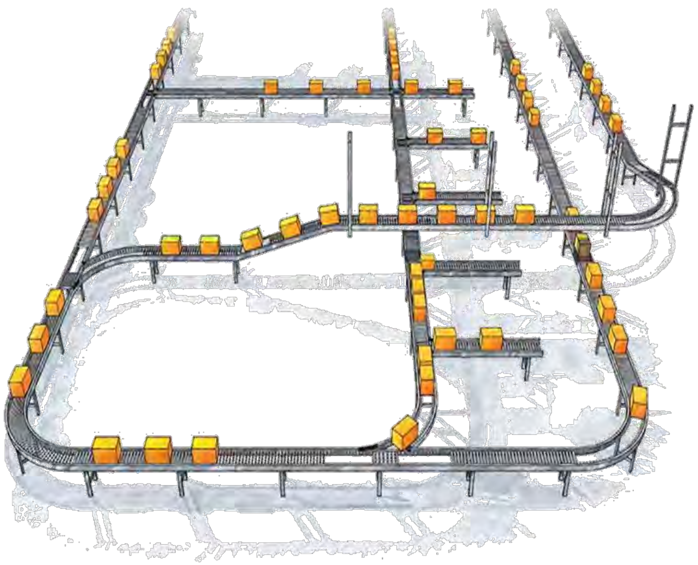
- ***Real-time connectivity to sub-controllers and to additional I/Os Mid-tier computation power (single-CPU, x86 class) for internal control***
- ***High-tier computational power (multi-CPU, x86 class) for internal optimization***





Slide courtesy Franco Antonio Cavadini CTO @ Synesis *franco.cavadini@synesis-consortium.eu*

A “Control Engineer” perspective



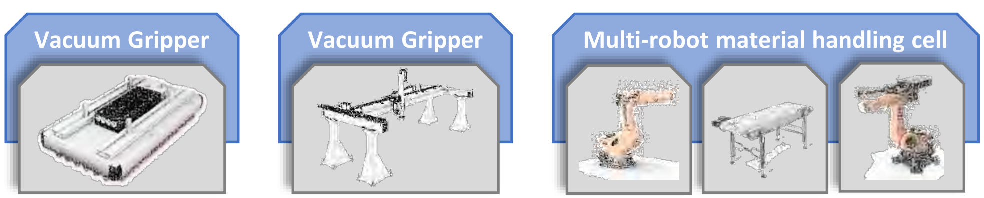
«Asset-4»: **Material streams sorting system**

Control SW requirements:

- ***Sub-application interfacing***
- ***I/Os****: cameras, safety sensors, HMIs, etc.*
- ***Internal control****: material streams orchestration*
- ***Internal optimization****: hybrid-MPC real-time optimal scheduler*
- ***External configuration****: daily production planning.*
- ***Data analysis****: power consumption, productivity KPIs*

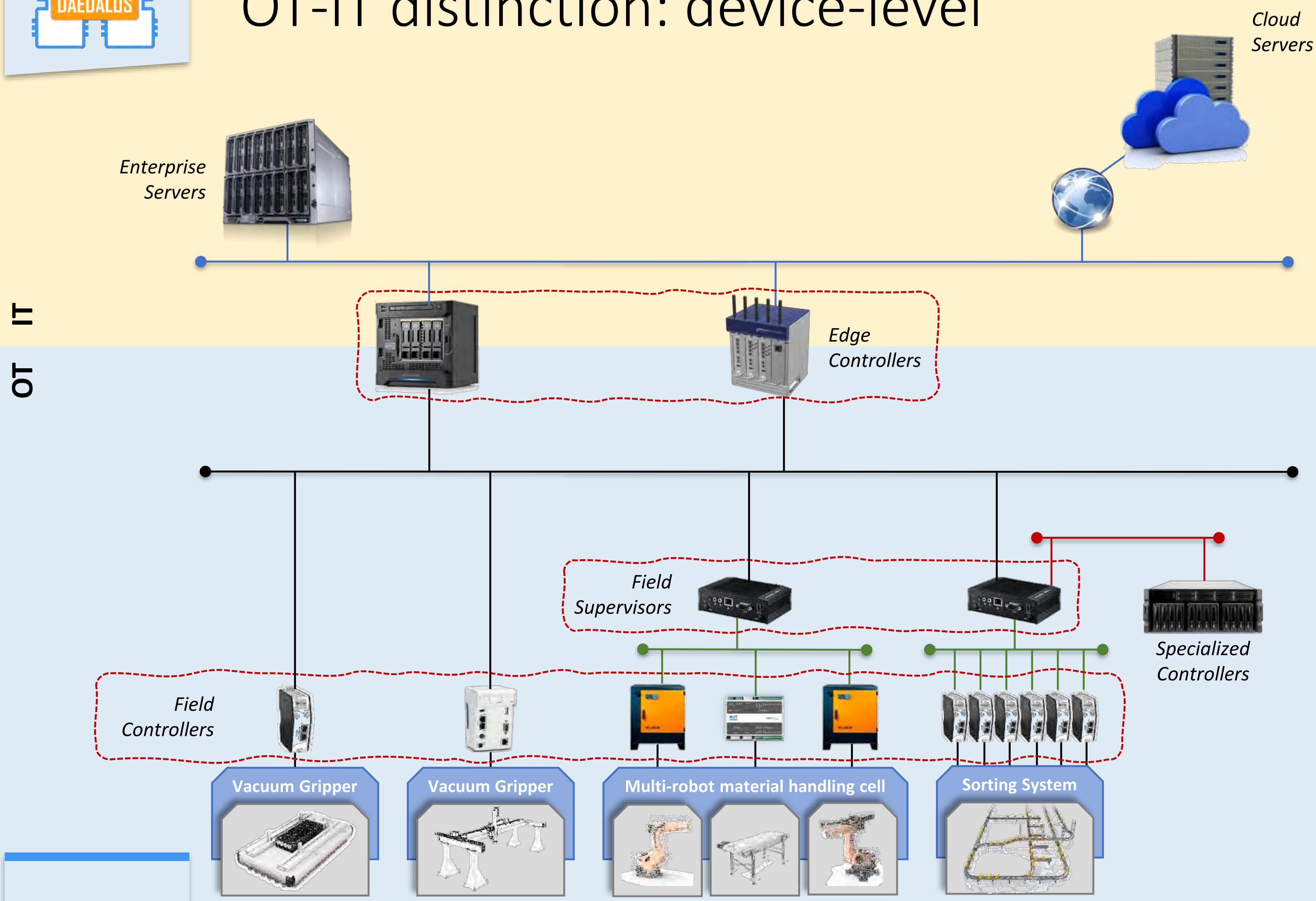
Control HW requirements:

