



Integrating Capella, SCADE and medini analyze, for MBSE, Embedded SW Development and Safety Analysis

Thierry Le Sergent
SCADE Architect Product Manager

Agenda

- Introduction
- Models Synchronization
- Domain Specific Languages
- Capella – SCADE & medini toolchain

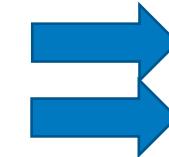


Model-Based Systems Engineering (MBSE)

MBSE Workflows, Methods, Tools

- Objective
 - Design the right Product/System from user needs
- Means
 - Successive levels of abstraction to refine system requirements from user needs to real product

- Difficulties
 - Management of complexity and ambiguity at each level
 - Management of consistency between the levels of concerns
 - Multiple languages for engineering disciplines & domains



Models Synchronization
Domain Specific Language

- Models
 - Help managing the complexity and ambiguity
 - Not a single tool for everything !

➔ Managing the refinement process is key



Models Synchronization

SCADE Tools



Models consistency between models : 2 ways

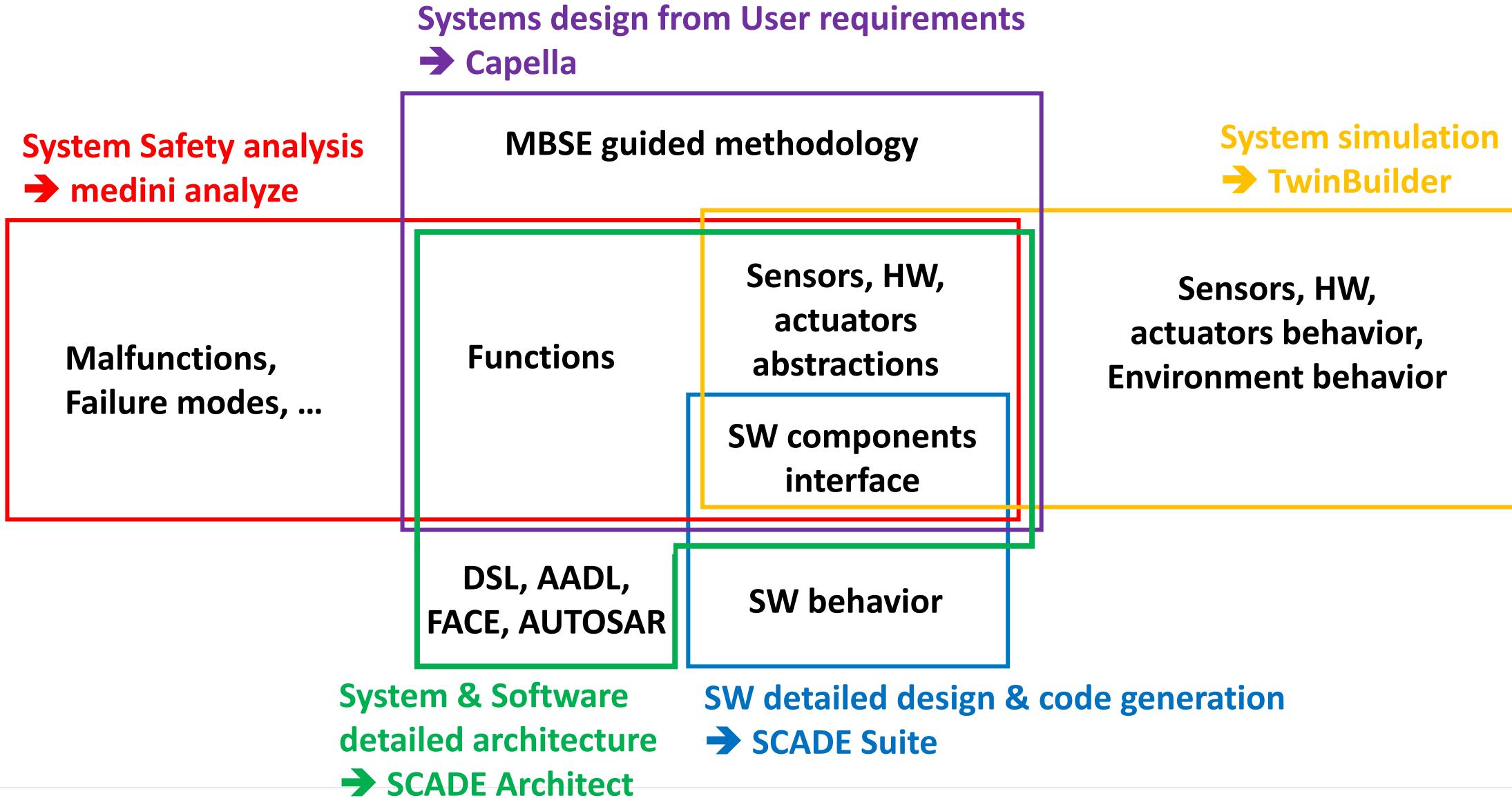
- “Traceability” or “allocation”

-  Verification of “completion” can be automated (checker)
-  Manual operation: creation of objects at both levels and manual links

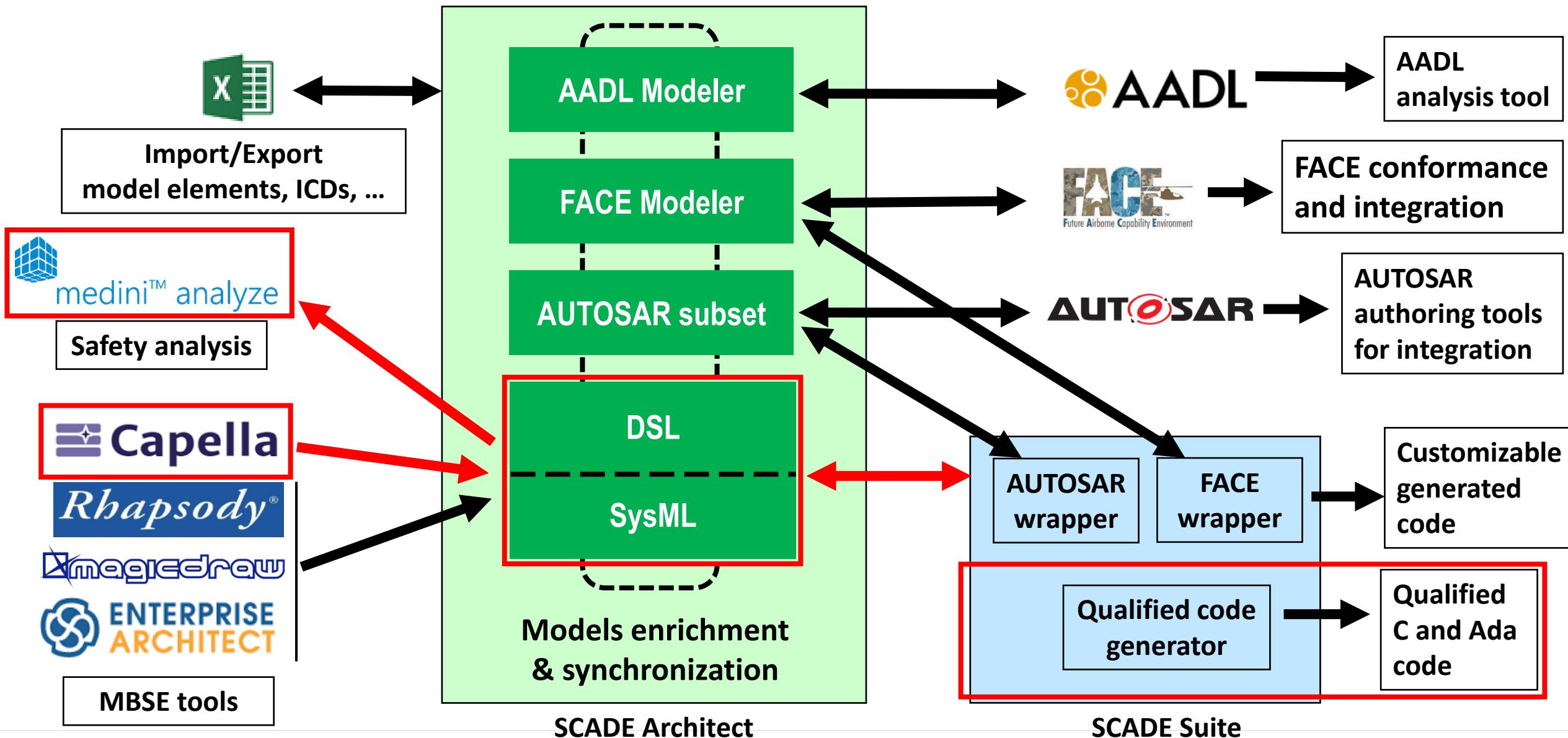
- Models “Synchronization”

- = Model transformation + model diff-merge (allows incremental edition on both sides)
 - Partial model transformation : each side has its own added value
 - Only a subset of the models represent the same information.
-  Automated transformation → Consistent by construction

