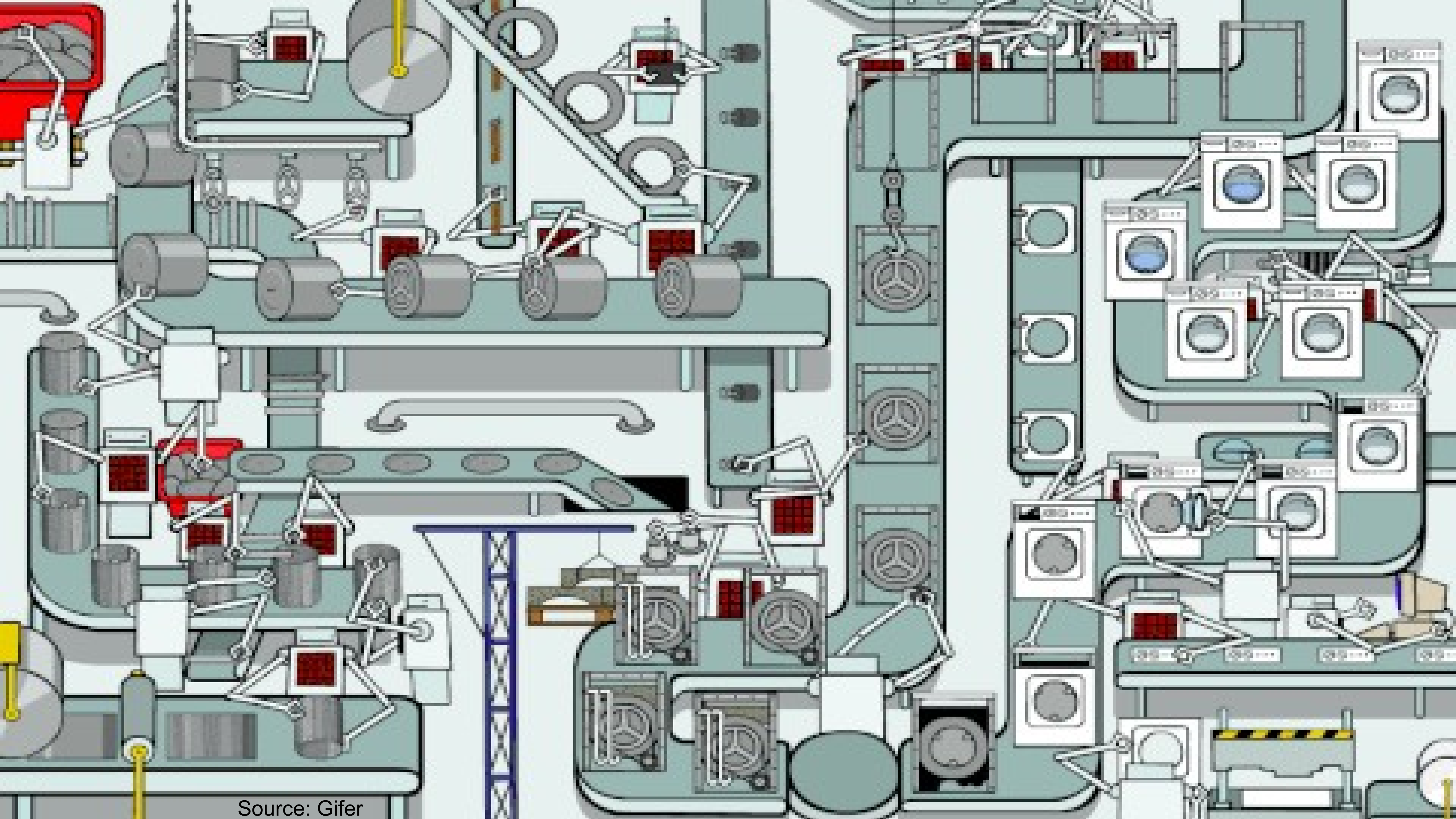
LINZ INSTITUTE
OF TECHNOLOGY

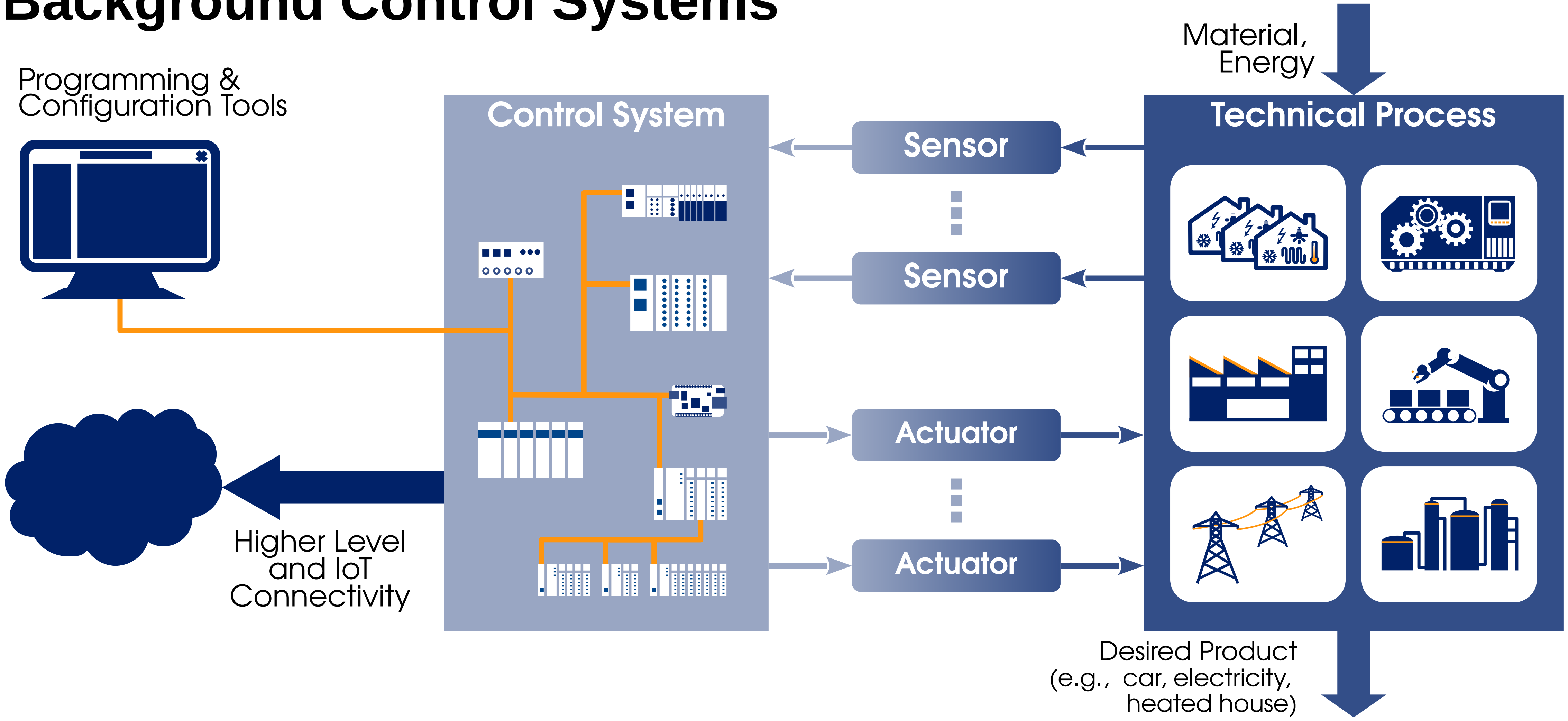
Modeling Distributed Production Control Systems with IEC 61499 and Eclipse 4diac™

Taming the Control Software Dragon!