- ***Real-time connectivity to sub-controllers and to additional I/Os (i.e. EtherCAT)***
- ***Mid-tier computation power (single-CPU, x86 class) for orchestration***
- ***High-tier computational power (multi-CPU, x86 class + GPU) for internal optimization***



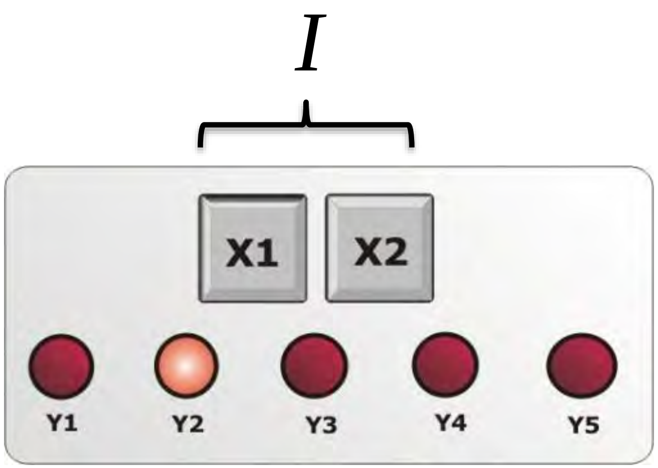
Slide courtesy Franco Antonio Cavadini CTO @ Synesis *franco.cavadini@synesis-consortium.eu*

