



CLEAN AVIATION



ODE4HERA

Open Digital Environment for Hybrid-Electric Regional Architectures

Systems-as-Code for Optimizing System Architectures

Systems-as-Code Fest | Vilnius, Lithuania | 2025-11-04

Dr. Jasper Bussemaker | German Aerospace Center (DLR)

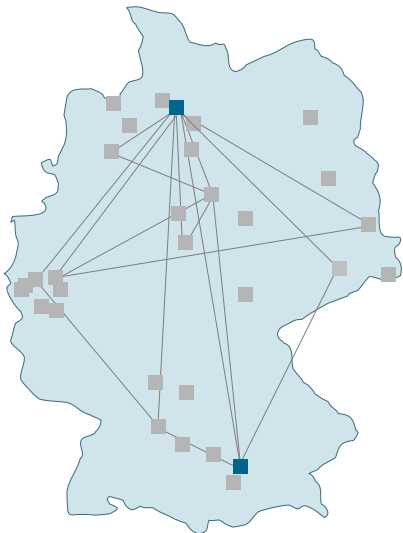


Co-funded by
the European Union

DLR Institute of System Architectures in Aeronautics

German Aerospace Center (DLR)

- 11000 employees
- 53 research institutes
- 30 locations



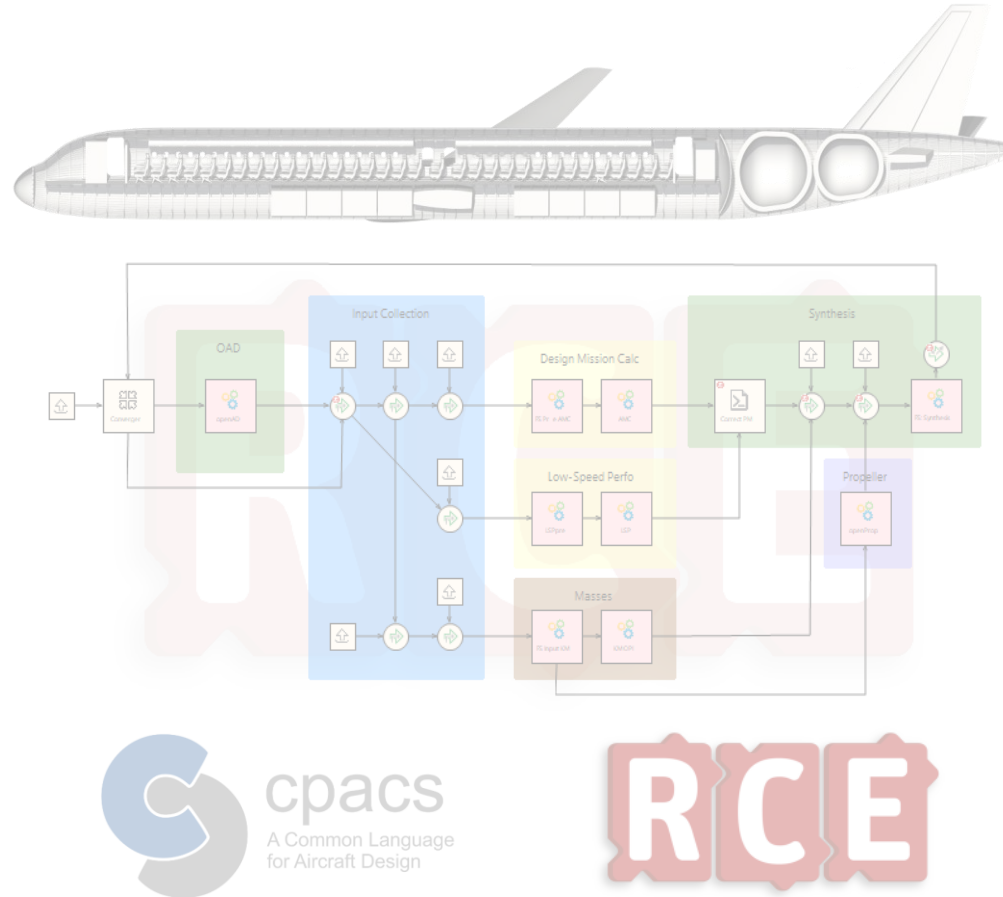
DLR-SL

- Hamburg & Munich
- 110 employees
- Design methods
- Aviation systems

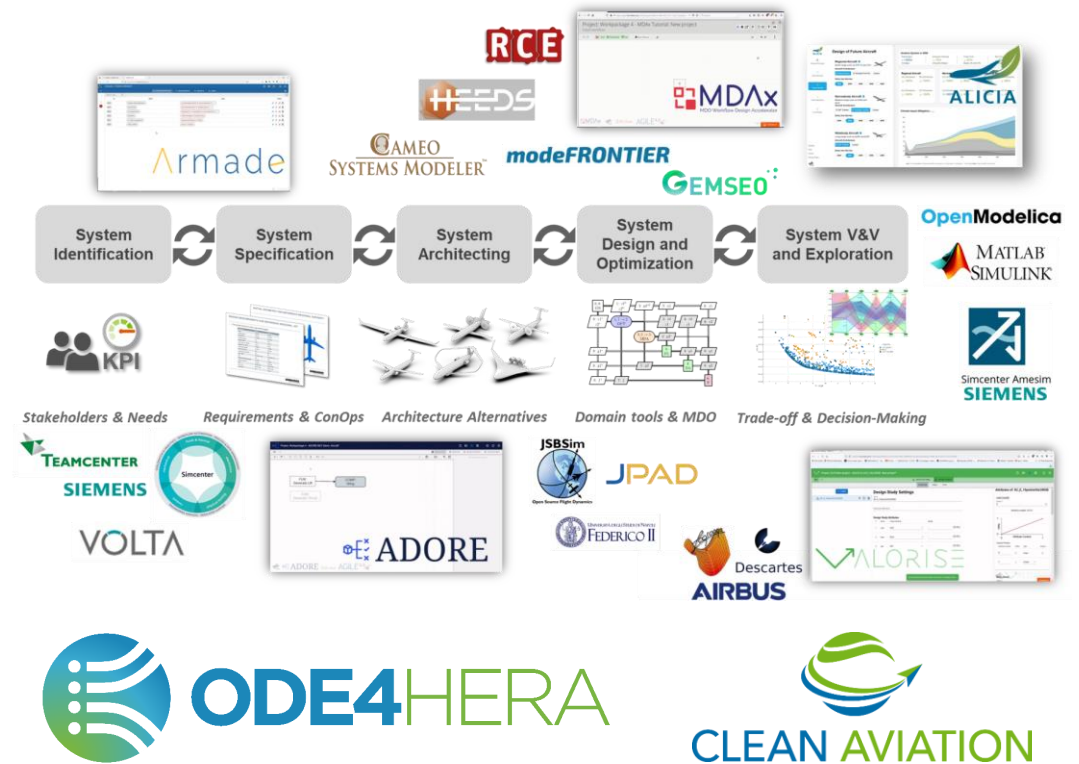


Mastering Complexity through Digital Engineering

20 Years of CPACS



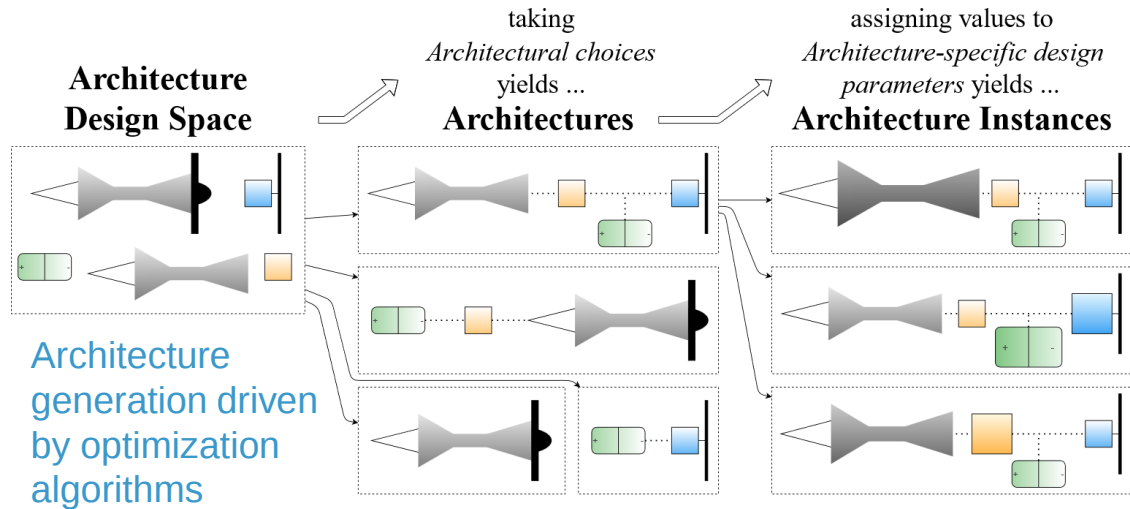
Next Generation Digital Engineering Platform



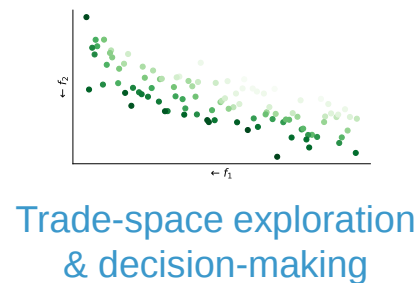
System Architecture Design Space Exploration