Univ.-Prof. Dr. Alois Zoitl
LIT | Cyber-Physical Systems Lab
Johannes Kepler University Linz


Source: Gifer

Background Control Systems



Problem: Software Development Effort

„... increases the software-engineering portion of the overall Manufacturing costs of a machine: Starting from currently 50% share for electronics and software the share will rise in 2020 up to 80%. “

translated from IEE 01-2006

„We have so far mastered most topics and could ***save*** *up to* ***70%*** *of the engineering effort.* What makes us still problems is the **software effor**t.

translated from SPS Magazin 08-2012

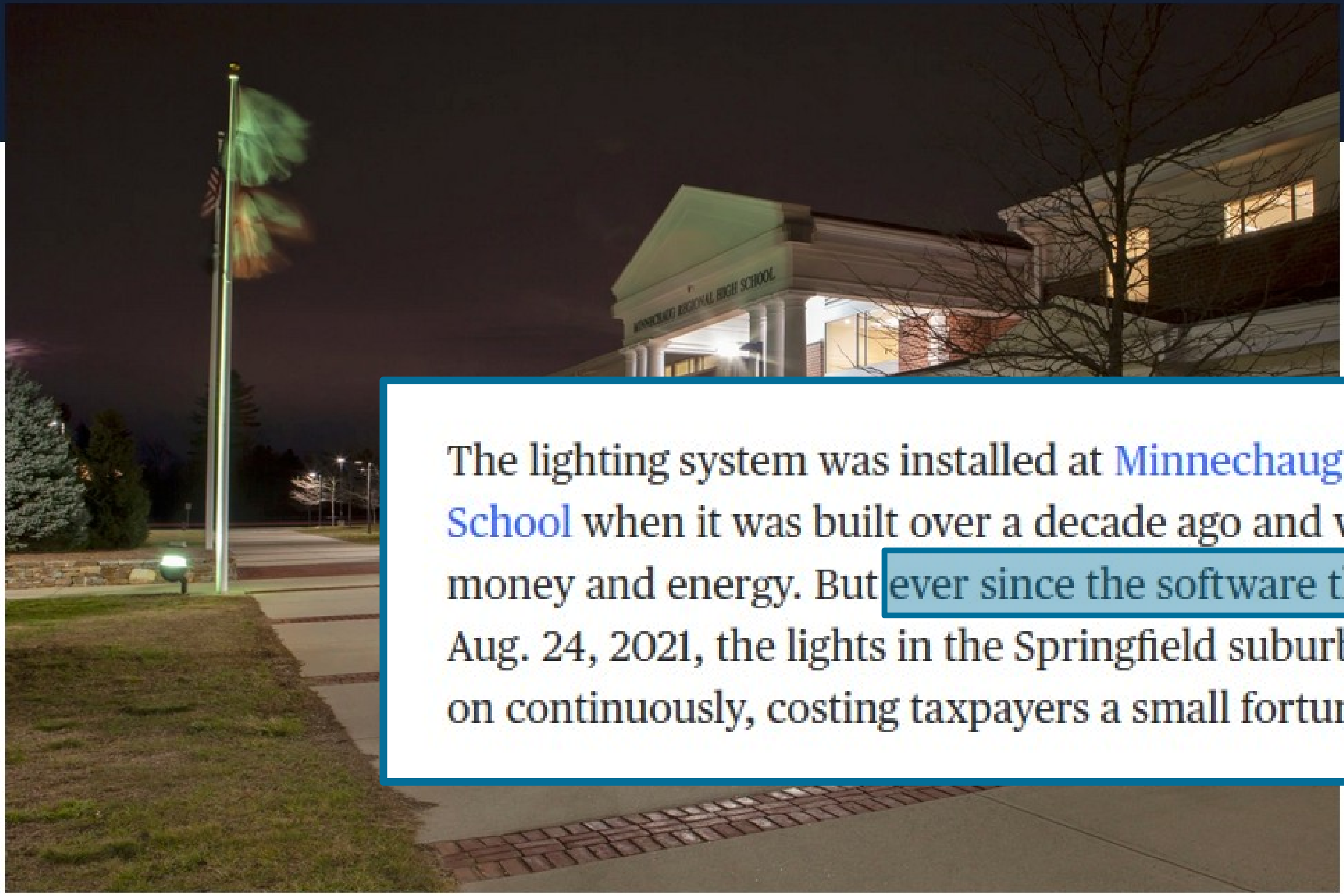
NEWS The lights have been on at a Massachusetts school for over a year because no one can turn them off SHARE & SAVE —

EXCLUSIVE

U.S. NEWS

The lights have been on at a Massachusetts school for over a year because no one can turn them off

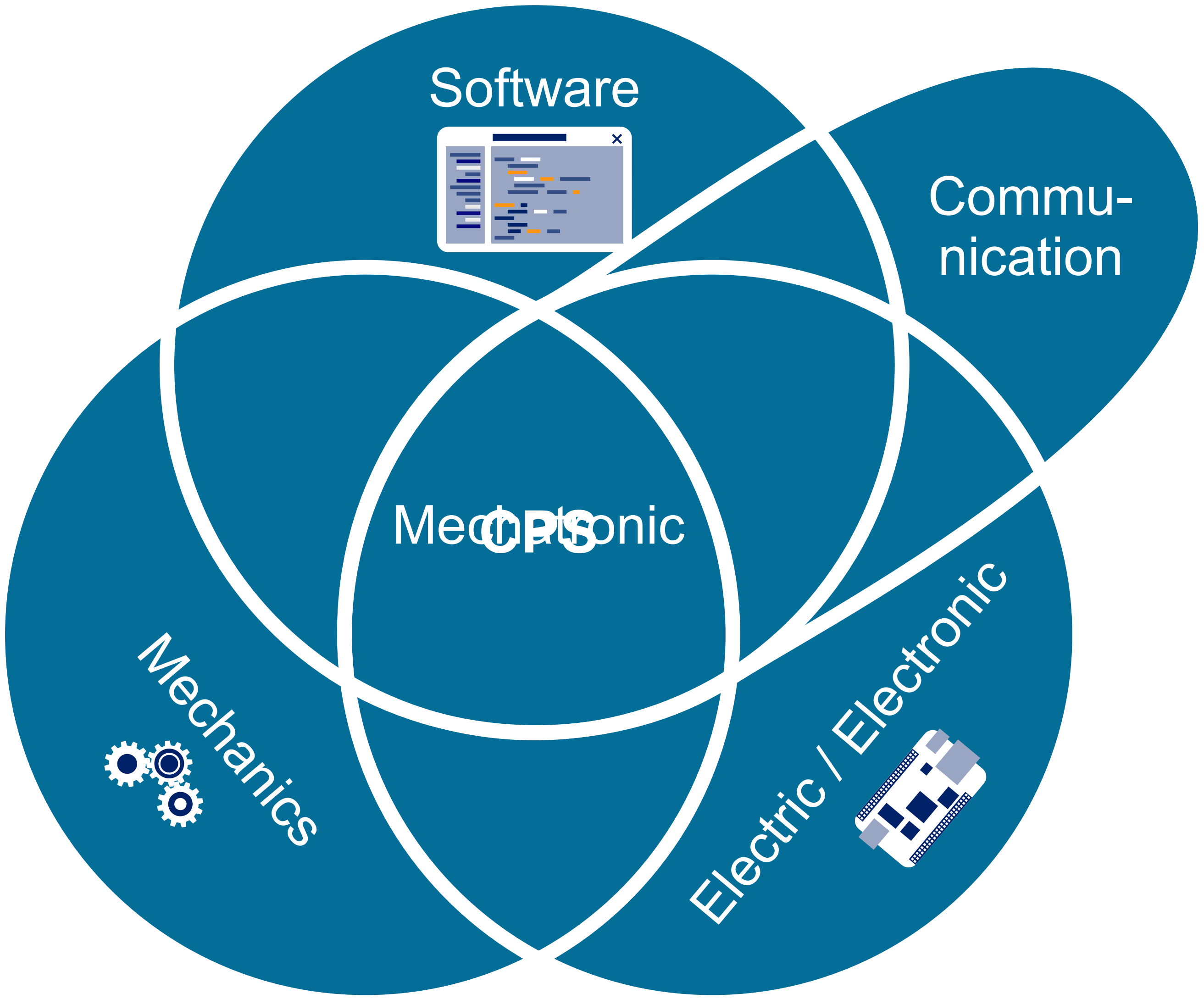
Blame it on the pandemic and "supply chain problems," says the school district's assistant superintendent of finance.



The lighting system was installed at Minnechaug Regional High School when it was built over a decade ago and was intended to save money and energy. But ever since the software that runs it failed on Aug. 24, 2021, the lights in the Springfield suburbs school have been on continuously, costing taxpayers a small fortune.

All the lights at Minnechaug Regional High School in Wilbraham, Mass., have been on since Aug. 24, 2021. Matt Nighswander / NBC News

Source: Wikipedia


Software
Commu-
nication
Mechatronic
CPS
Mechanics
Electric / Electronic

IEC 61499
Domain-specific Modeling Language for **Distributed** Industrial Process Measurement and Control Systems

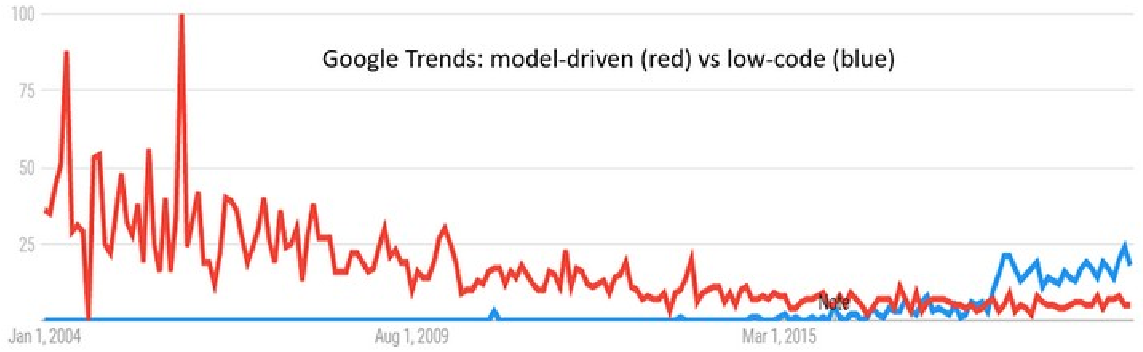
How the UML diagram describes the software

How the code is actually written

Quelle: 9Gag

Low-code vs model-driven: are they the same?