OT-IT distinction: device-level



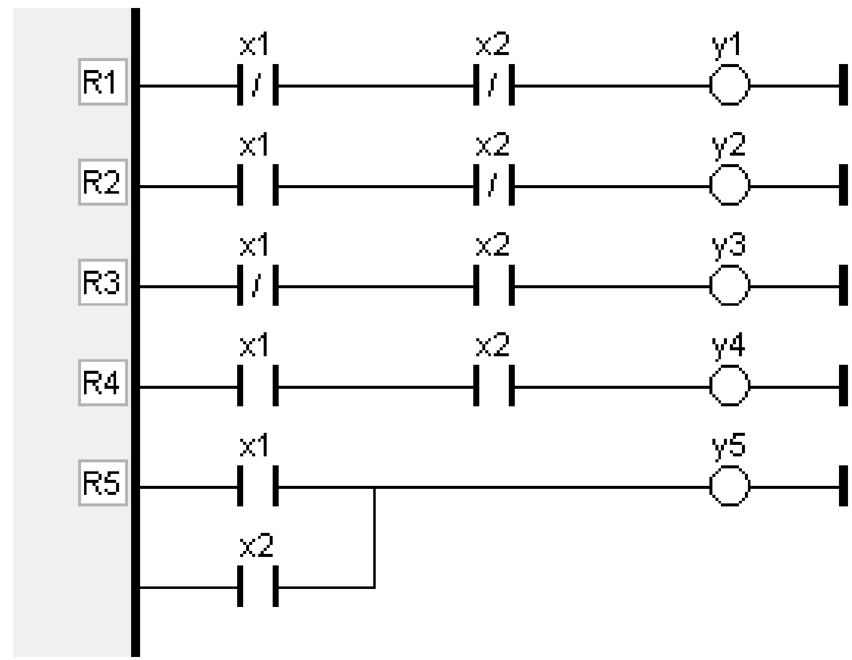
Application Logic Design

Combinatorial Model



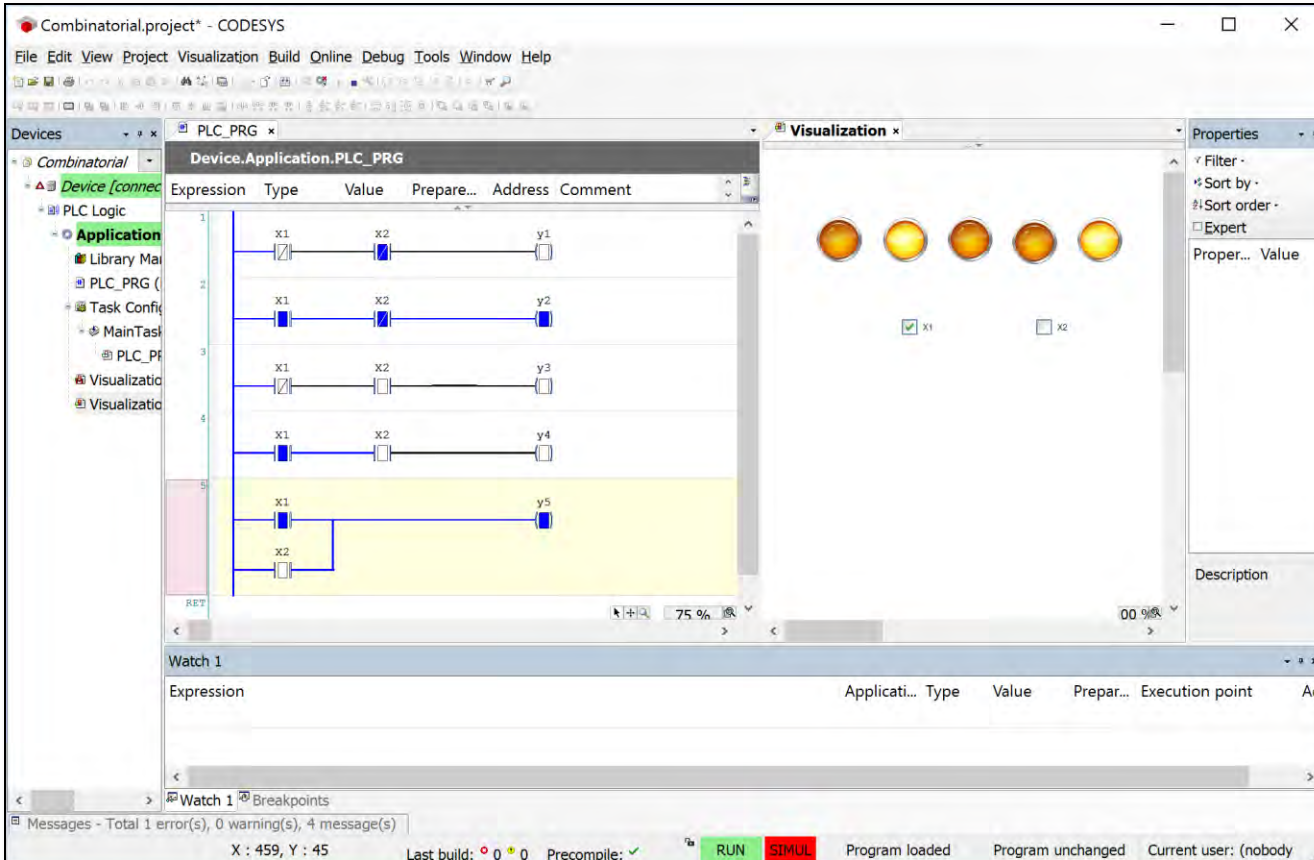
Combinatorial:

$O(i)=\mathbf{F}(I(i))$

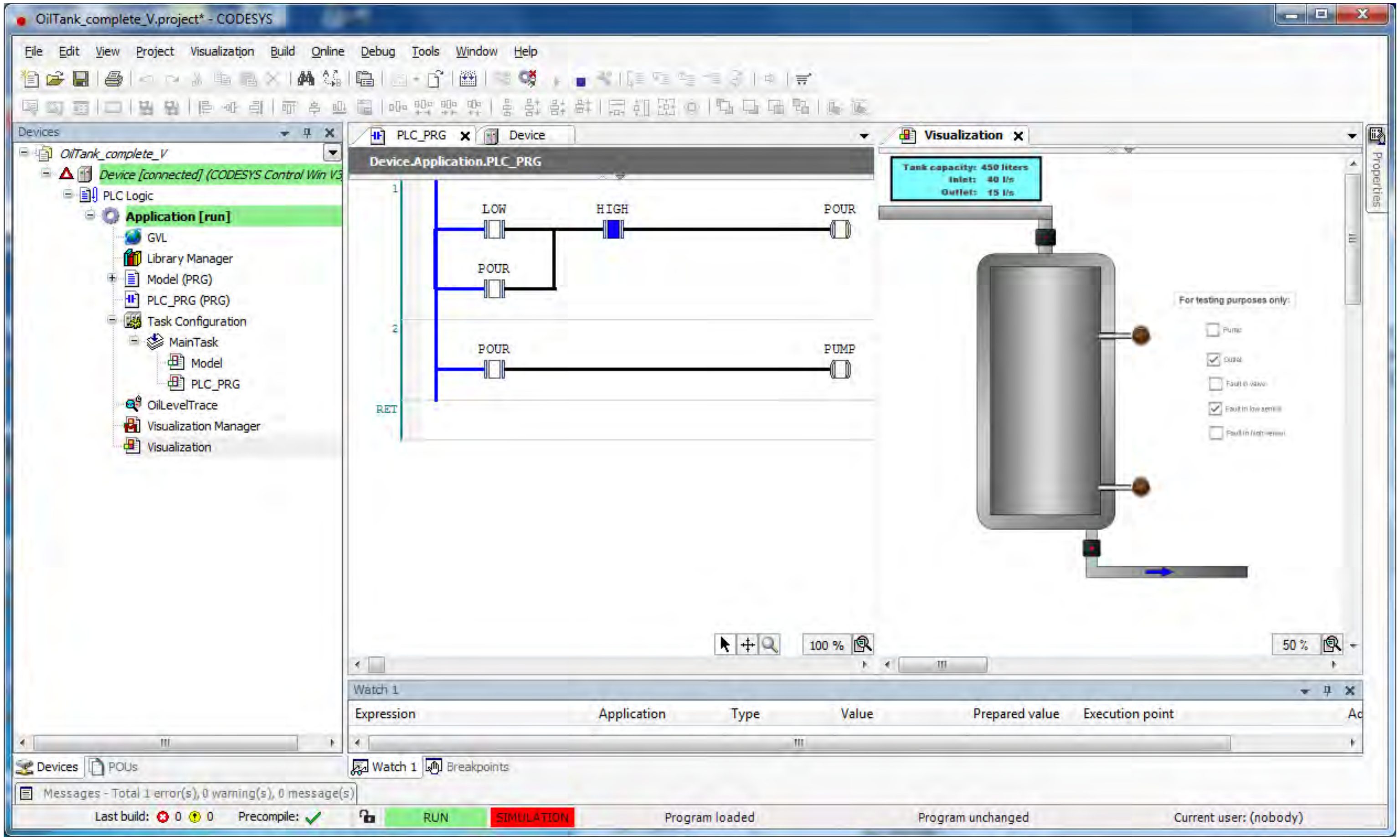


O $\left\{ \begin{array}{l} \text{Y1:= NOT X1 AND NOT X2;} \\ \text{Y2:= X1 AND NOT X2;} \\ \text{Y3:= NOT X1 AND X2;} \\ \text{Y4:= X1 AND X2;} \\ \text{Y5:= X1 OR X2 ;} \end{array} \right.$

Implementation in CoDeSys



Ladder Logic for Convenient Diagnostics



Aalto University

Example: Combinatorial logic design

Example: Lubricating Oil dispense from a tank

Inputs

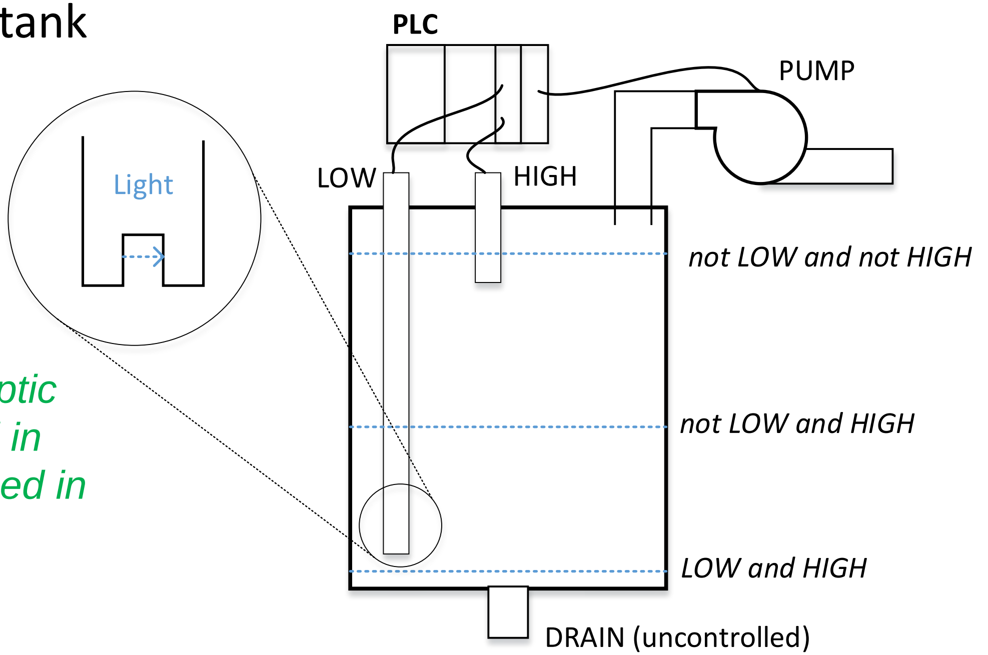
LOW - Low level sensor

HIGH - High level Sensor

Both of the inputs are normally closed fiber-optic level sensors. When they are NOT immersed in liquid they will be ON. When they are immersed in liquid they will be OFF.