Models consistency between models



SCADE capabilities for MBSE workflows



Model-based Functional Safety Analysis and Design with medini



SCADE Architect

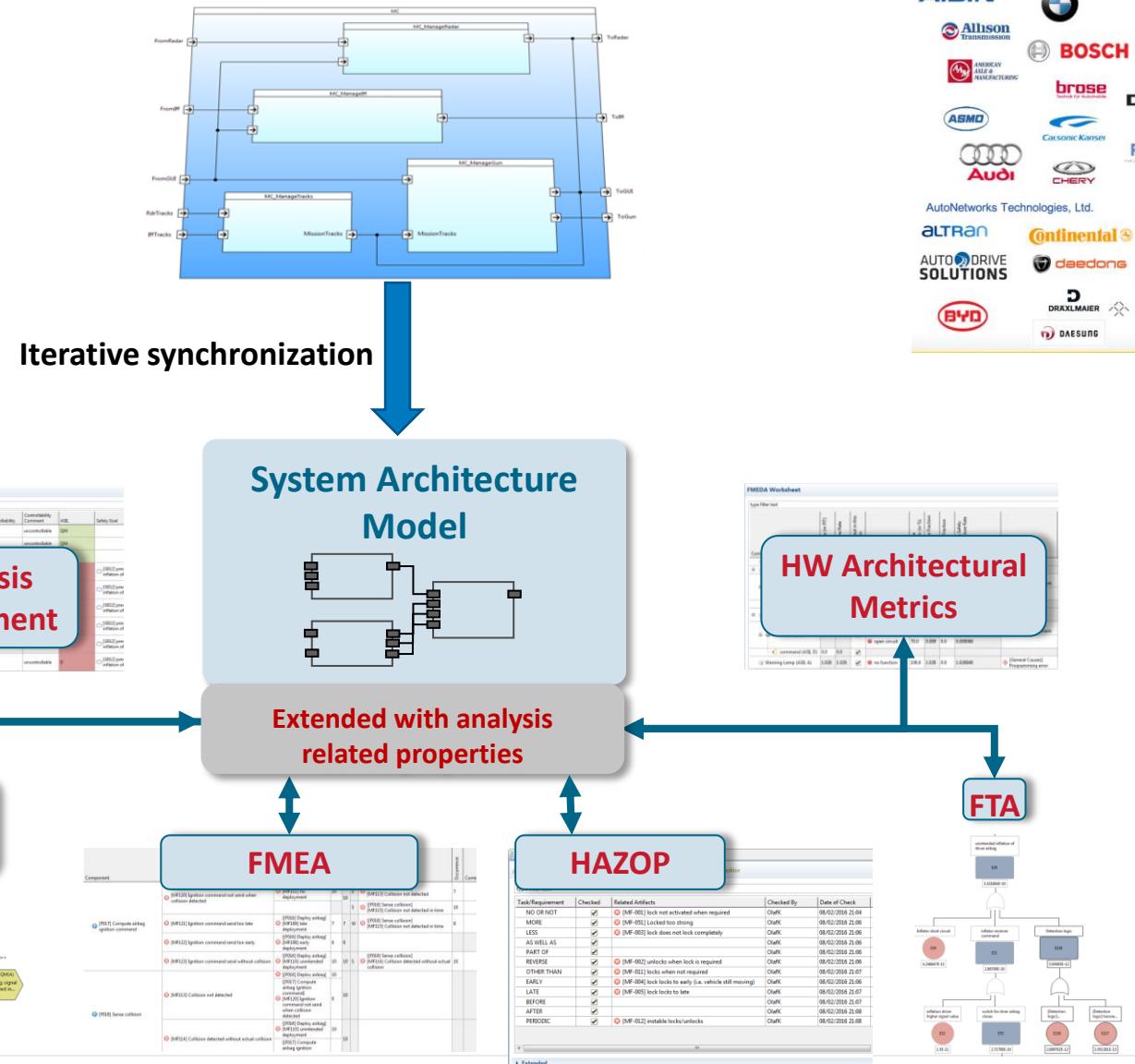
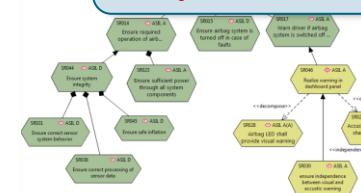
System/SW Architecture



Functional Safety
Analysis and
Design



Safety Requirements



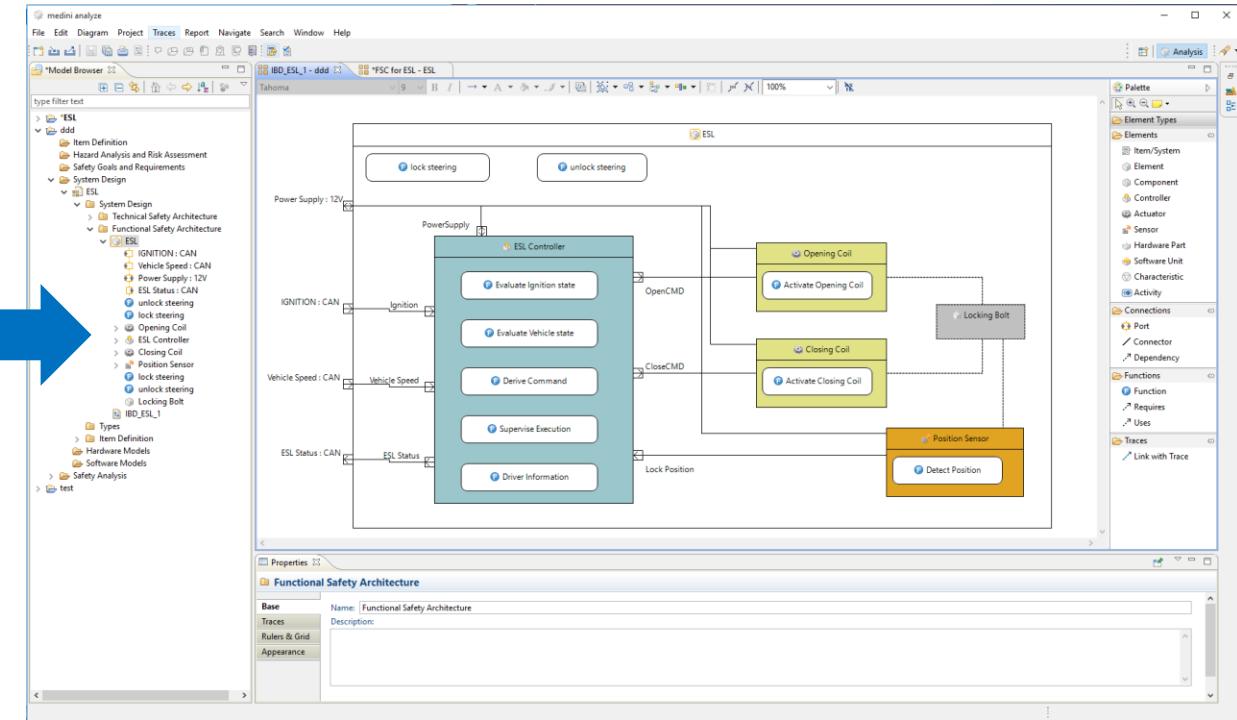
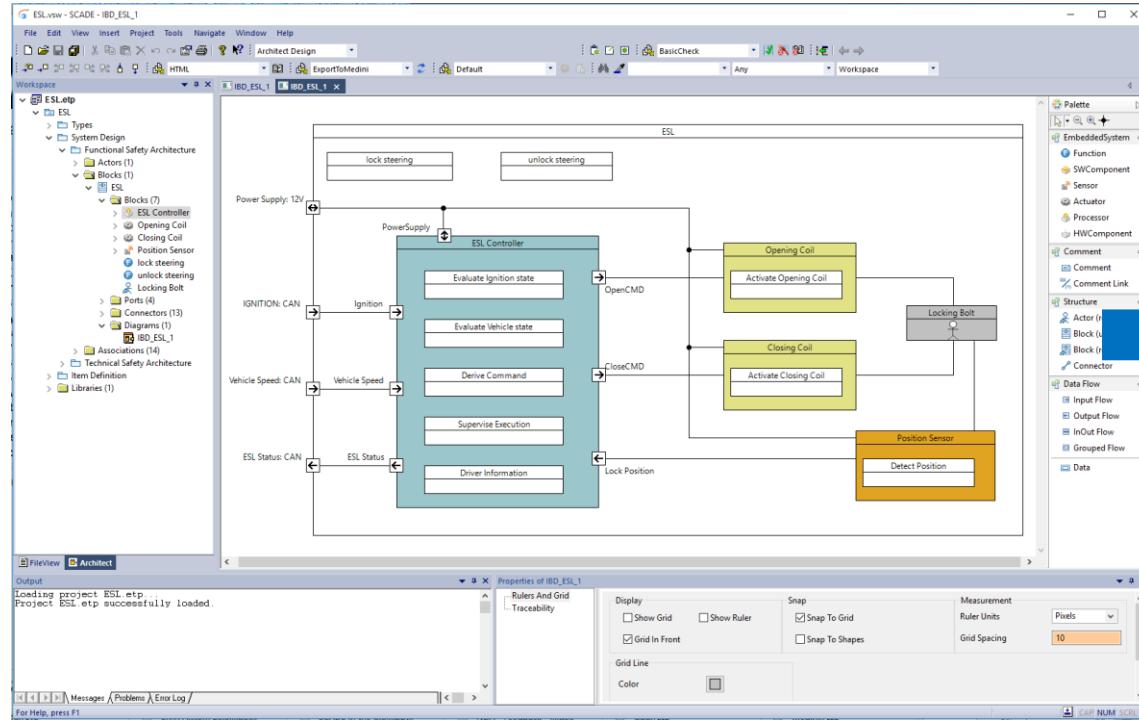
Safety process seamlessly integrated with system development