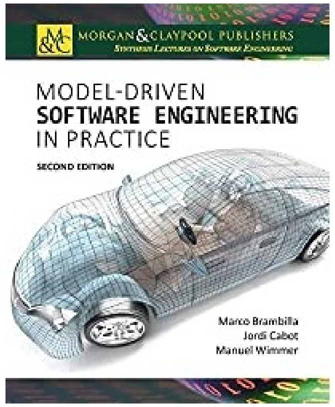
by Jordi Cabot | Aug 23, 2020 | Model-driven Engineering, opinion | 12 comments



Since low-code became the new buzzword, I wondered whether there was anything really different in the low-code movement compared to what we used to call model-driven engineering/development. The 1st Low-code workshop (part of the Models 2020 conference) was the perfect excuse to take some time to reflect and write down my thoughts on this topic.

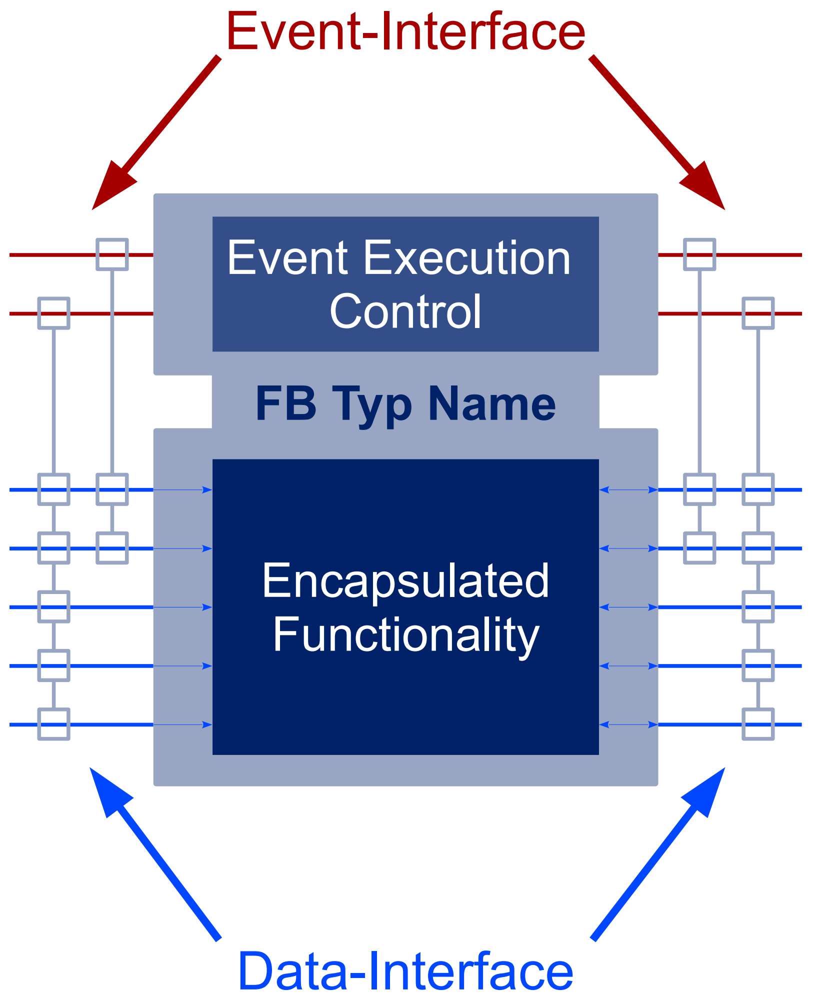
This is the current "raw" version of the paper I wrote for the workshop. The paper was accepted and I need to prepare in the next days a new version of the text that takes into account the feedback from the reviewers. The goal of this post is to try to collect additional feedback from You that I can also integrate into the new version of the paper or that I can use for the presentation of this work at the workshop. **I do believe this (the positioning of low-code in the model-driven world) is a discussion we need to have as a community**. Even if we don't reach any consensus.

DISCLAIMERS: 1 – This is a short position paper and should be read and interpreted as such. 2 – It is probably controversial. If you feel offended when reading it, I did a good job. I think the point of position papers is making strong and bold statements that help to start a discussion. 3 – It is difficult to "scientifically" compare two terms when one of them "low-code" is not scientifically defined but needs to be inferred from studying the set of tools that call themselves as such.


MORGAN & CLAYPOOL PUBLISHERS
MODEL-DRIVEN SOFTWARE ENGINEERING IN PRACTICE
SECOND EDITION
Marco Brambilla
Jordi Cabot
Manuel Wimmer

Core Element: Function Block

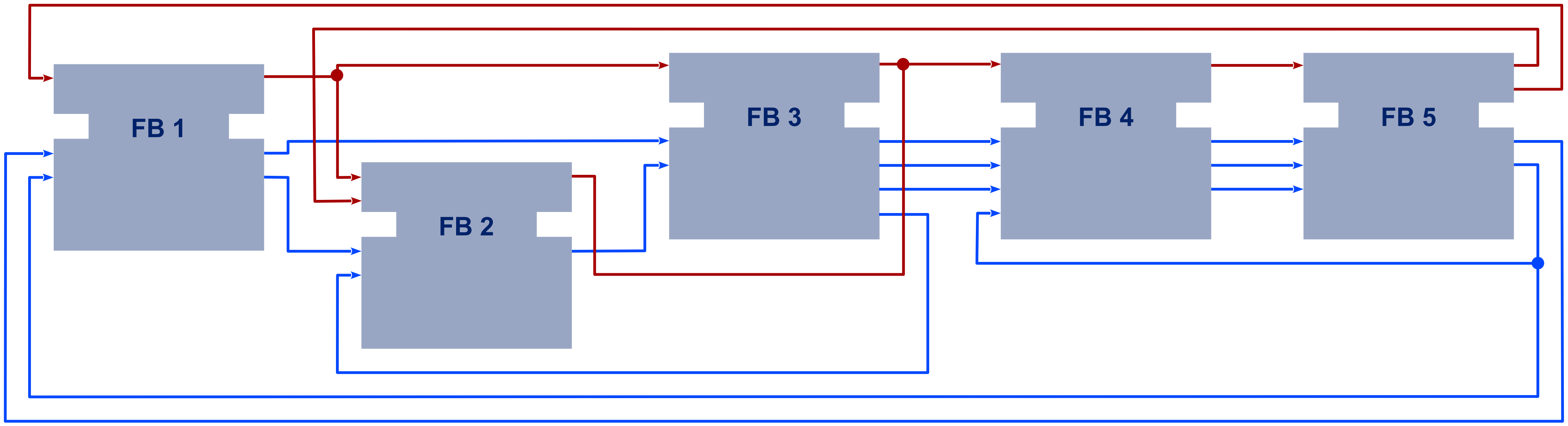
- Function Blocks extended with event interface
- Pure **event-driven** execution model
- Data types based on **IEC 61131-3**
- Focus on **encapsulation** and **reuse**
- No global or directly addressed variables
- Hardware access with special function block type: **Service Interface Function Block**