Outputs

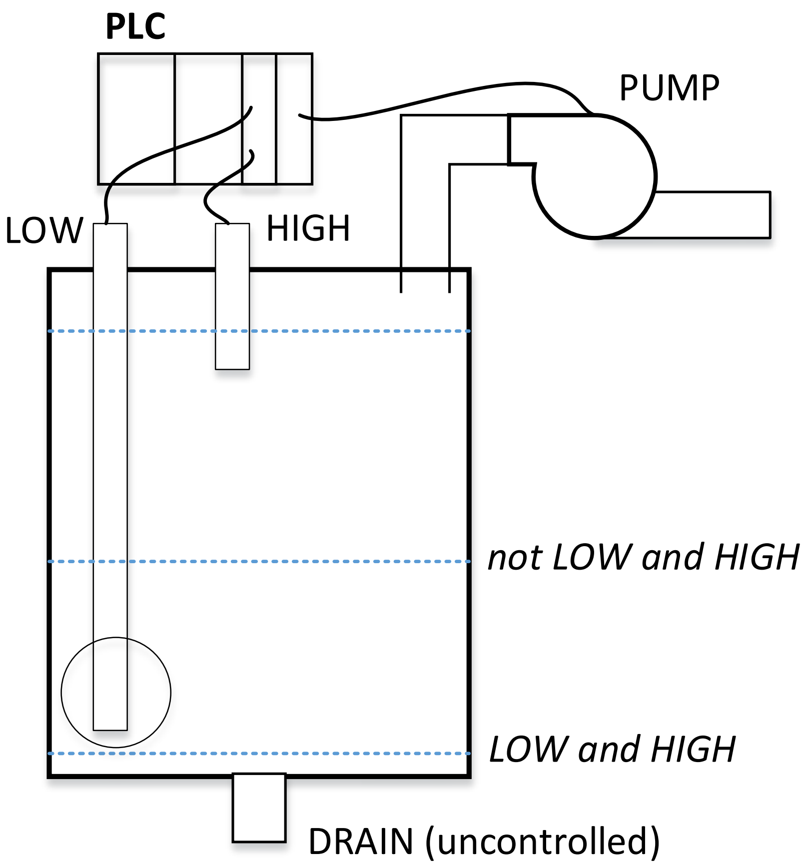
PUMP



Requirement:

- Maintaining the level between high and low

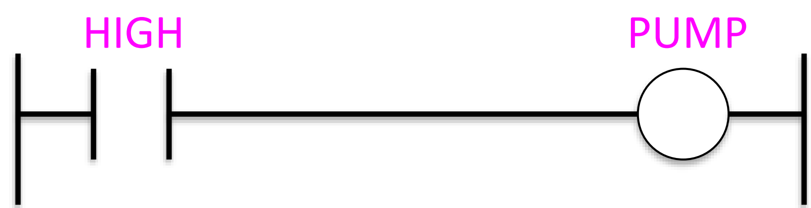
Combinatorial logic design



LOW= TRUE when no oil at the lower level (i.e. tank is empty).

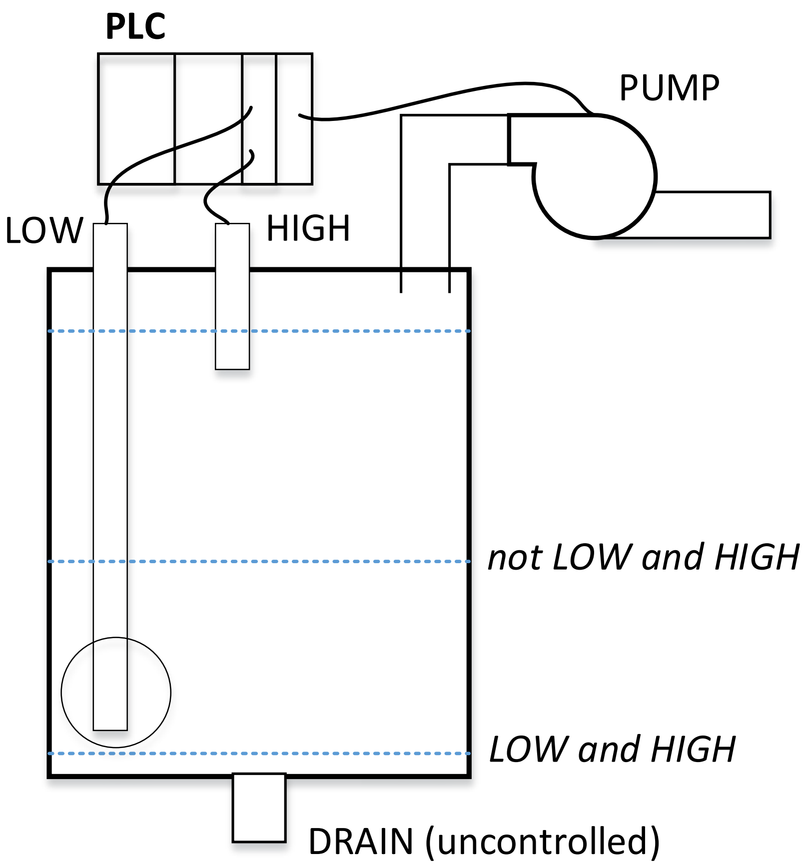
HIGH= TRUE when no oil at the high level.

The pump needs to be turned on if oil level is lower than high.