Safety analysis results always consistent

Safety requirements discovered and considered early in the design process

Model-based Functional Safety Analysis and Design with medini

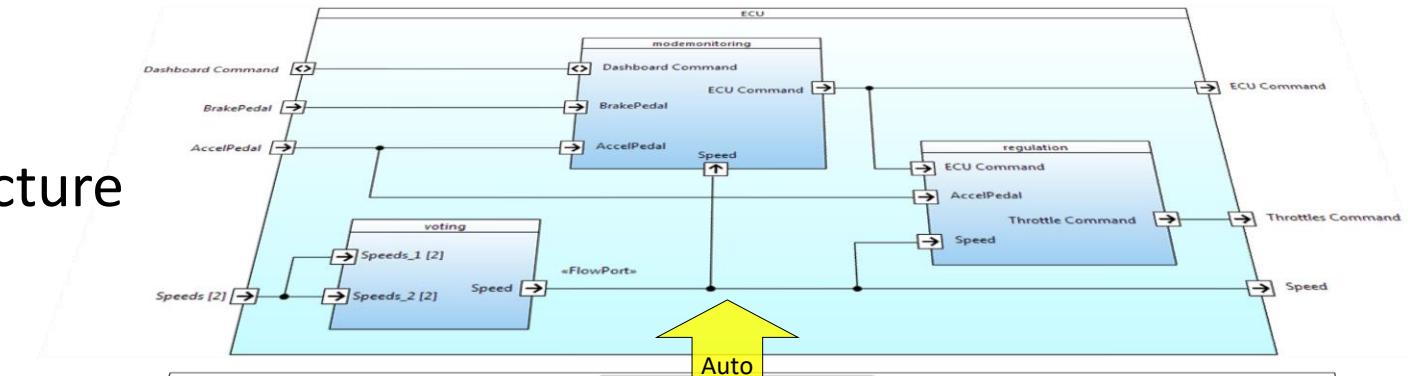
SCADE Architect



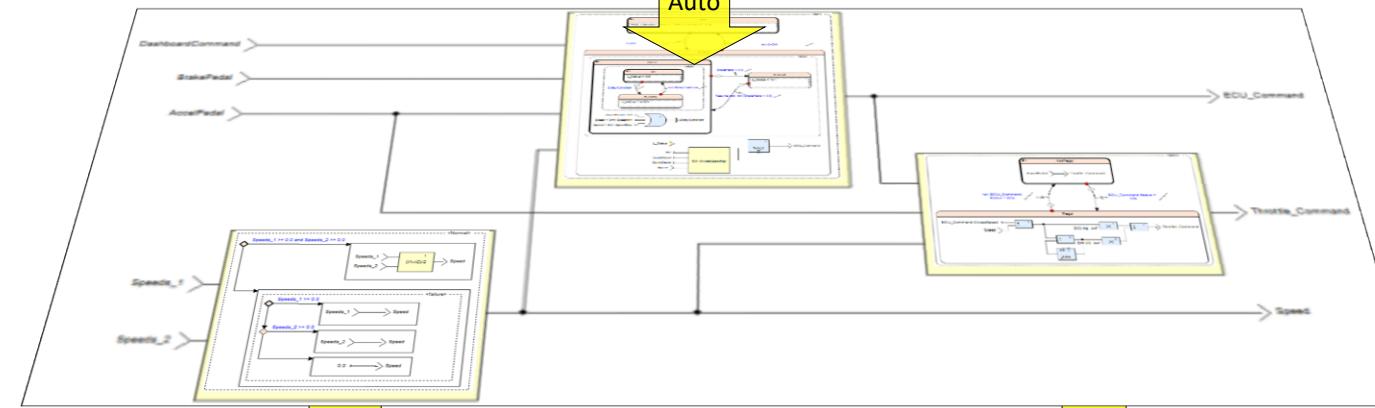
Synchronization of blocks diagram

Model-based Embedded Safety Critical SW with SCADE Suite

SW Architecture



SW Design



SW Coding

```
/* Architecture::Regulation
void Regulation_Architecture(
    tECU_cmd_Architecture *tECU_Command,
    /* AccelPedal */
    tPercent_Architecture AccelPedal,
    /* Speed */
    tVehicleSpeed_Architecture Speed,
    outC_Regulation_Architecture *outC)
{
    kcg_float32 tmp;
    /* SMI_Regul_L3 */ 
    kcg_float32 _L3_Regul_SMI;
    /* SMI */
    SSM_ST_SMI SMI_state_act;
    /* SMI */
    kcg_bool SMI_reset_act;
    /* SMI */
    switch(outC->SMI_state_nxt) {
        case SSM_st_NotRegul_SMI :
            SMI_reset_act = (*ECU_Command).Status == ON_Architecture;
            if(SMI_reset_act) {
                SMI_state_act = SSM_st_Regul_SMI;
            }
            else {
                SMI_state_act = SSM_st_NotRegul_SMI;
            }
            break;
    }
}
```

C

```
-- Architecture::Regulation
procedure Regulation(
    tECU_Command : in tECU_Command;
    -- AccelPedal/
    AccelPedal : in tPercent;
    -- Speed/
    Speed : in tVehicleSpeed;
    Ctx : in out Context_Regulation)
is
    -- SMI:
    SMI_state_act : Kcg_Types.SSM_ST_SMI;
    -- SMI:
    SMI_reset_act : Boolean;
    SMI_Regul_L3 /
    L3 : Kcg_Config.Kcg_Float32;
    tmp : Kcg_Config.Kcg_Float32;
begin
    case (Ctx.SMI_state_nxt) is
        when Kcg_Types.SSM_st_NotRegul =>
            SMI_reset_act := ECU_Command.Status = Kcg_Types.ON;
        if(SMI_reset_act) then
            SMI_state_act := Kcg_Types.SSM_st_Regul;
        else
            SMI_state_act := Kcg_Types.SSM_st_NotRegul;
        end if;
    end case;

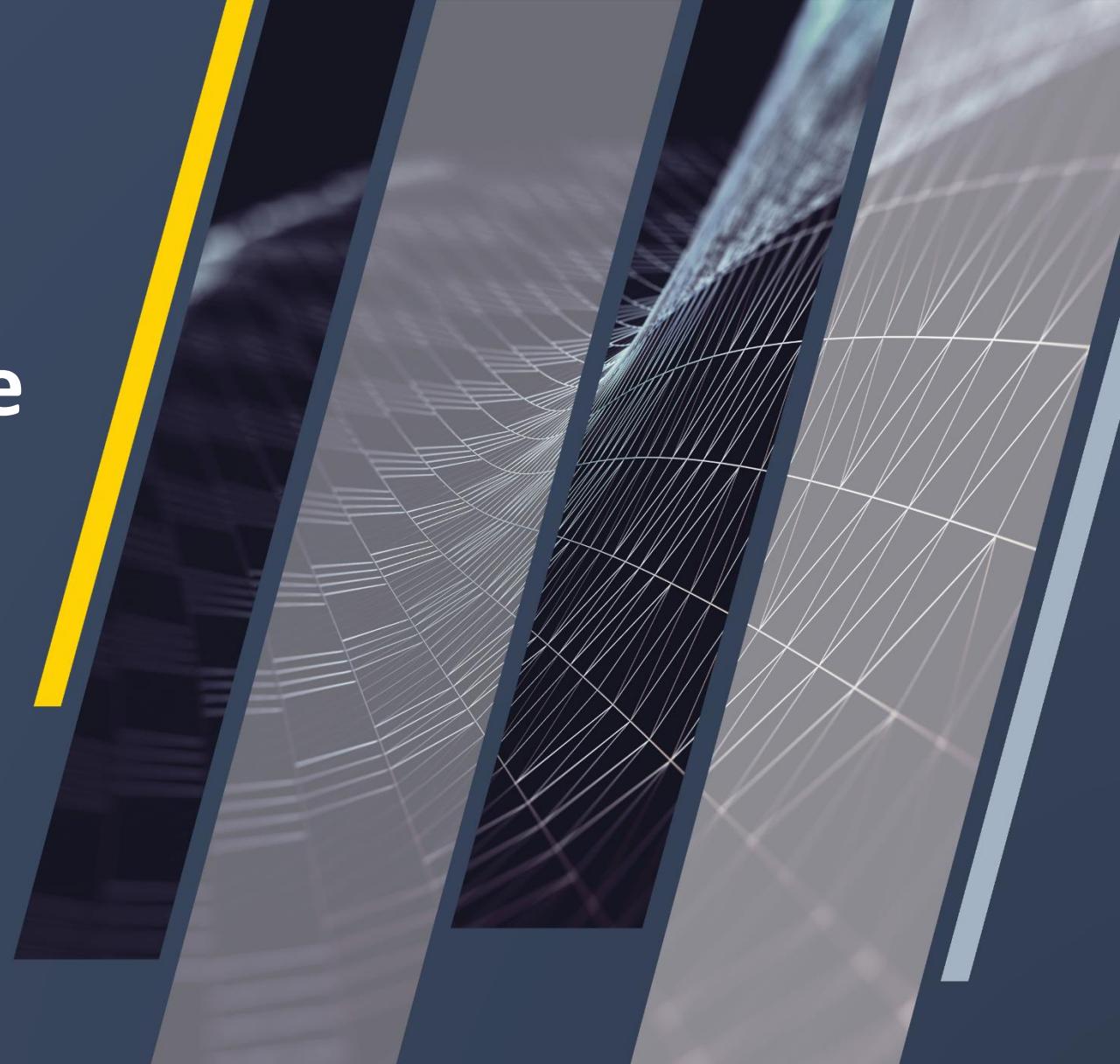
```