IEC 61499 Application Model

PIM

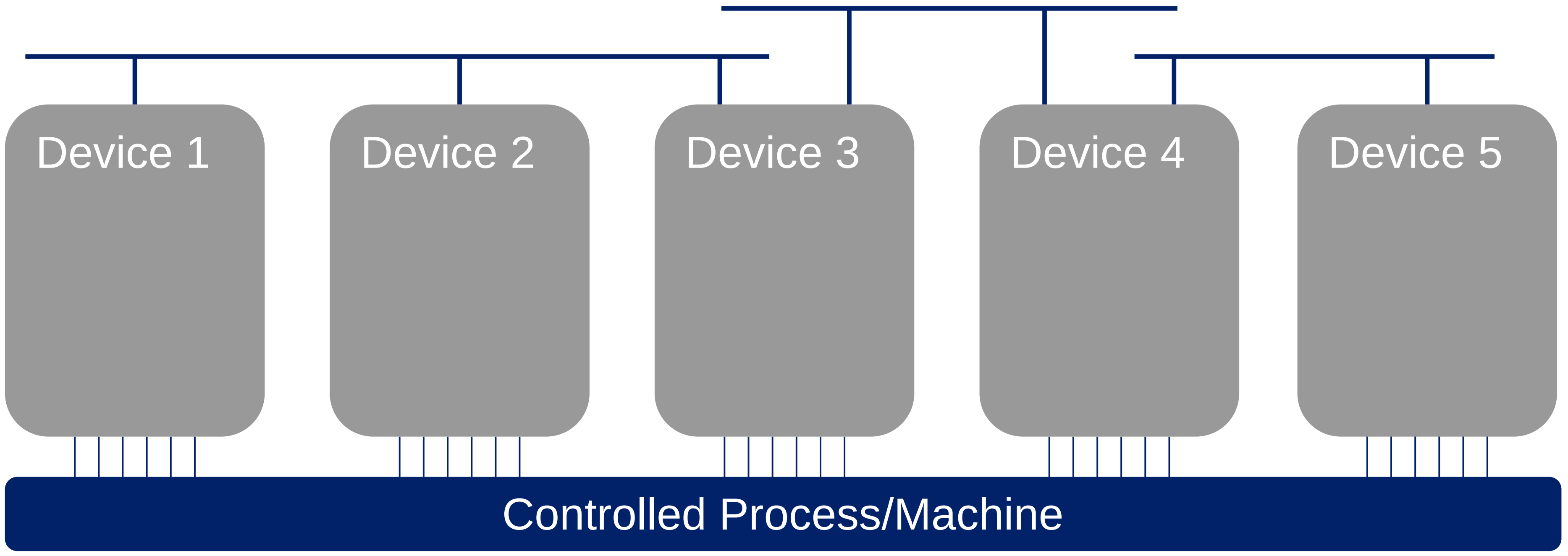
- Function Blocks instances
- Data connections
- Event connections