This solution, however has a major drawback. **WHICH?**

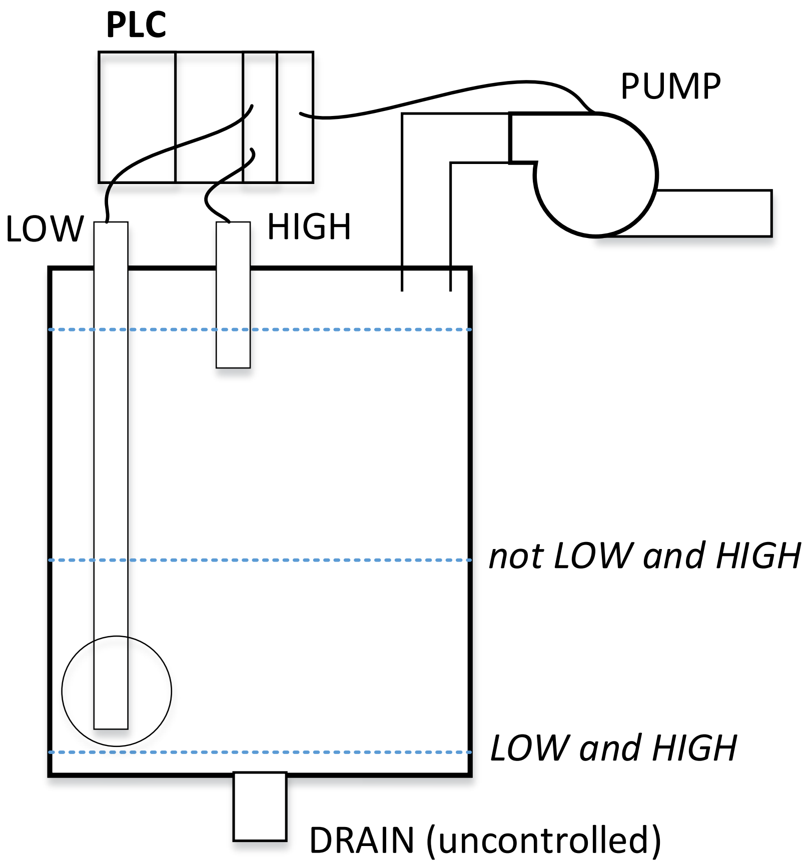
State variable introduced

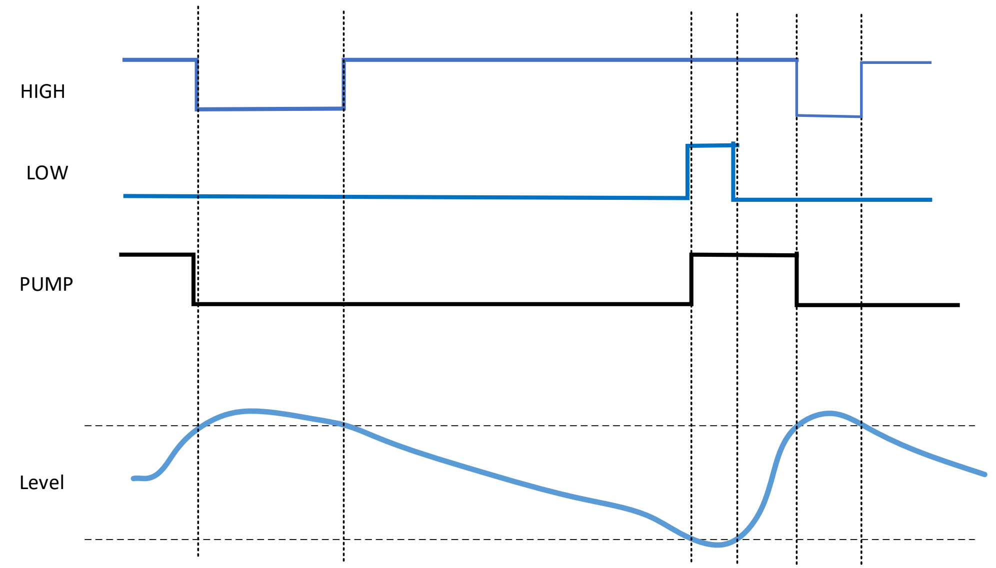


A better solution is to pour oil continuously from the lower level to the top.

Introduce internal variable (state) POUR.