Ada



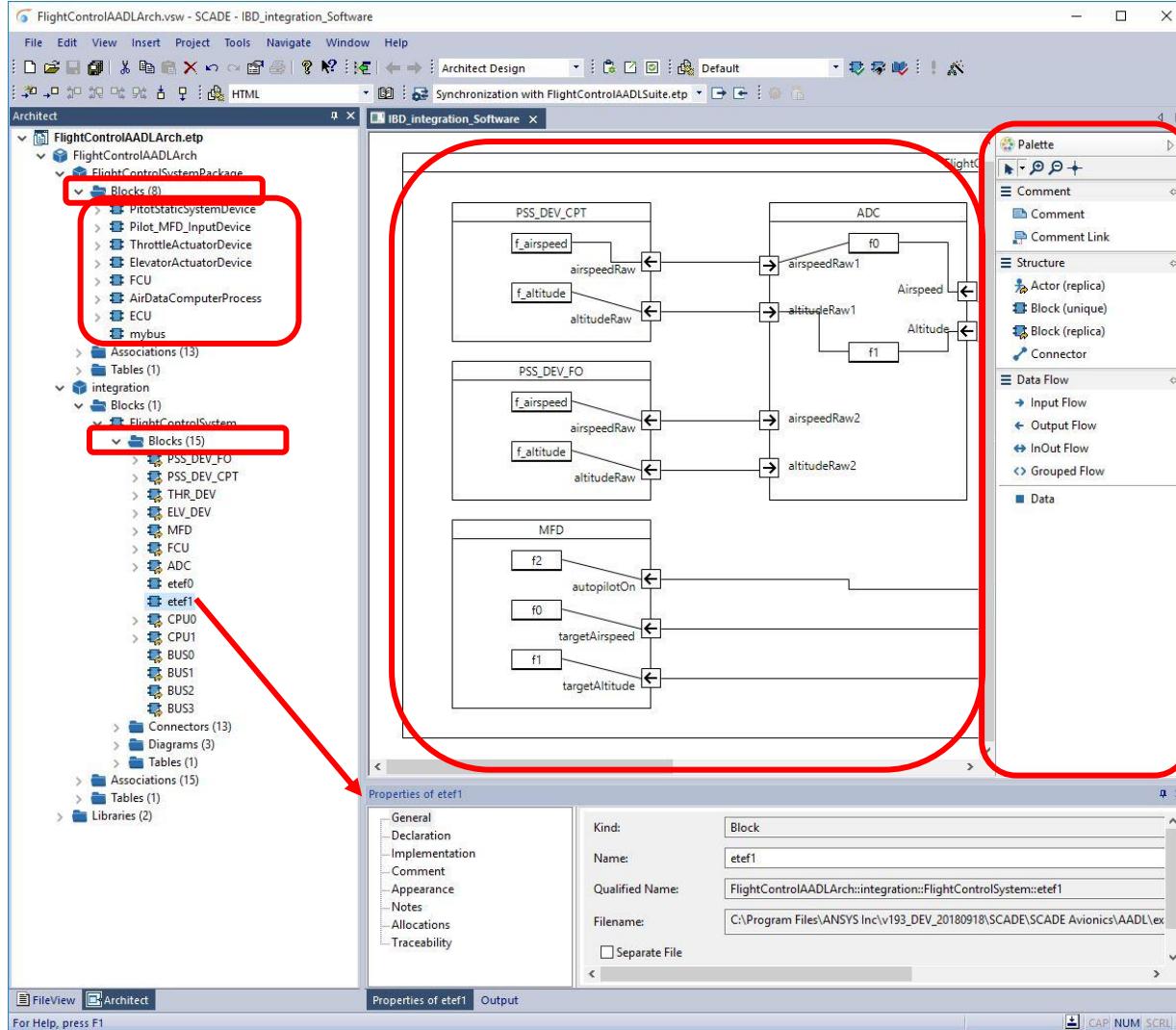


Domain Specific Language SCADE Architect Configurations

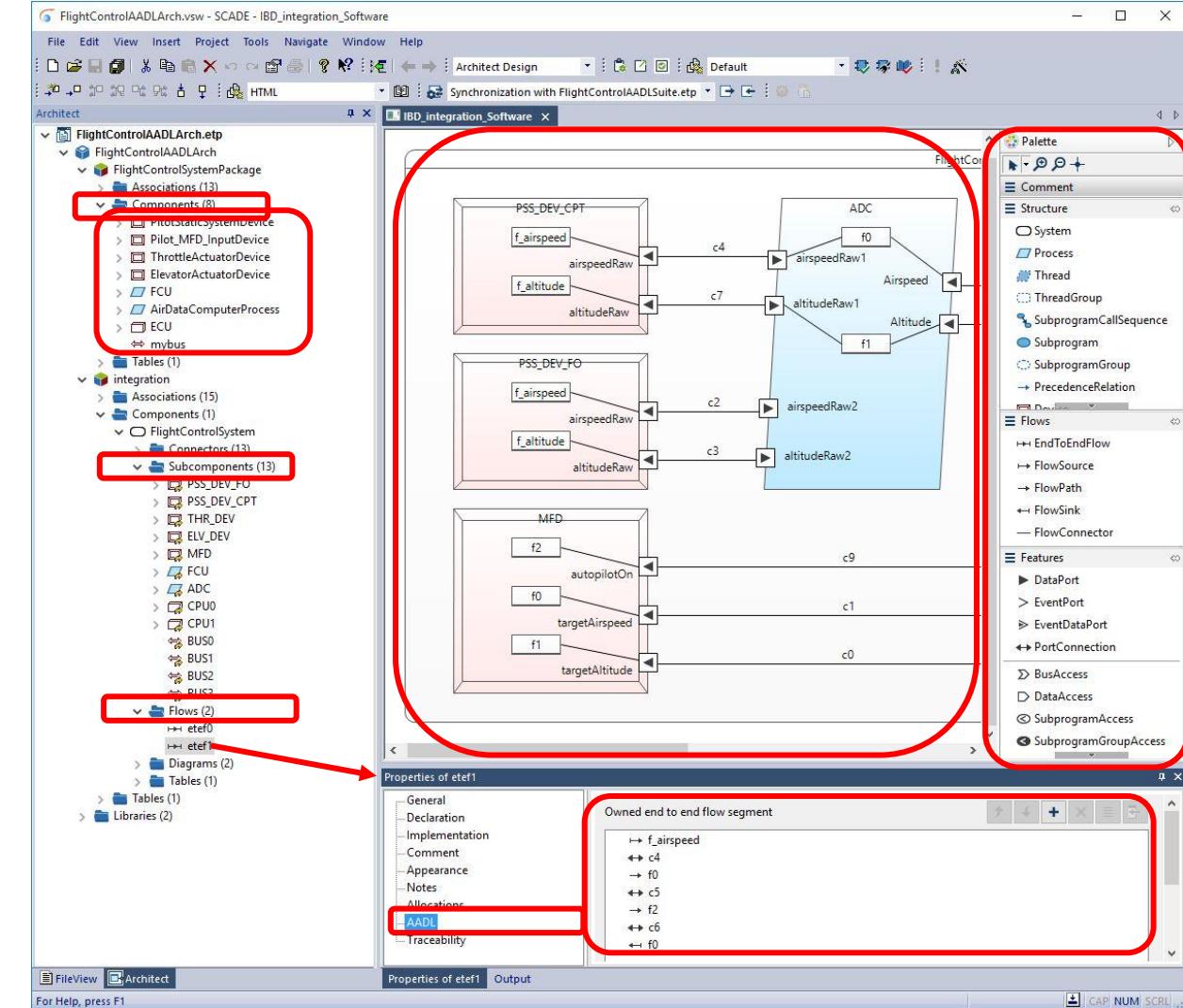


SysML vs. DSL : tool perspectives

Default IDE

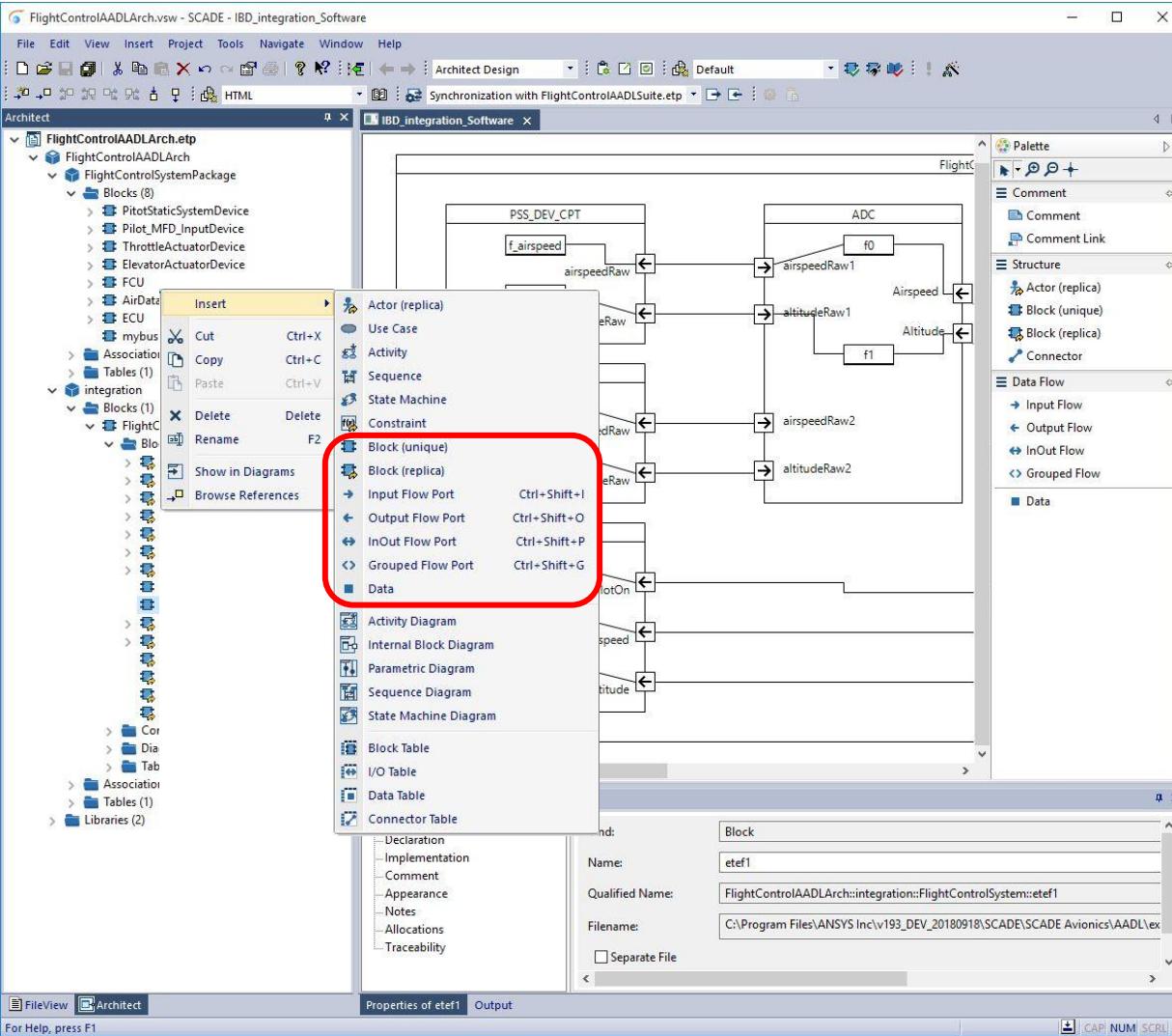


Customized IDE



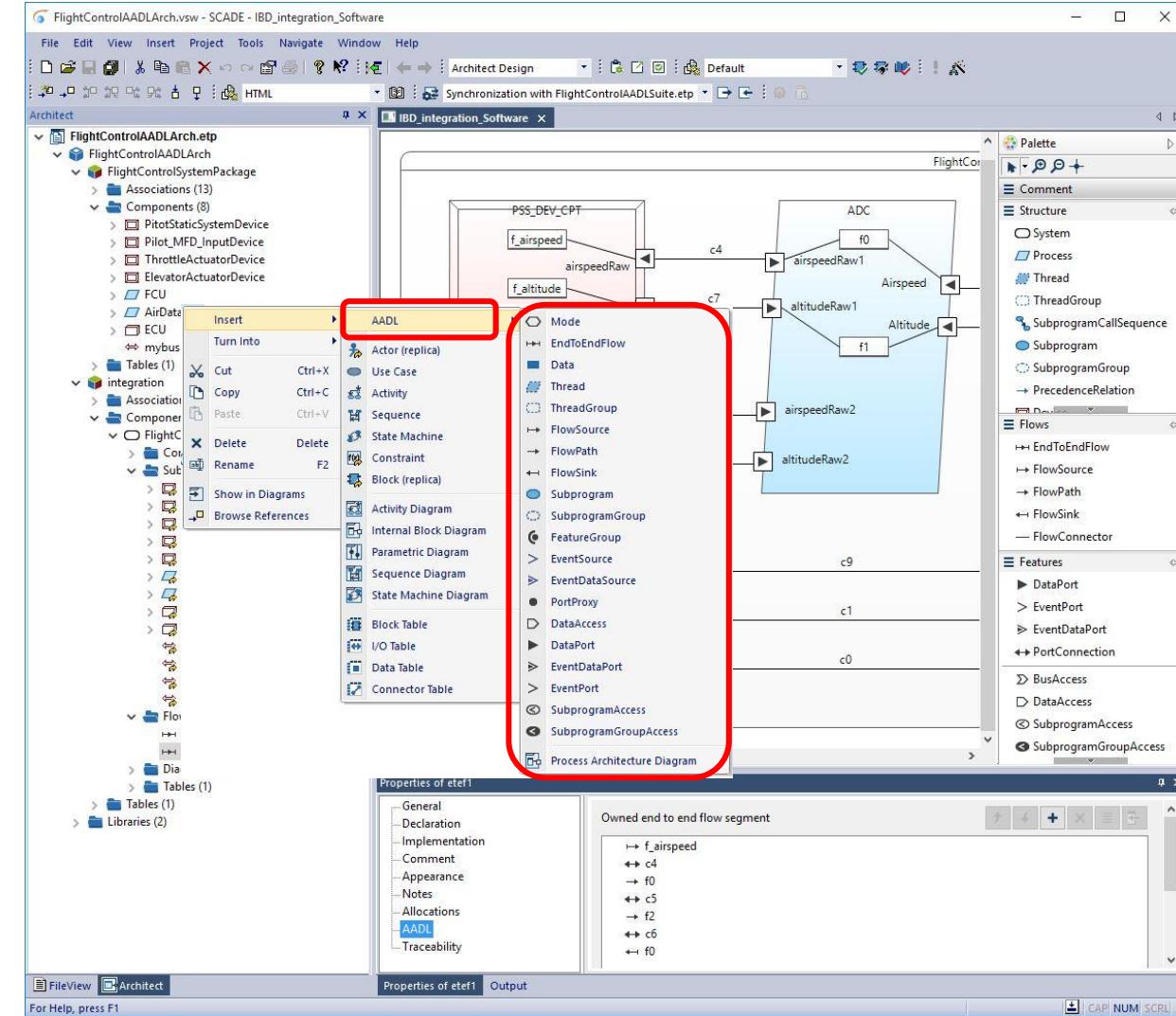
SysML vs. DSL : tool perspectives

Default IDE



Dedicated insert menu
Depending on object context

Customized IDE

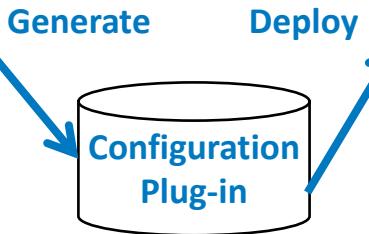
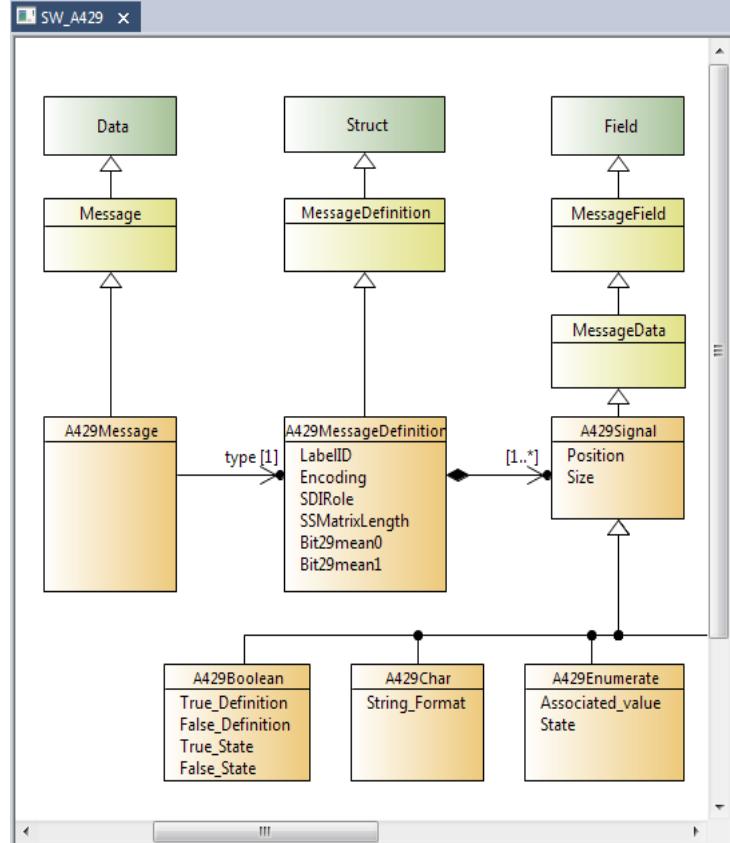


SCADE Architect Configurator

Workflow

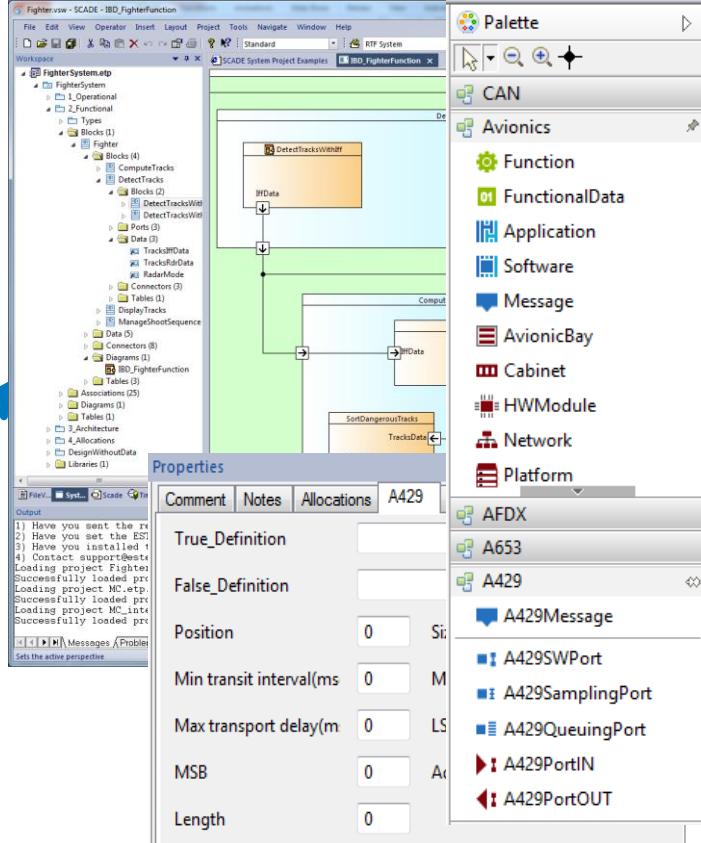
Specialist