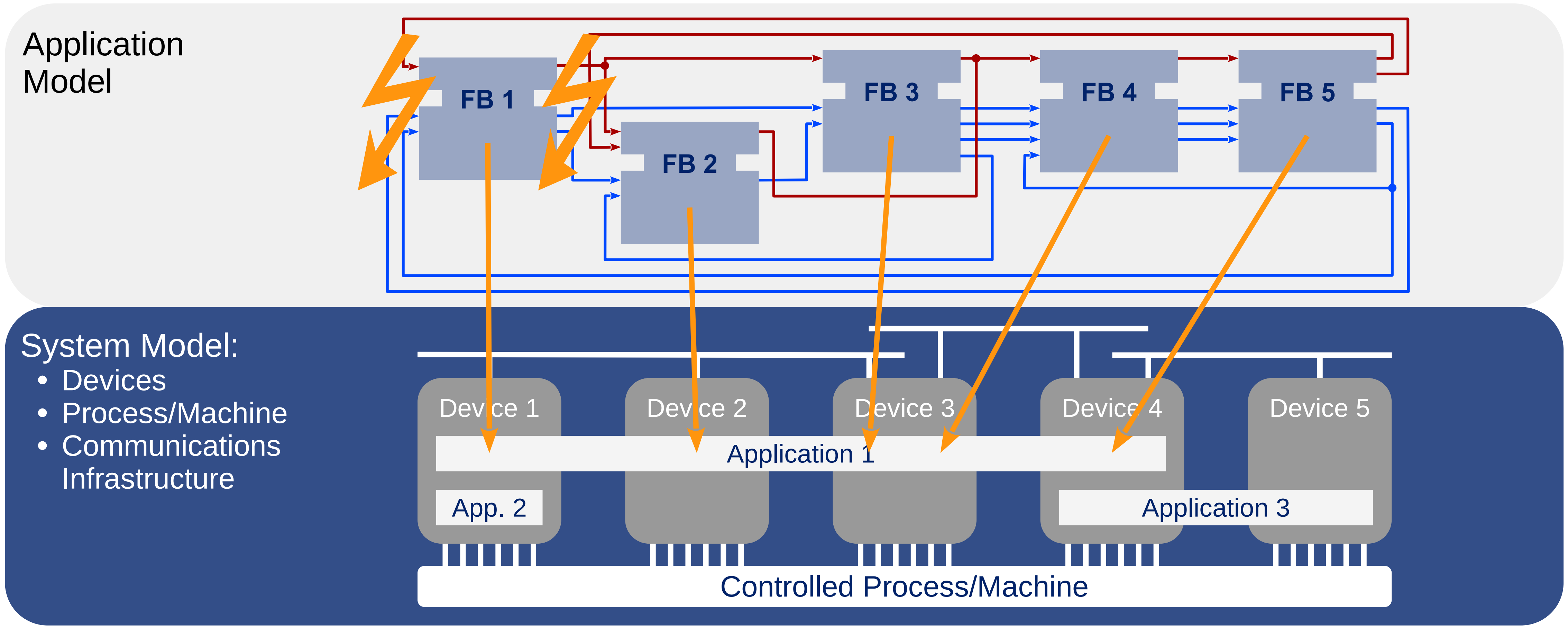
System Model

PM

- Devices
- Process/Machine
- Communication infrastructure

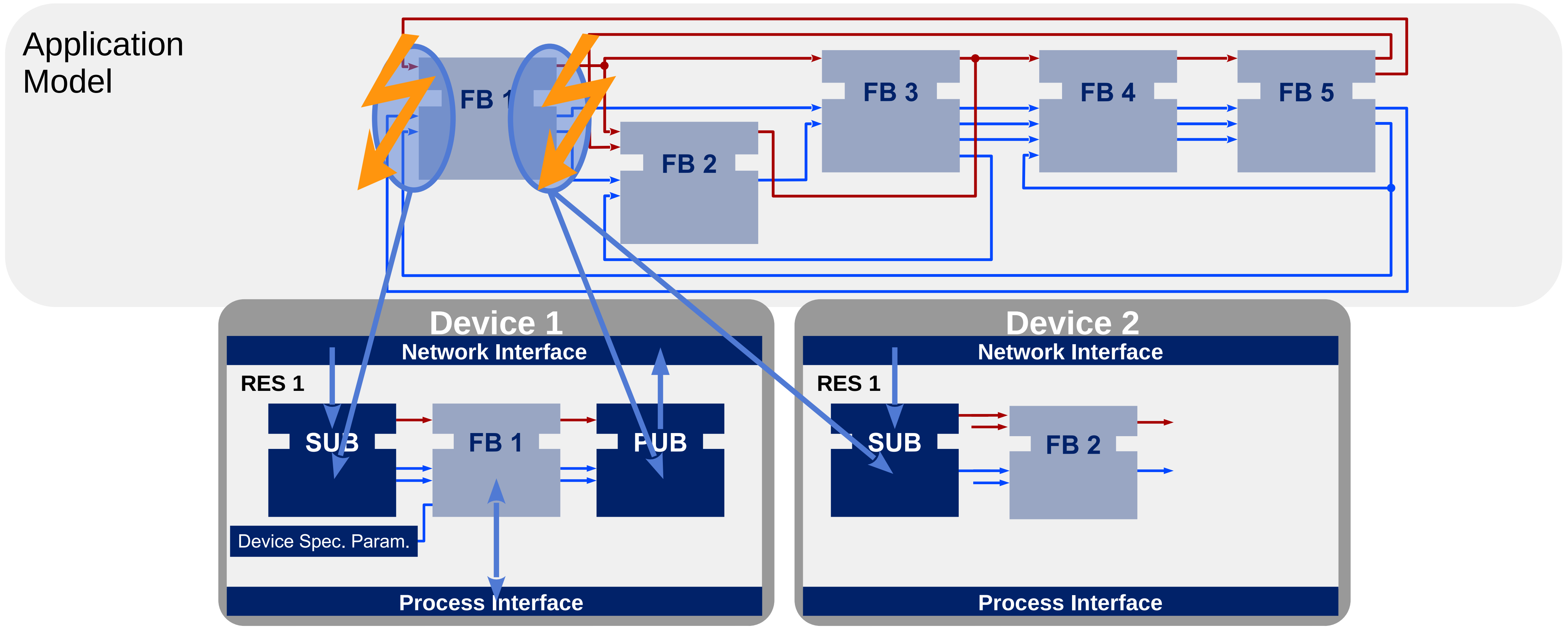


Distribution Model

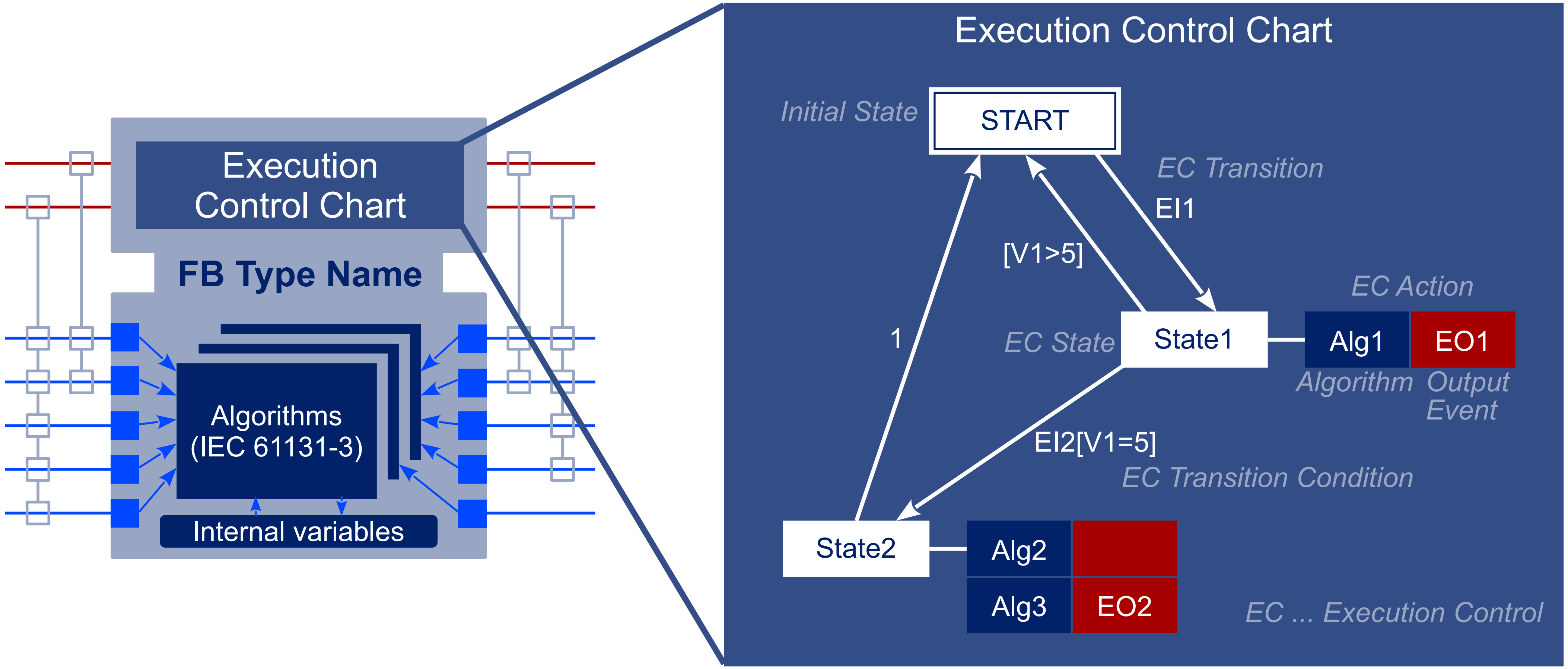


Device Specific Adjustments and Parameters

PSM



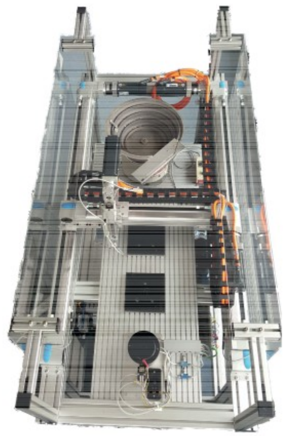
Basic Function Block



Comparing IEC 61113-3 and IEC 61499 with Code Metrics

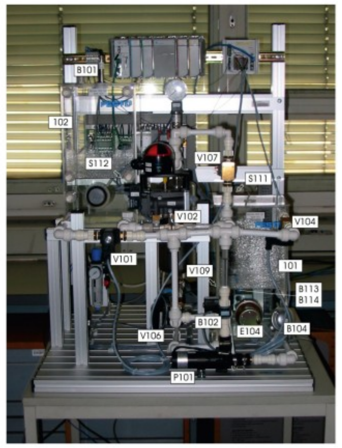
Sequential process

	IEC 61131-3	IEC 61499
Program vocabulary	587	217
Program length	581	439
Estimated length	4839.35	1503.98
Purity ratio	8.33	3.43
Program volume	5343.58	3407.32
Program difficulty	109.50	46.49
Program effort	585122.33	158408.48
Cyclomatic Complexity	33	45 (33)

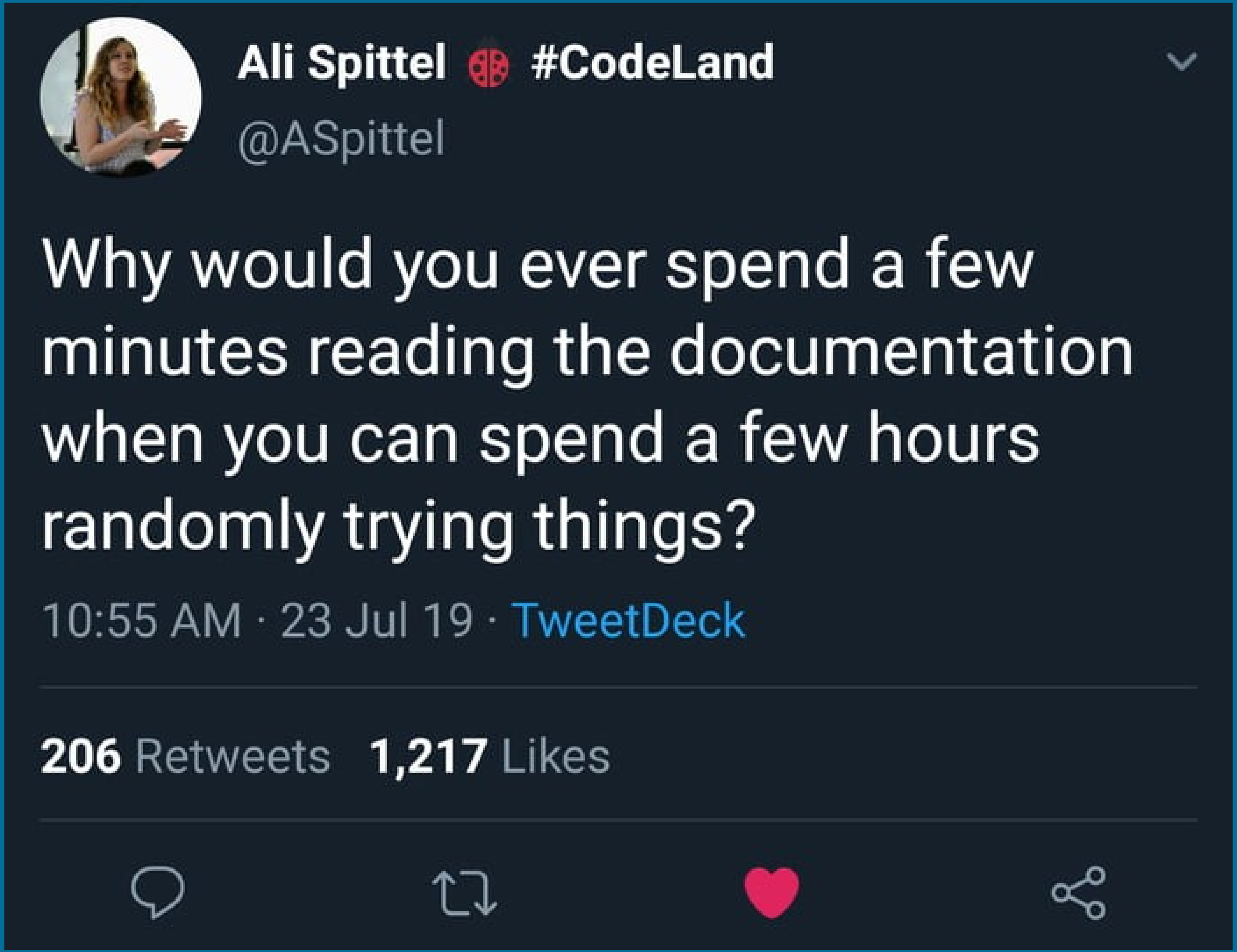


Continuous process

	IEC 61131-3	IEC 61499
Program vocabulary	68	82
Program length	61	117
Estimated length	346.63	455.60
Purity ratio	5.68	3.89
Program volume	371.34	743.83
Program difficulty	15	11.77
Program effort	5570.03	8758.04
Cyclomatic Complexity	3	6 (3)



P. Gsellmann, M. Melik-Merkumians and G. Schitter, "Comparison of Code Measures of IEC 61131–3 and 61499 Standards for Typical Automation Applications," 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, 2018, pp. 1047-1050.