State-based design





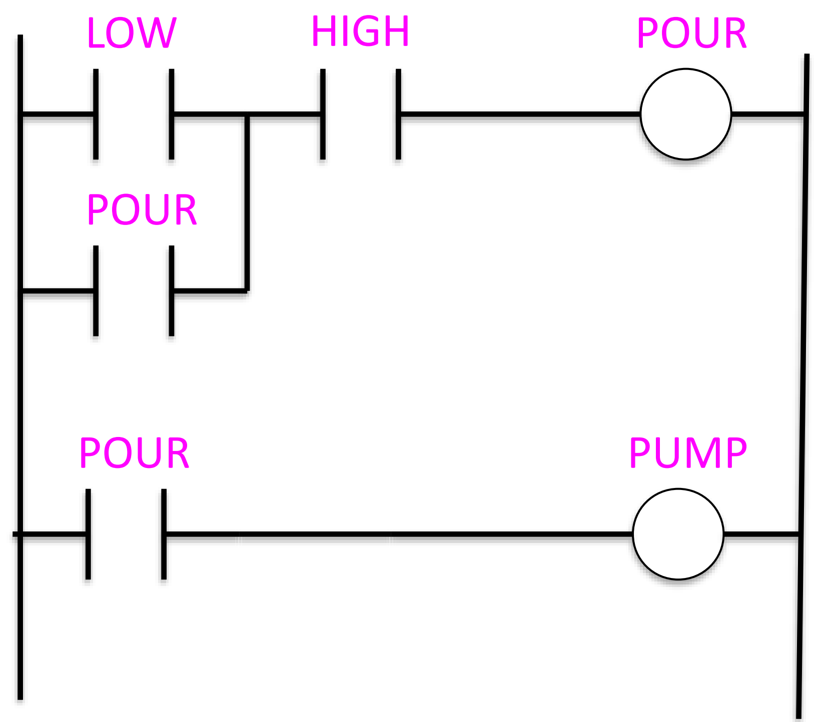
Implementation in PLC

// State transition function

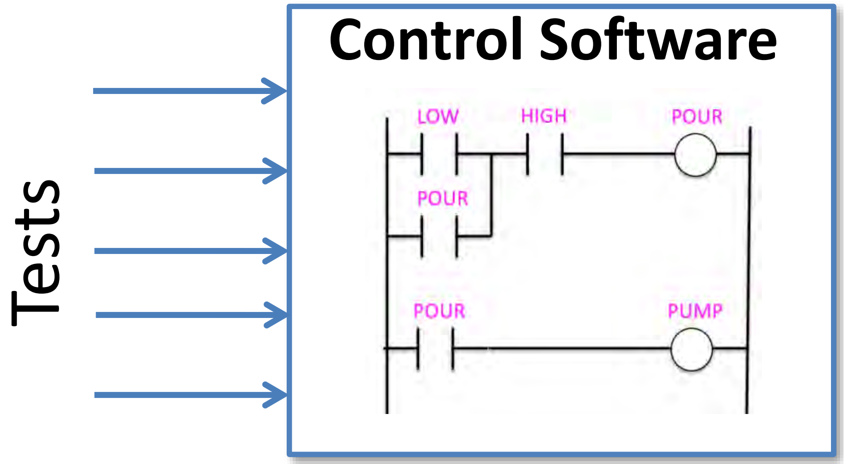
S1:= S1 /LOW + S2 /HIGH;
S2:= S2 HIGH + S1 LOW;

// using the following facts:
// LOW = LOW & HIGH
// S1 = not S2

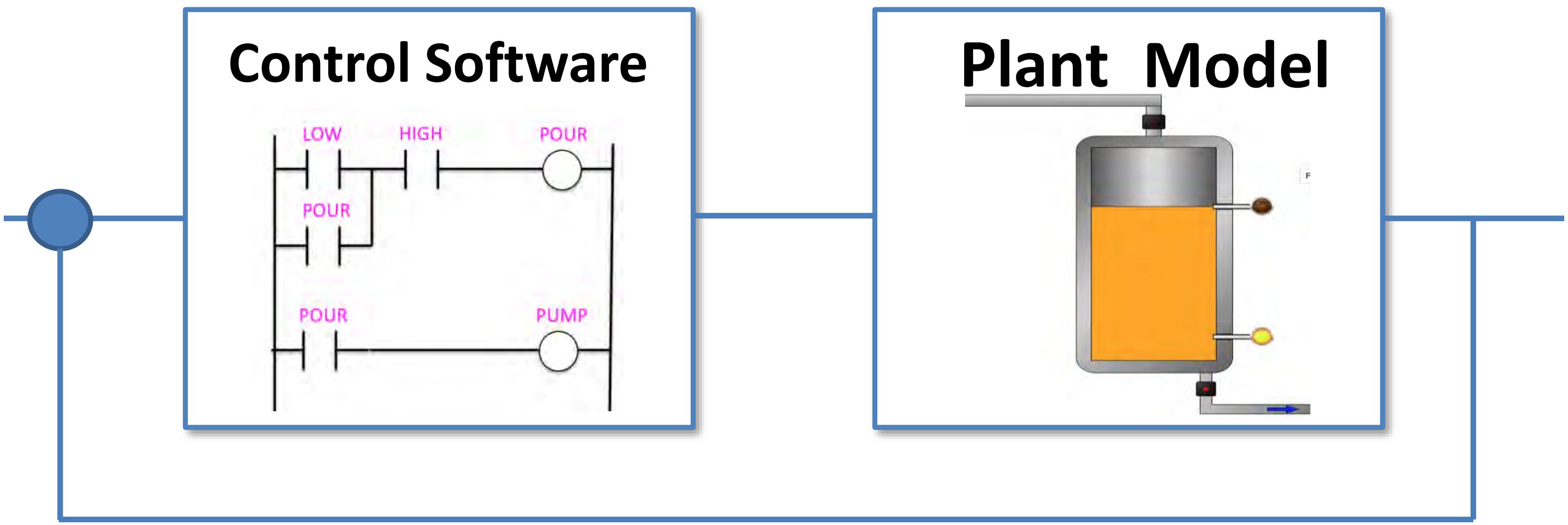
S2 := S2 HIGH + /S2 LOW =
= S2 HIGH + /S2 LOW HIGH =
= (S2 + LOW) HIGH
// Change S2 to POUR