SCADE Architect Configurator



End-User

SCADE Architect
Modeler



Define customized object kinds,
derived from SCADE Architect objects

Domain specific modeler



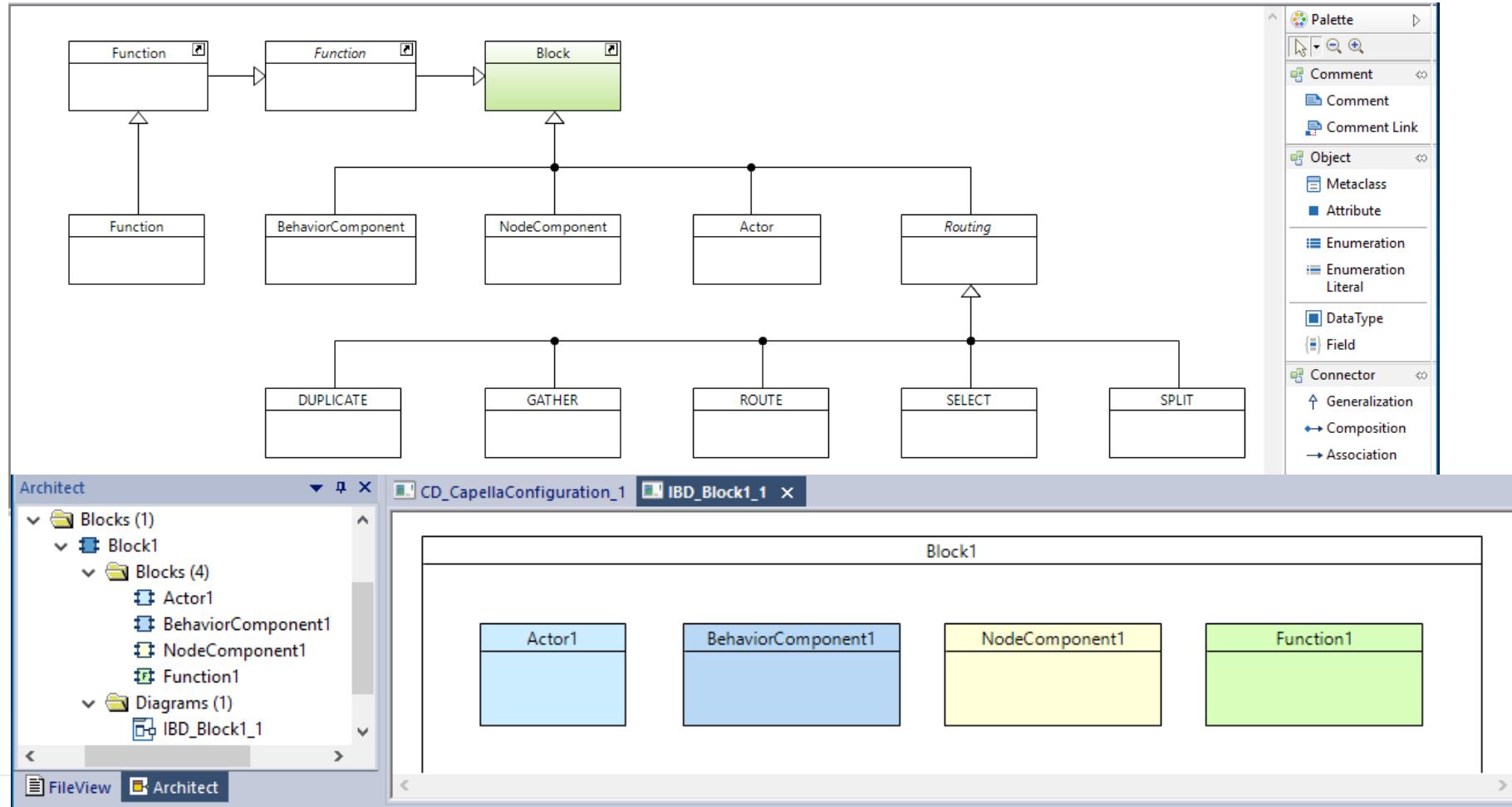
Capella – SCADE & medini Toolchain



SCADE Architect Configuration for Capella

Definition of

- Object kinds Matching Capella
- Graphical styles & icons



SCADE Architect
for Capella



Models Synchronization

Capability and limitation

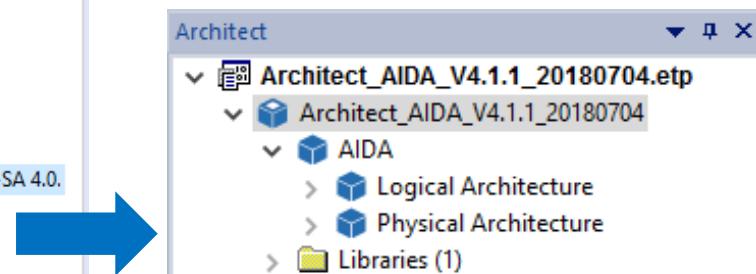
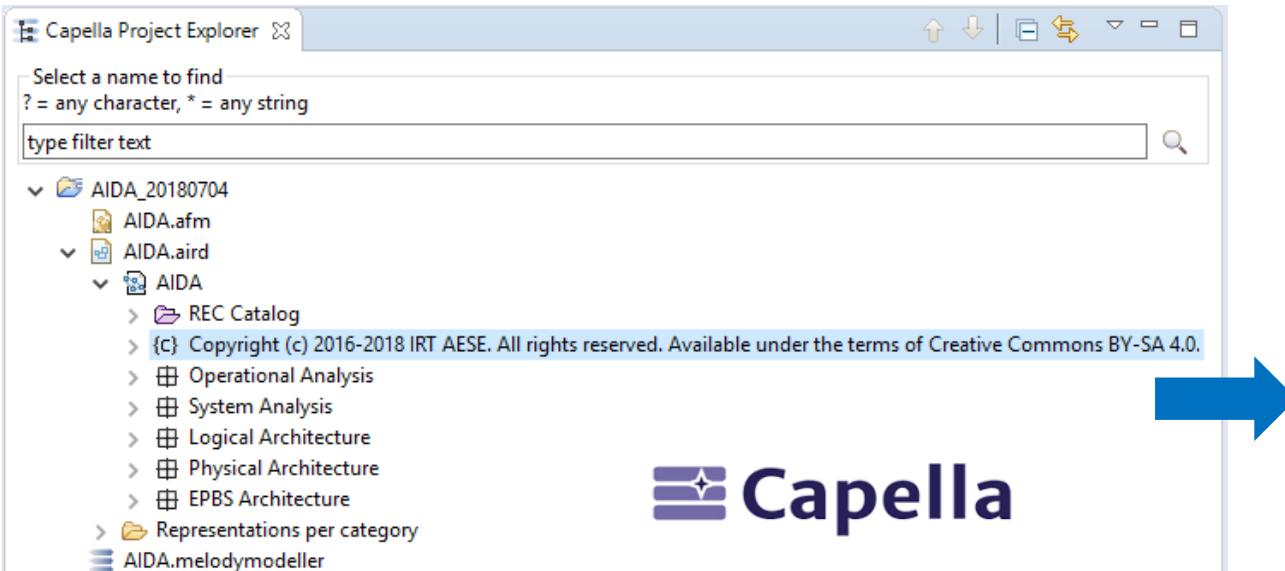
- Models synchronization  **Capella** → **SCADE Architect**