Ali Spittel 🐞 #CodeLand
@ASpittel

Why would you ever spend a few minutes reading the documentation when you can spend a few hours randomly trying things?

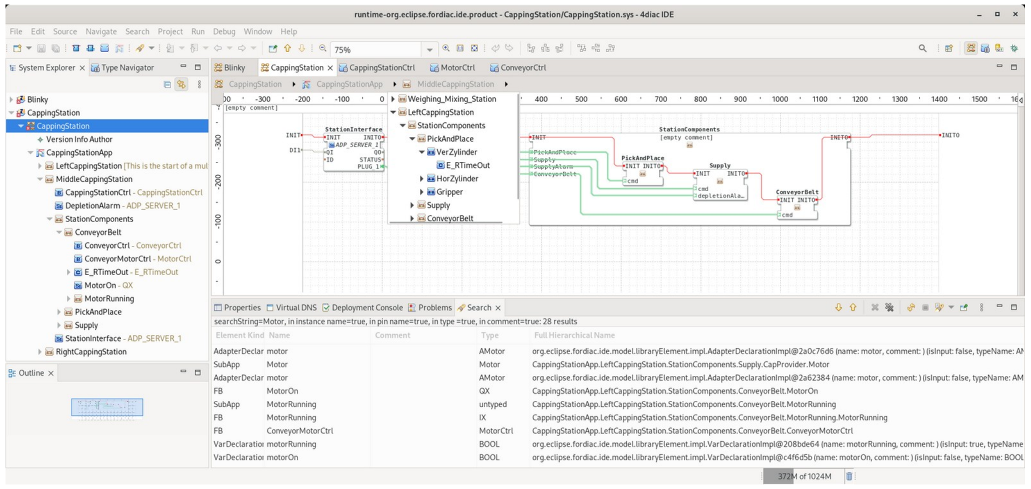
10:55 AM · 23 Jul 19 · TweetDeck

206 Retweets **1,217** Likes

Eclipse 4diac

https://www.eclipse.org/4diac

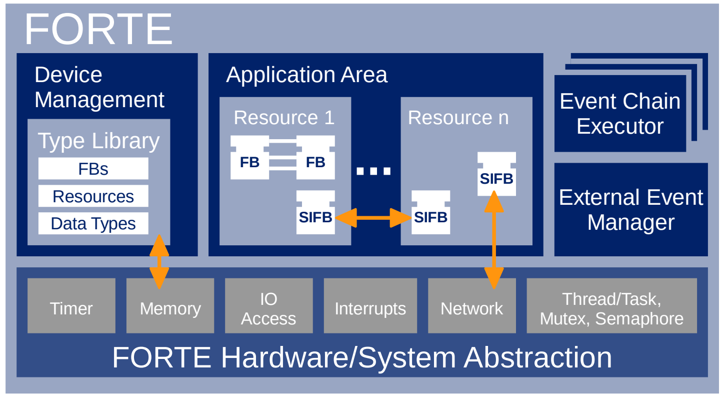
- Open Source Solution for **IEC 61499**
  - Founded 2007
  - Since 2015 Eclipse Project
- Main Components
  - Development Environment: **4diac IDE**
  - Hardware-abstracting execution environment: **4diac FORTE**
- Open Source License
  - Eclipse Public License
  - Allows
    - Usage in products
    - Proprietary extensions


runtime-org.eclipse.fordiac.ide.product - CappingStation/CappingStation.sys - 4diac IDE

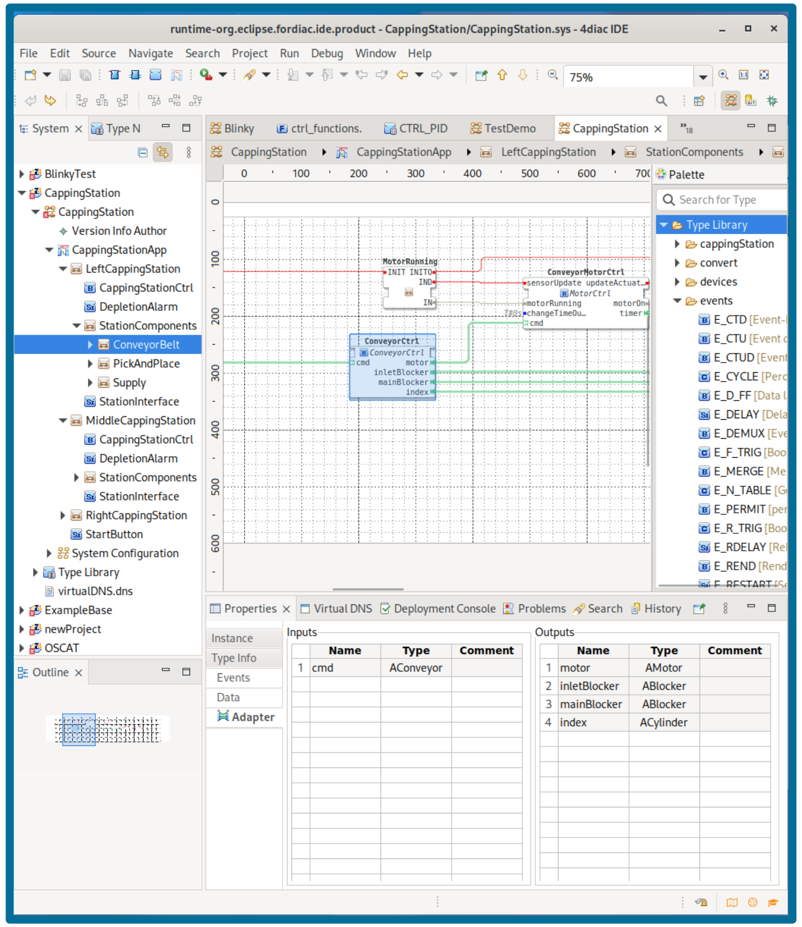
searchString=Motor, in instance name=true, in pin name=true, in type =true, in comment=true: 28 results

Element Kind	Name	Comment	Type	Full Hierarchical Name
AdapterDeclar	motor		AMotor	org.eclipse.fordiac.ide.model.libraryElement.impl.AdapterDeclarationImpl@2a0c76d6 (name: motor, comment:) (isInput: false, typeName: AN
SubApp	Motor		Motor	CappingStationApp.LeftCappingStation.StationComponents.Supply.CapProvider.Motor
AdapterDeclar	motor		AMotor	org.eclipse.fordiac.ide.model.libraryElement.impl.AdapterDeclarationImpl@2a62384 (name: motor, comment:) (isInput: false, typeName: AM
FB	MotorOn		QX	CappingStationApp.LeftCappingStation.StationComponents.ConveyorBelt.MotorOn
SubApp	MotorRunning		untyped	CappingStationApp.LeftCappingStation.StationComponents.ConveyorBelt.MotorRunning
FB	MotorRunning		IX	CappingStationApp.LeftCappingStation.StationComponents.ConveyorBelt.MotorRunning.MotorRunning
FB	ConveyorMotorCtrl		MotorCtrl	CappingStationApp.LeftCappingStation.StationComponents.ConveyorBelt.ConveyorMotorCtrl
VarDeclaratio	motorRunning		BOOL	org.eclipse.fordiac.ide.model.libraryElement.impl.VarDeclarationImpl@208bde64 (name: motorRunning, comment:) (isInput: true, typeName
VarDeclaratio	motorOn		BOOL	org.eclipse.fordiac.ide.model.libraryElement.impl.VarDeclarationImpl@c4f6d5b (name: motorOn, comment:) (isInput: false, typeName: BOOL