Verification of Automation Software



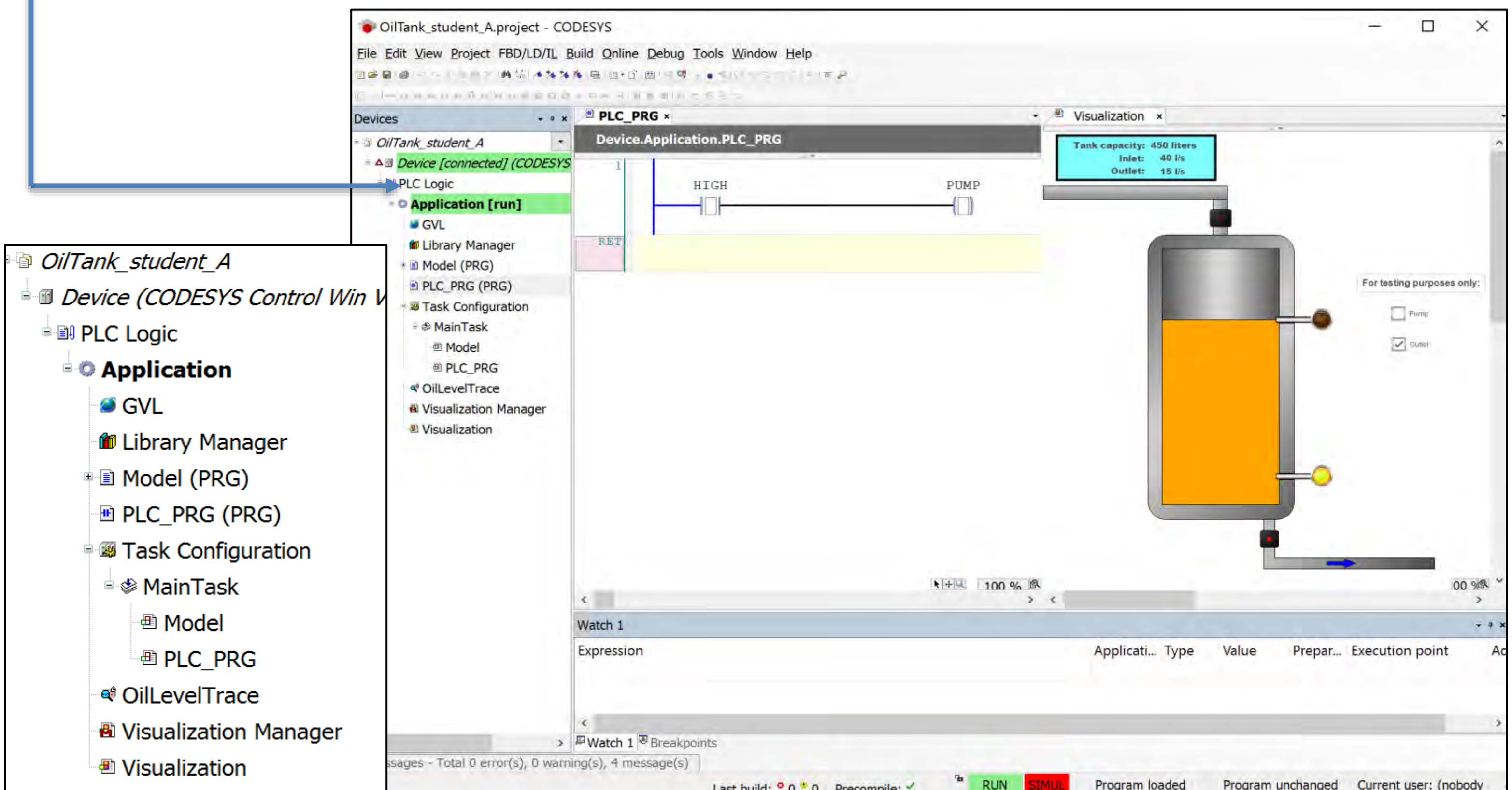
Closed-Loop Simulation Environment



Implementation in CoDeSys

GVL

```
VAR_GLOBAL
    // Sensors from the model.
    LOW:BOOL;   // Normally closed sensor.
    HIGH:BOOL;  // Normally closed sensor.
    // Actuators of the model.
    PUMP: BOOL; // Turn on the pump (TRUE) or off (FALSE).
END_VAR
```



Features of the future PLC technology

- Programs specify timing, not priorities.
- Timing as a logical notion (logical synchrony).
- Deterministic concurrency and multicore execution.
- Event-triggered computation as well as periodic.
- Synchronized clocks.
- Common logical time origin across a distributed system.
- Logical clock domains.
- Message passing to replace shared variables.
- Less reliance on polling.
- Sandboxing.
- Private memories.
- Temporal isolation.
- Authentication and authorization.
- Encrypted communication.
- Mixed criticality networking.
- Faithful virtual prototypes.

Sehr, M.A., Lohstroh, M., Weber, M., Ugalde, I., Witte, M., Neidig, J., Hoeme, S., Niknami, M. and Lee, E.A., 2020. **Programmable Logic Controllers in the Context of Industry 4.0**. *IEEE Transactions on Industrial Informatics*.

What to remember?

- Why the ladder logic is used in automation systems programming?
- What is the advantage of flat architecture compared to the "ICT pyramid"?
- Why intelligent machines are needed at the factory floor?
- What is main difference of combinatorial logic from state-based logic?
- What are pros and contras of using simulation in the loop?