→ Import structural elements (packages, components, connections, allocations, types) and graphical diagrams

- Graphical diagrams
 - Capella can represent components from other context, and their connections
 - SCADE Architect can represent these thanks to “references”, but not the “derived” connection
 - Medini analyze and SCADE Suite can represent only components of the block diagram container
- Diagrams are fully imported in SCADE Architect / medini analyze / SCADE Suite when drawn with these constraints, otherwise partially imported.

Demonstration model

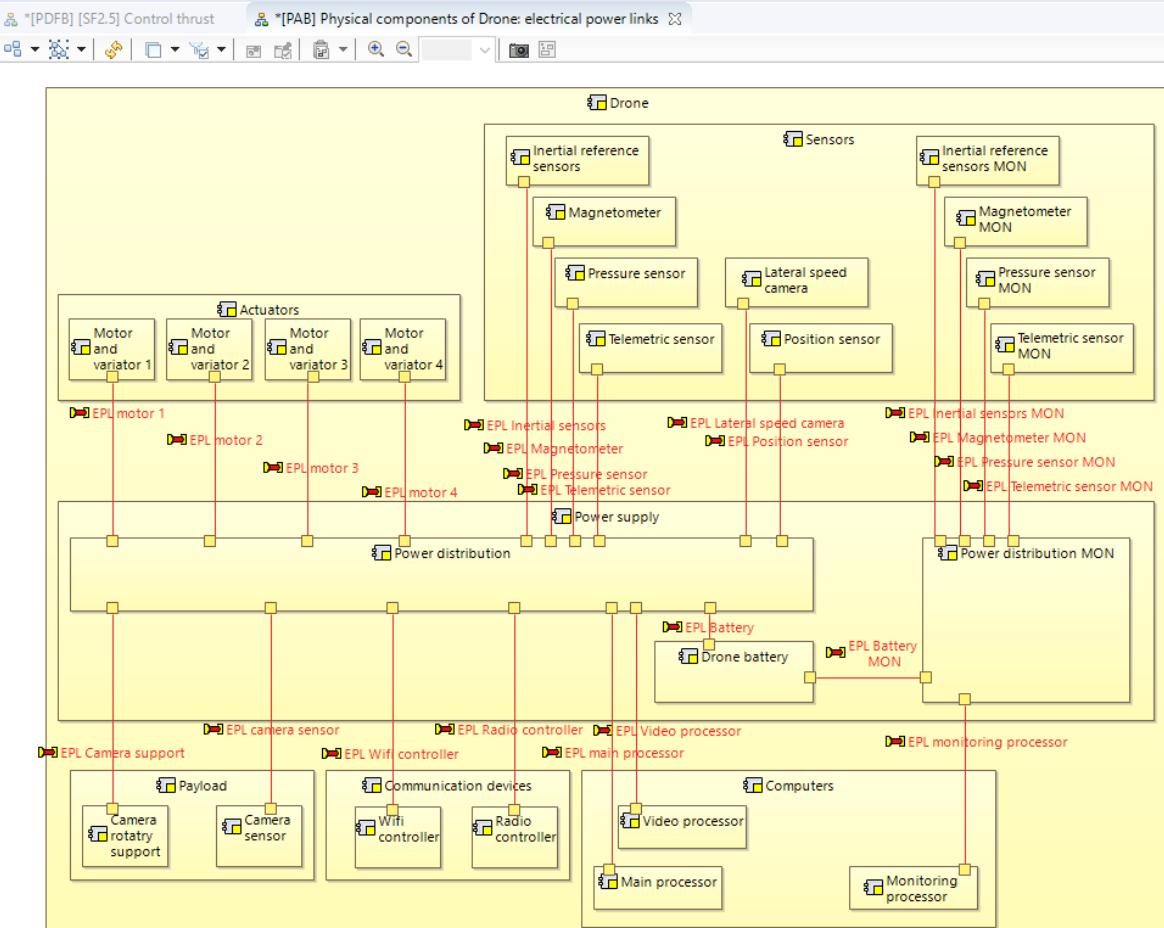
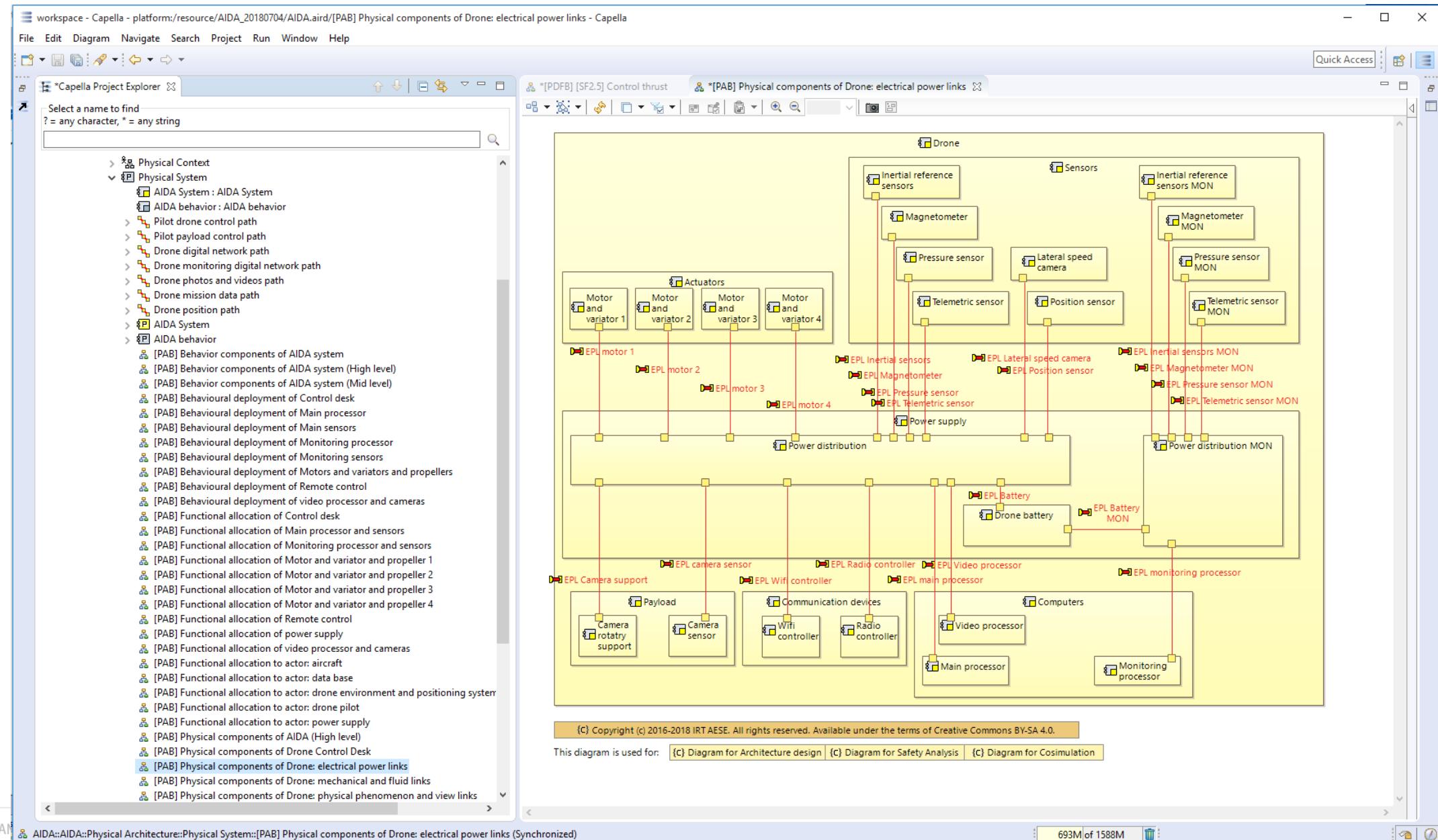
- Inspection drone
 - Based on AIDA models developed by IRT Saint-Exupéry
 - IRT forge : [https://sahara\(pf.irt-saintexupery.com](https://sahara(pf.irt-saintexupery.com)