MBSE / System Architecting

- Requirements, system functions, parts, parameters, etc.



- Capture architecture-level choices and trade-offs
- Explore full impact of new technologies at the system-level

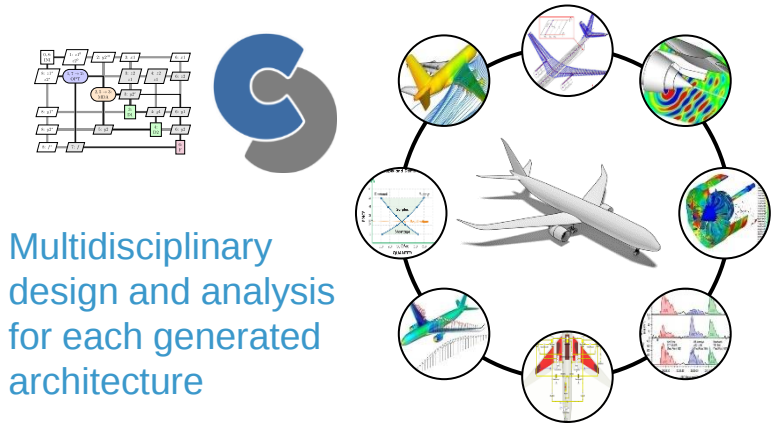


Sync
Architecture +
Parameters

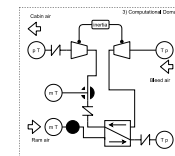


Multidisciplinary System Analysis & Design

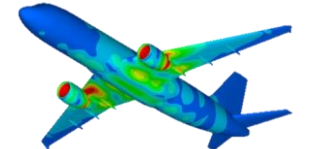
- Physical models, geometry, (dynamic) simulation