372M of 1024M



Overview 4diac IDE

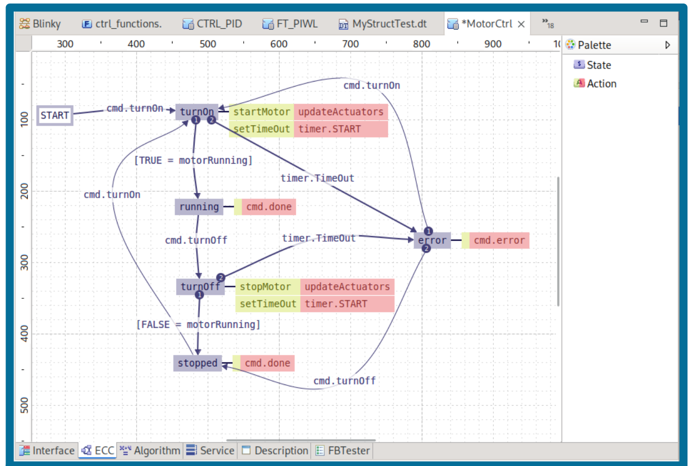


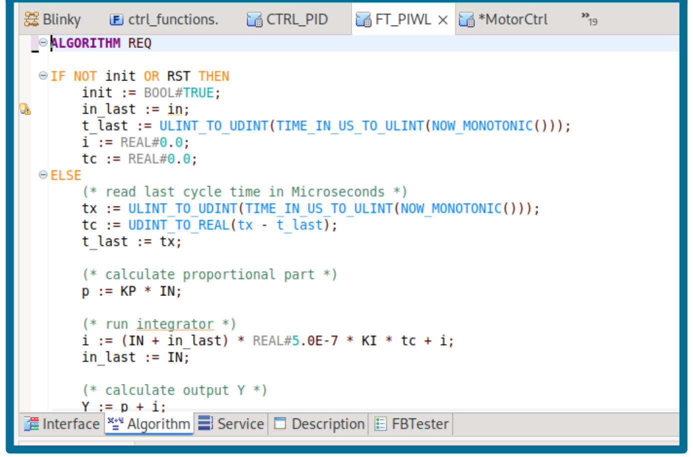
Inputs

	Name	Type	Comment
1	cmd	AConveyor	

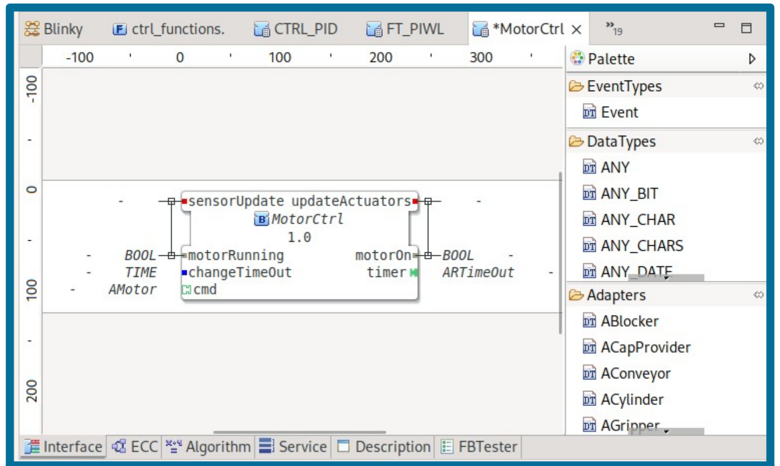
Outputs

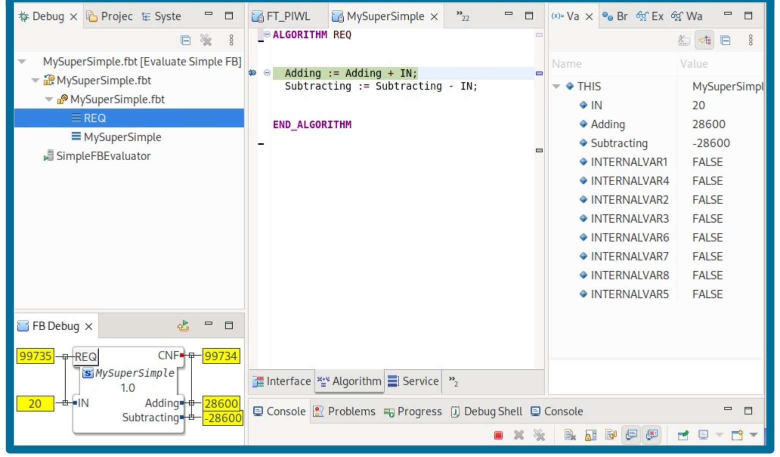
	Name	Type	Comment
1	motor	AMotor	
2	inletBlocker	ABlocker	
3	mainBlocker	ABlocker	
4	index	ACylinder	





```
ALGORITHM REQ

IF NOT init OR RST THEN
    init := BOOL#TRUE;
    in_last := in;
    t_last := ULINT_TO_UDINT(TIME_IN_US_TO_ULINT(NOW_MONOTONIC()));
    i := REAL#0.0;
    tc := REAL#0.0;
ELSE
    (* read last cycle time in Microseconds *)
    tx := ULINT_TO_UDINT(TIME_IN_US_TO_ULINT(NOW_MONOTONIC()));
    tc := UDINT_TO_REAL(tx - t_last);
    t_last := tx;

    (* calculate proportional part *)
    p := KP * IN;

    (* run integrator *)
    i := (IN + in_last) * REAL#5.0E-7 * KI * tc + i;
    in_last := IN;

    (* calculate output Y *)
    Y := p + i;
```



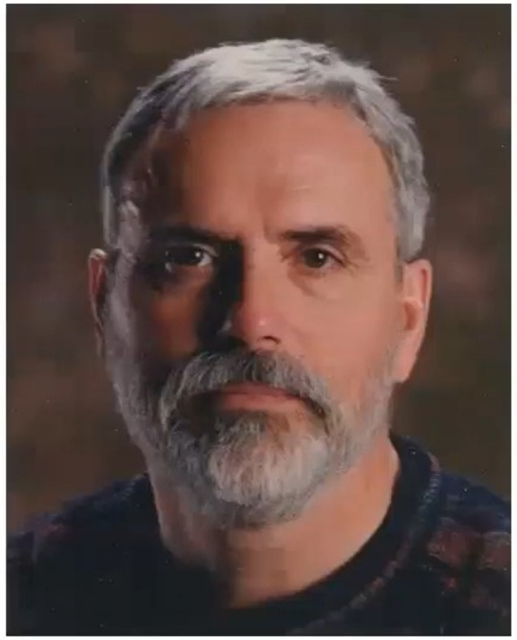


```
ALGORITHM REQ
    Adding := Adding + IN;
    Subtracting := Subtracting - IN;

END_ALGORITHM
```

Name	Value
THIS	MySuperSimpl
IN	20
Adding	28600
Subtracting	-28600
INTERNALVAR1	FALSE
INTERNALVAR4	FALSE
INTERNALVAR2	FALSE
INTERNALVAR3	FALSE
INTERNALVAR6	FALSE
INTERNALVAR7	FALSE
INTERNALVAR8	FALSE
INTERNALVAR5	FALSE