SCADE Architect



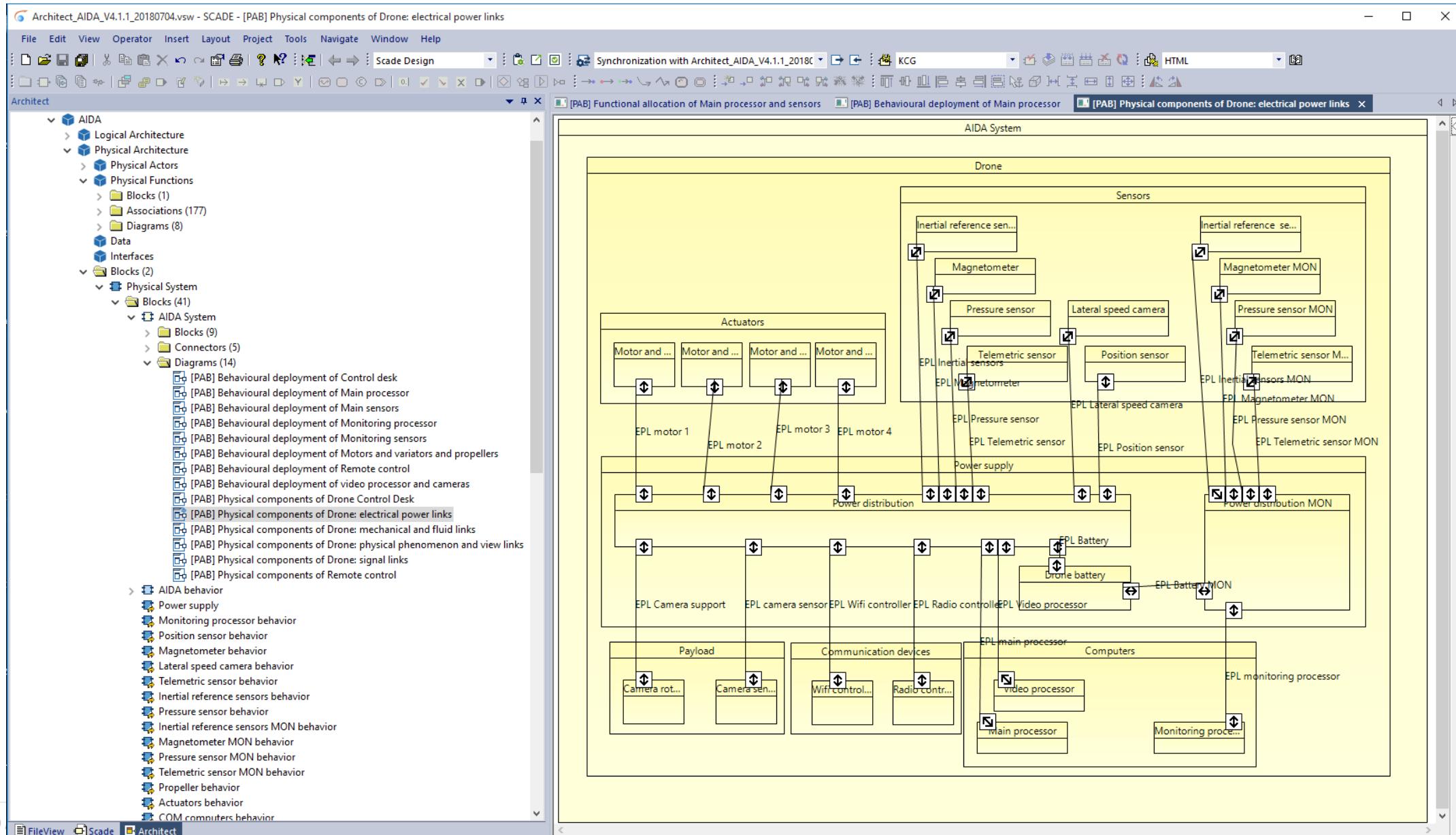
Capella (Physical component of Drone: electrical power links)



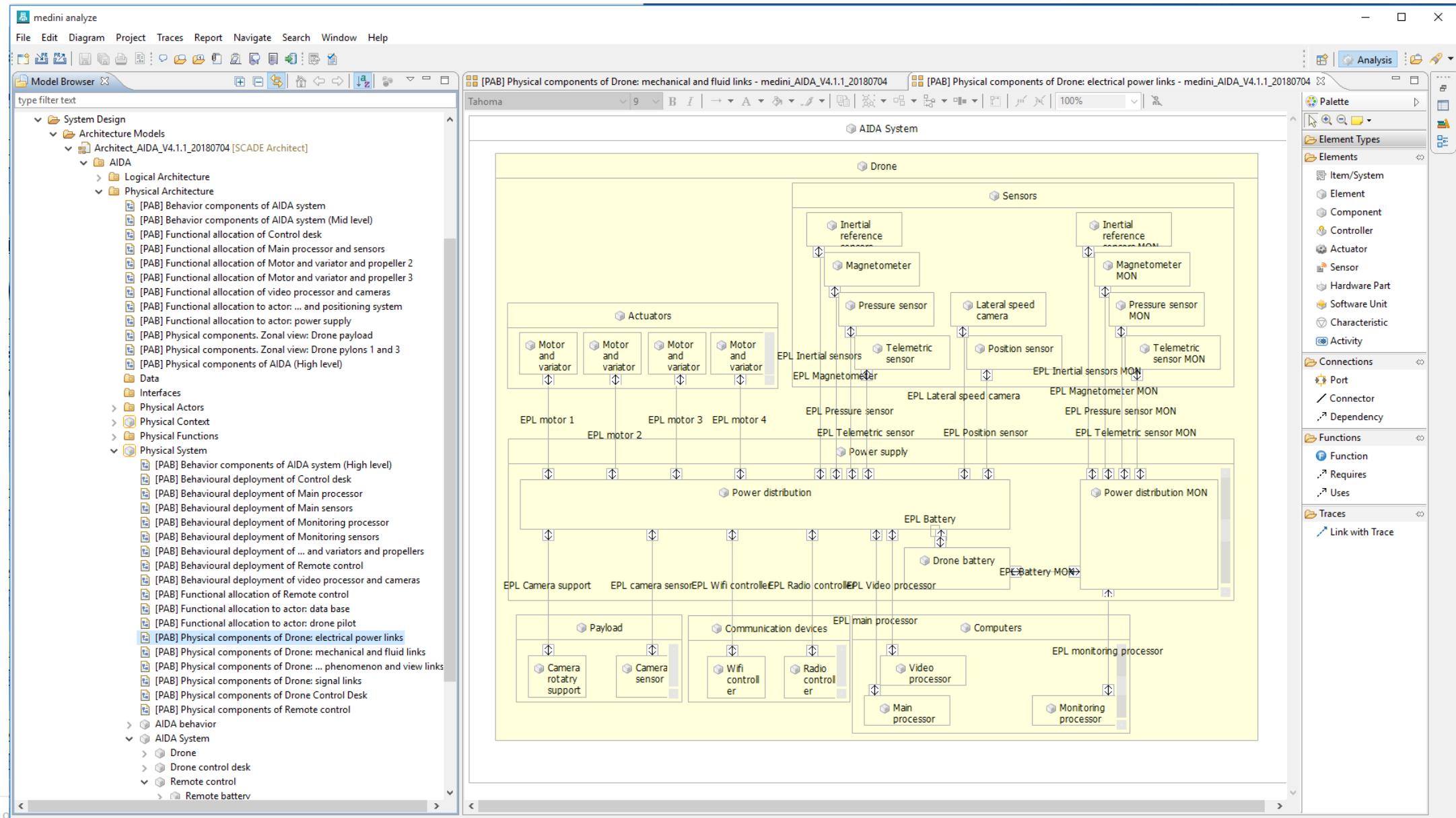
(C) Copyright (c) 2016-2018 IRT AESE. All rights reserved. Available under the terms of Creative Commons BY-SA 4.0.

This diagram is used for: (C) Diagram for Architecture design | (C) Diagram for Safety Analysis | (C) Diagram for Cosimulation

SCADE Architect (Physical component of Drone: electrical power links)

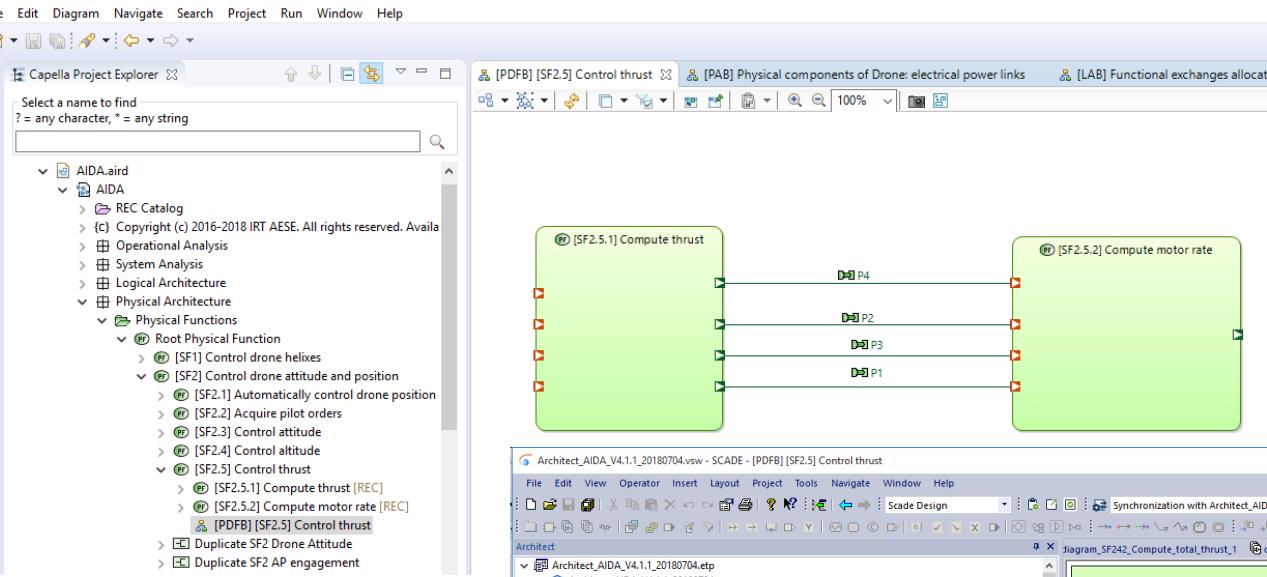


Medini analyze (Physical component of Drone: electrical power links)