fmi Functional Mock-Up Interface



Modelica Language



Resulting
Values / Metrics

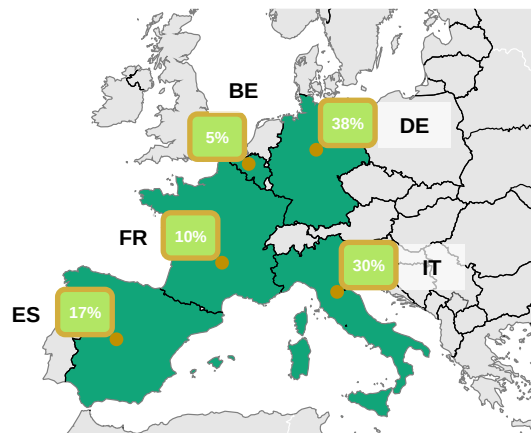


ODE4HERA

Open Digital Environment for Hybrid-Electric Regional Aircraft



€7M EU funding
2024 - 2026



Open Digital Platform

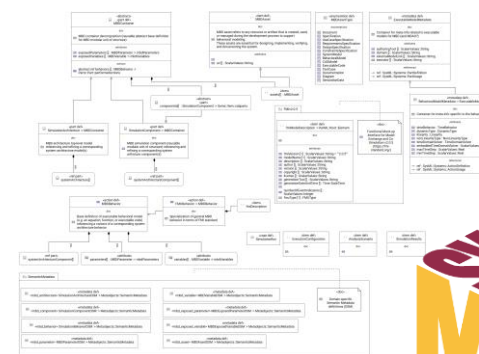


OpenModelica

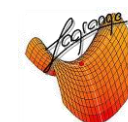


modeFRONTIER

Central Ontology



Simcenter Amesim



Descartes



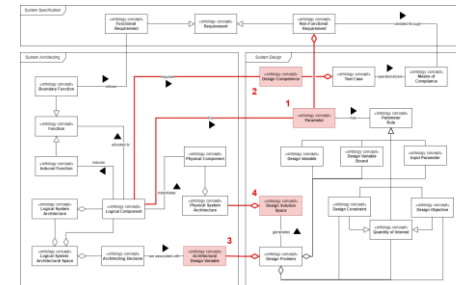
Open Source Flight Dynamics

Ontology Development Process

- Collaborative, accessible development process



- Powerful parser library needed
- Note: we started in 2024!

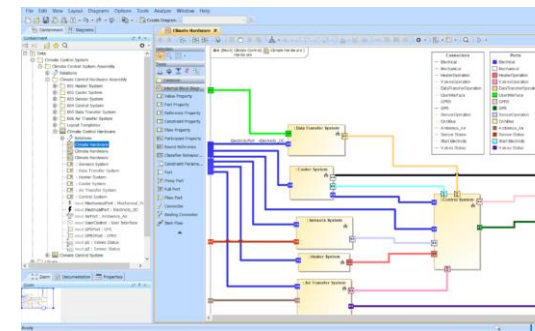


Ontology

```
1 package Sample {
2   import SI::*;
3   import NumericalFunctions::sum;
4
5   enum def Colors {black;grey;red;}
6
7   enum def DiameterChoices :> ISO::LengthValue{
8     enum = 60 [mm];
9     enum = 80 [mm];
10    enum = 100 [mm];
11  }
12
13  part def Vehicle {
14    doc /* This is a doc about vehicle */
15    /* This is a comment */
16    attribute partMasses : ISO::MassValue[*];
17    attribute totalMass = sum(partMasses);
18  }
```

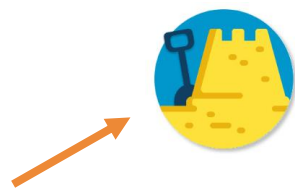
SysML v2: native + extension library

Connect
authoring tools



Elucidation	ID	Name	Statement	Links
Specification	REQ-1	30_000	The Thermal Management System shall exhibit volume < 2.5 m³	30_000
Functional Requirement	REQ-2	30_001	The Thermal Management System shall exhibit energy consumption < 3000 kWh	30_001
Performance Requirement	REQ-3	30_002	The Thermal Management System shall exhibit mass < 100 kg	30_002

Architecture Design Ontology



Soon on
Sysand

```
private import ArchitectureDesign::*;

action def GenerateThrust {
  attribute seaLevelThrust default 200e3[SI::N];
}

part def Turbofan {
  #contX attribute bypassRatio {
    :>> lowerBound = 5.0; :>> upperBound = 12.0; }

  #evaluationOutput attribute weight : ISQ::MassValue {
    :>> role = OutputRole::Metric; }

  #evalOut attribute specificFuelConsumption {
    :>> role = OutputRole::MinimizationObjective; }
}

part def Turboprop {
  #disX attribute nrBlades { :>> values = (3, 4, 6, 8); }

  #evalOut attribute bladeTipMach : ISQ::MachNumberValue {
    :>> role = OutputRole::Constraint;
    assert constraint { that <= 0.9 }
  }
}
```