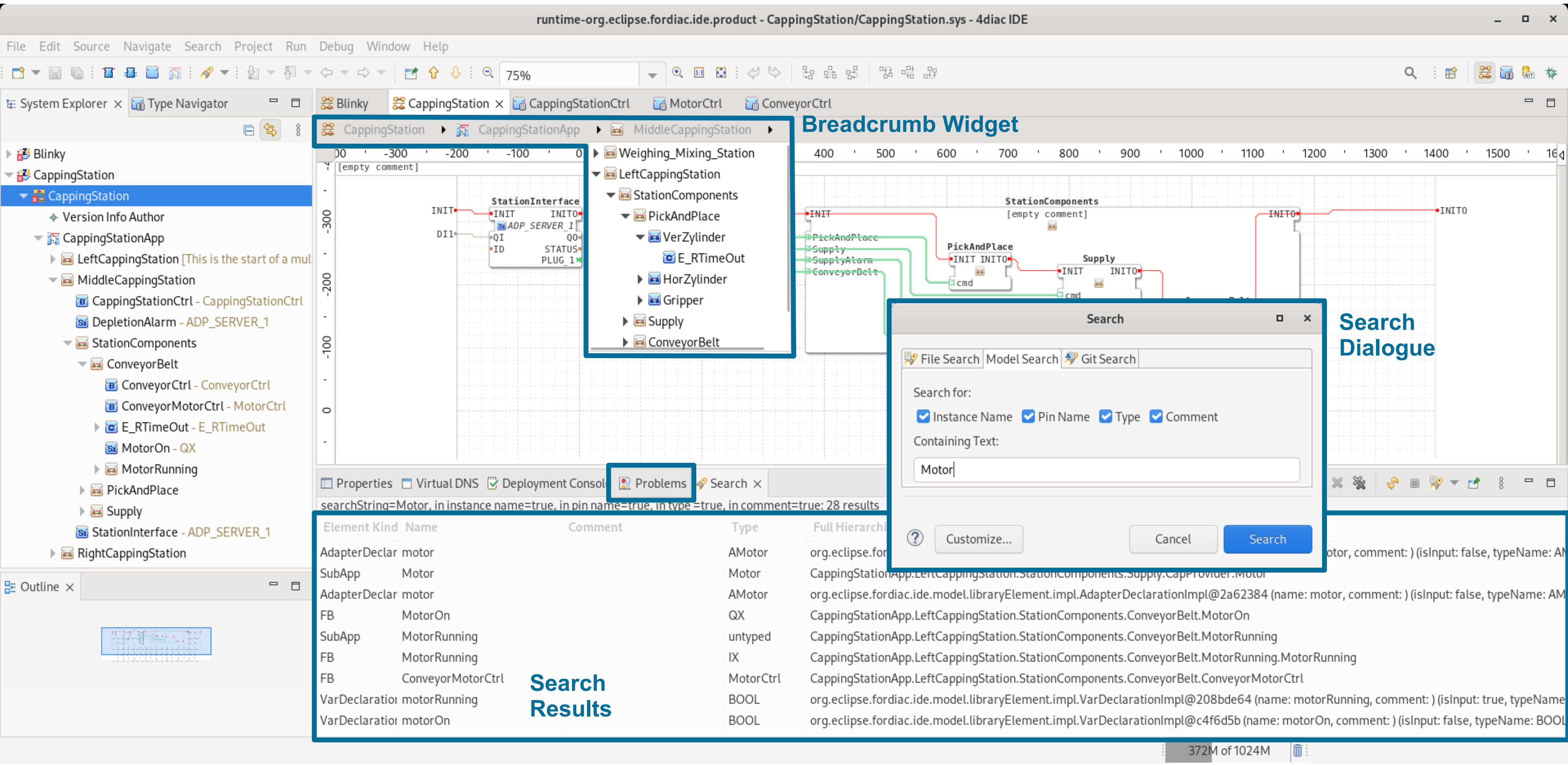
Keynote

What's Wrong with "Users"?



Bran Selic

Malina Software Corp., Canada



System Explorer tree:

- Blinky
- CappingStation
  - CappingStation
    - Version Info Author
    - CappingStationApp
      - LeftCappingStation [This is the start of a mul
      - MiddleCappingStation
        - CappingStationCtrl - CappingStationCtrl
        - DepletionAlarm - ADP_SERVER_1
        - StationComponents
          - ConveyorBelt
            - ConveyorCtrl - ConveyorCtrl
            - ConveyorMotorCtrl - MotorCtrl
            - E_RTimeOut - E_RTimeOut
            - MotorOn - QX
            - MotorRunning
          - PickAndPlace
          - Supply
        - StationInterface - ADP_SERVER_1
      - RightCappingStation

Breadcrumb dropdown:

- Weighing_Mixing_Station
- LeftCappingStation
  - StationComponents
    - PickAndPlace
      - VerZylinder
        - E_RTimeOut
      - HorZylinder
      - Gripper
    - Supply
    - ConveyorBelt

Search dialog:

File Search | Model Search | Git Search

Search for:

[✓] Instance Name [✓] Pin Name [✓] Type [✓] Comment

Containing Text:

Motor

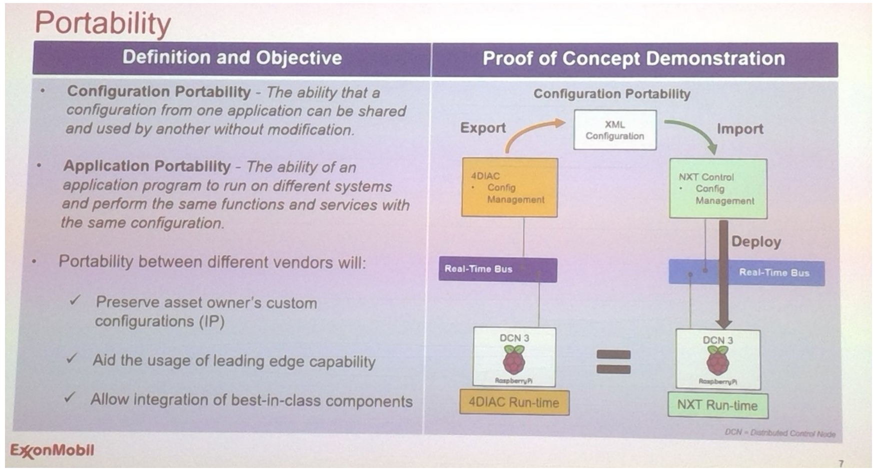
Customize... Cancel Search

searchString=Motor, in instance name=true, in pin name=true, in type =true, in comment=true: 28 results

Element Kind	Name	Comment	Type	Full Hierarchi
AdapterDeclar	motor		AMotor	org.eclipse.for… otor, comment:) (isInput: false, typeName: AM
SubApp	Motor		Motor	CappingStationApp.LeftCappingStation.StationComponents.Supply.CapProvider.Motor
AdapterDeclar	motor		AMotor	org.eclipse.fordiac.ide.model.libraryElement.impl.AdapterDeclarationImpl@2a62384 (name: motor, comment:) (isInput: false, typeName: AM
FB	MotorOn		QX	CappingStationApp.LeftCappingStation.StationComponents.ConveyorBelt.MotorOn
SubApp	MotorRunning		untyped	CappingStationApp.LeftCappingStation.StationComponents.ConveyorBelt.MotorRunning
FB	MotorRunning		IX	CappingStationApp.LeftCappingStation.StationComponents.ConveyorBelt.MotorRunning.MotorRunning
FB	ConveyorMotorCtrl		MotorCtrl	CappingStationApp.LeftCappingStation.StationComponents.ConveyorBelt.ConveyorMotorCtrl
VarDeclaratio	motorRunning		BOOL	org.eclipse.fordiac.ide.model.libraryElement.impl.VarDeclarationImpl@208bde64 (name: motorRunning, comment:) (isInput: true, typeName
VarDeclaratio	motorOn		BOOL	org.eclipse.fordiac.ide.model.libraryElement.impl.VarDeclarationImpl@c4f6d5b (name: motorOn, comment:) (isInput: false, typeName: BOOL

Users

- Industry
  - ABB
  - AIMIRIM
  - Awite
  - embedded ocean GmbH
  - ExxonMobil
  - HIT robot group
  - HR Agrartechnik
  - meo Energy
  - NOJA Power
  - Schneid GmbH
  - …
- Universities & Research Institutions in Austria, Australia, Brasilia, Finland, Germany, Italy, Japan, Spain, Sweden, ...


Portability

Definition and Objective

- **Configuration Portability** - *The ability that a configuration from one application can be shared and used by another without modification.*
- **Application Portability** - *The ability of an application program to run on different systems and perform the same functions and services with the same configuration.*
- Portability between different vendors will:
  - ✓ Preserve asset owner's custom configurations (IP)
  - ✓ Aid the usage of leading edge capability
  - ✓ Allow integration of best-in-class components

Proof of Concept Demonstration

ExxonMobil 7

4diac FORTE with ISOBUS
IEC 61499 combined with ISO 11783

Architecture

- ECU with a dual core Xtensa LX6, 16MB Flash, 8MB RAM
- *FreeRTOS*
- *CCI ISOBUS driver*
- *2 CAN Bus (1x ISOUBS, 1x internal)*
- *DataPanel i/o on internal*
- *Harsh environment*

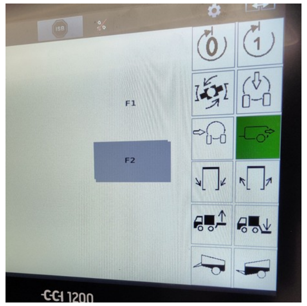
Communication

- ISO11783
- Masks Designed with Jetter ISO-Designer
- Support AUX-O and AUX-N
- *Soon more (e.g., Task-Controller)*

Usage

- low-code platform for ISOBUS
  - targeted to batch size 1 agricultural machines

AUX-N



ISOBUS-Display

The Worlds first Open Source Slurry Tanker.
The Photo shows the real machine, no Marketing Photo

https://github.com/Meisterschulen-am-Ostbahnhof-Munchen/4diac-SlurryTanker-sample
https://hr-agrartechnik.de/

Thank you!

LIT | Cyber-Physical Systems Lab
Johannes Kepler University Linz

IEC 61499, a system so fine,
Controls processes, bringing efficiency in line.
A network built of functions divine,
It offers us control, we need it all the time.

Networks of functions, a beautiful view,
Flowing together, with seamless control too.
They interact, to bring forth what is true,
And with IEC 61499, everything comes through.

In factories and plants, near and far,
Its presence is felt, a shining star.
Safe and reliable, never any lag,
It's the backbone of our era, a timeless piece of art.

Real-time control, is what we desire,
With IEC 61499, our dreams won't tire.
Functions always up-to-date, forever anew,
In a world that's constantly changing fast, that's true.

So let us celebrate, IEC 61499 so pure,
Our system that helps us, the future secure.
With efficiency, control and bravery in tow,
Together, let us reach our goal, in real-time, and grow.

ChatGPT in the style of Goethe, instructed by Bianca Wiesmayr, 2023