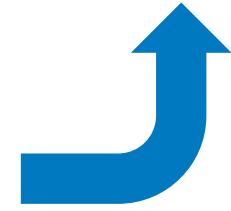
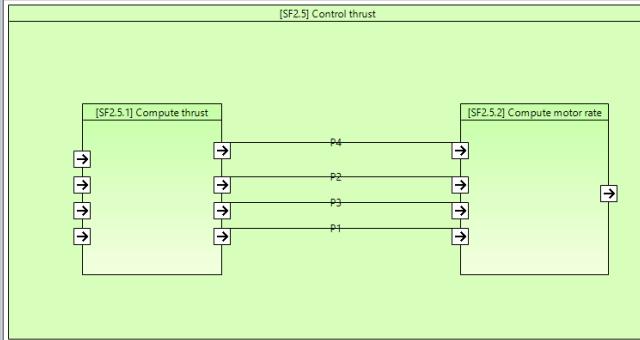
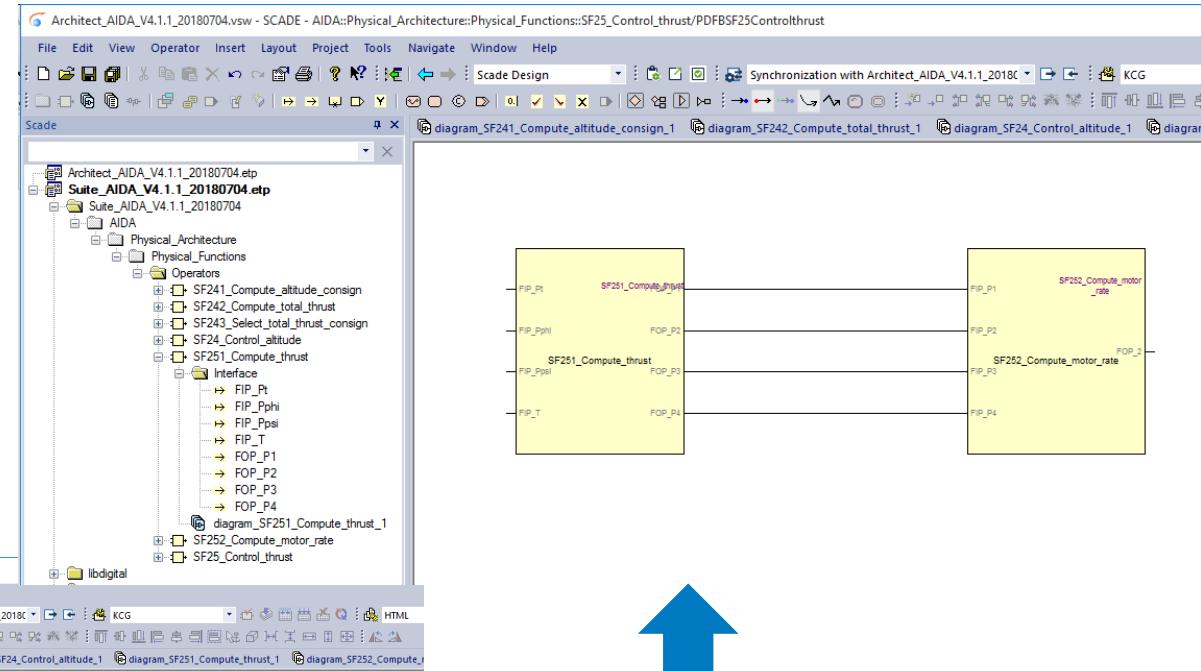
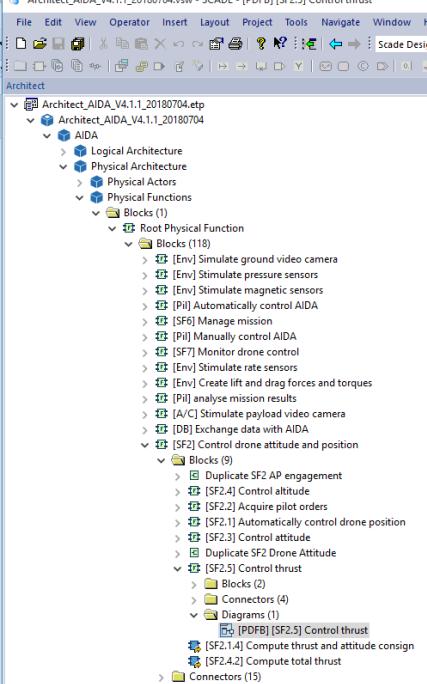


Function ([SF2.5] Control thrust)

workspace - Capella - platform/resource/AIDA_20180704/AIDA.aird/[PDFB] [SF2.5] Control thrust - Capella



Capella to
SCADE Architect

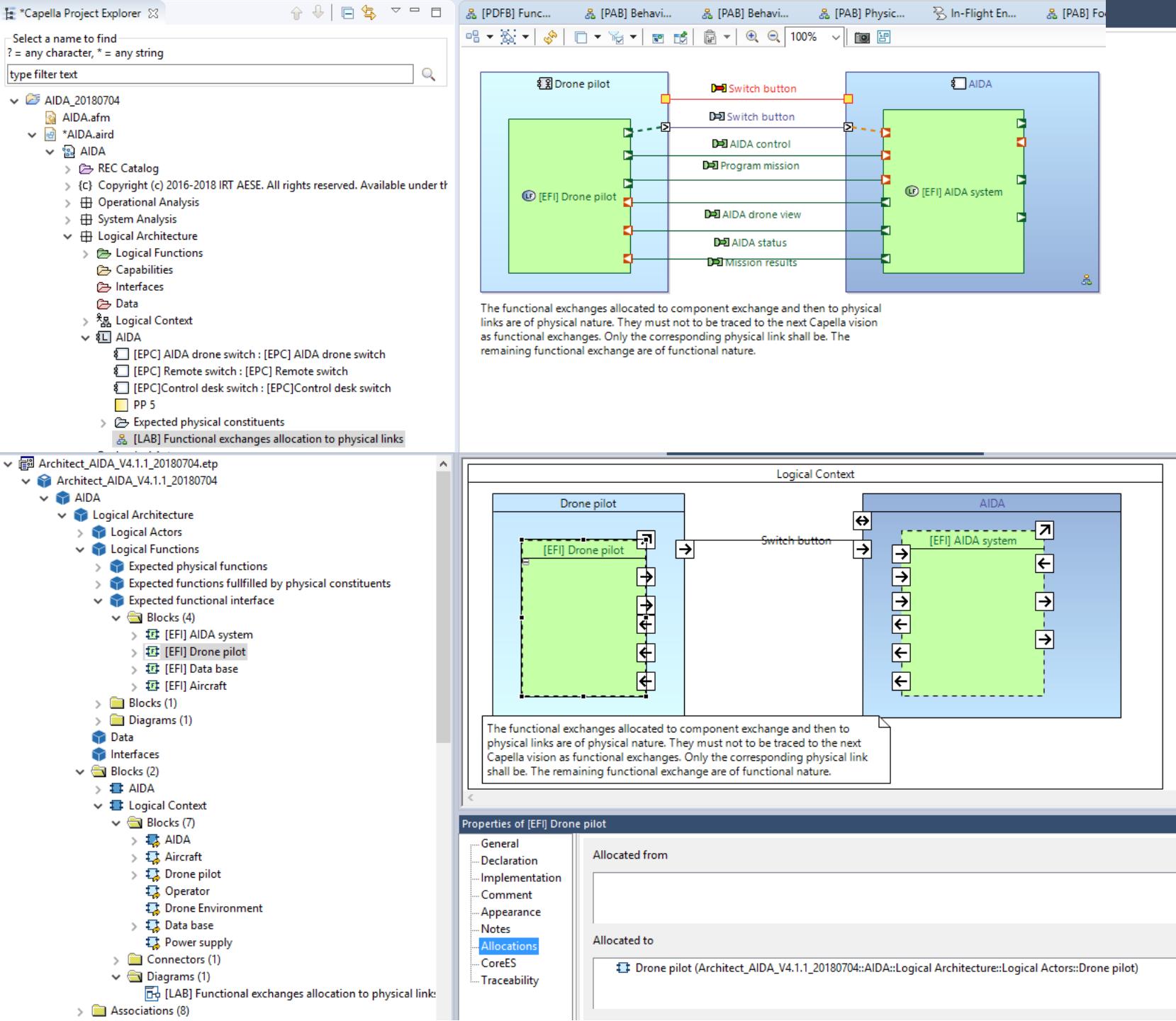


SCADE Architect
to SCADE Suite

Diagrams

Representations of Allocations and references

Capella
SCADE Architect





SCADE Architect – Capella models synchronization

Future evolutions TBD with interested customers

- Capella model scopes
 - Current importer imports the whole Capella model (Logical and Physical levels)
 - Capella internal feature: definition of consistent scopes
 - → Import to SCADE Architect would allow simple selection of a defined scope
- Functional chains
 - Capella allows for the definition of “functional chains”
 - SCADE Architect allows for a similar feature: “data propagation”
 - → Synchronization should translate Capella functional chains to SCADE Architect
- Navigation between projects
 - As done eg between SCADE Architect & SCADE Suite
- Productization

Conclusion

- Best in class industrially deployed tools
 - Capella
 - Established method for Systems Engineering; well guided by IDE
 - Very powerful graphical block diagrams
 - Medini for system safety analysis
 - SCADE for embedded SW architecture, design, code generation and V&V
- Synchronizer tool allows for consistent co-evolutions of models
 - Share what's make sense & value from each model
 - Synchronize = model transformation + diff-merge for incremental use
 - Guidelines to follow to synchronize nice diagrams between tools



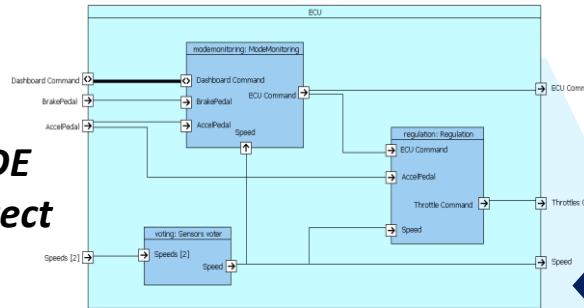
Backup slides



ANSYS Digital Safety & System Simulation Capabilities

Model-Based Systems Engineering

SCADE
Architect

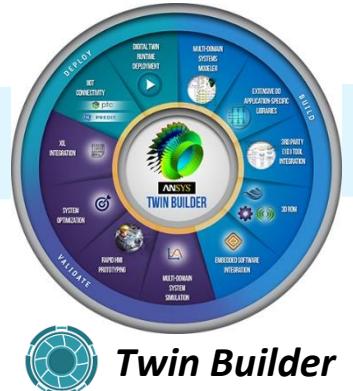


Model-Based System Safety Analysis

medini Analyze

System Architecture

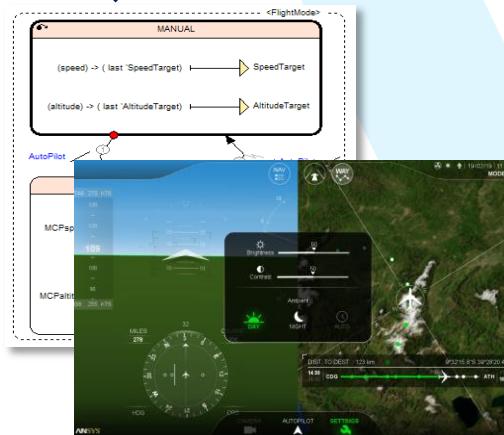
System Simulation & Digital Twins



System/Software Architecture

SCADE
Suite

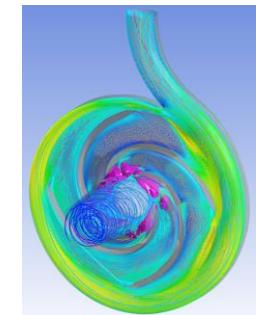
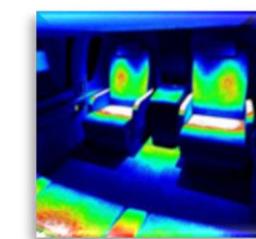
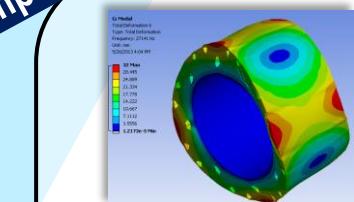
SCADE
Display



Model-Based Software Engineering

Software Components

ROM



SPEOS

3D Physics Simulation