Part
definitions with
design
parameters

Architecture
design space

```
#architectureDesignSpace part designSpace {

  action generateThrust : GenerateThrust;

  part turbofan : Turbofan {
    perform generateThrust;
    ref #needsAction action
      references provideFuel;
  }

  part turboprop : Turboprop {
    perform generateThrust;
    ref #needsAction action :>> provideFuel;
  }

  action provideFuel;
  part fuelSystem { perform provideFuel; }
}
```

Architecture
instance derived
from design
space

```
#architecture part turbofanPropSys
  subsets designSpace {

    action :>> generateThrust {
      :>> seaLevelThrust = 200e3[SI::N];
    }

    part :>> turbofan {
      :>> bypassRatio = 10.0;
      :>> weight = 5000[SI::kg];
      :>> specificFuelConsumption;
    }

    part :>> turboprop[0];
  }
```

System Architecture Optimization with ADORE

Architecture **design space** modeler

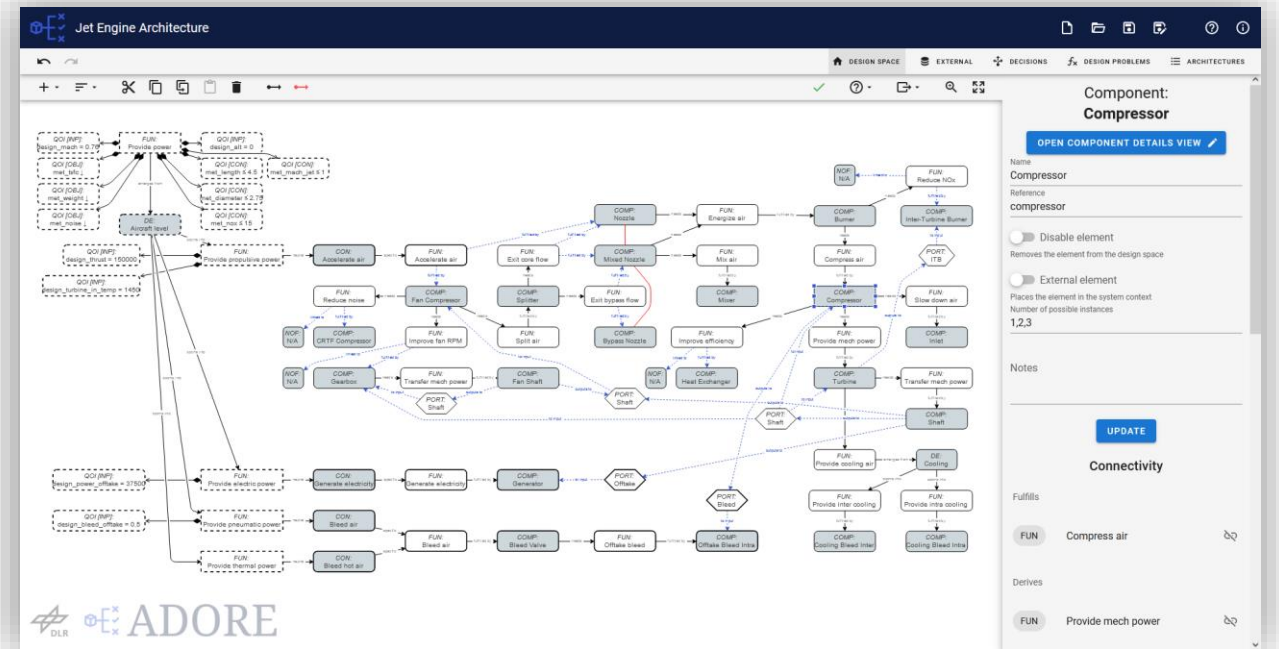
- Define functions, components, connections
- Identify architectural choices
- Define input parameters and metrics

Architecture **generator**

- Take architectural choices to create architectures
- Connect to evaluation environment

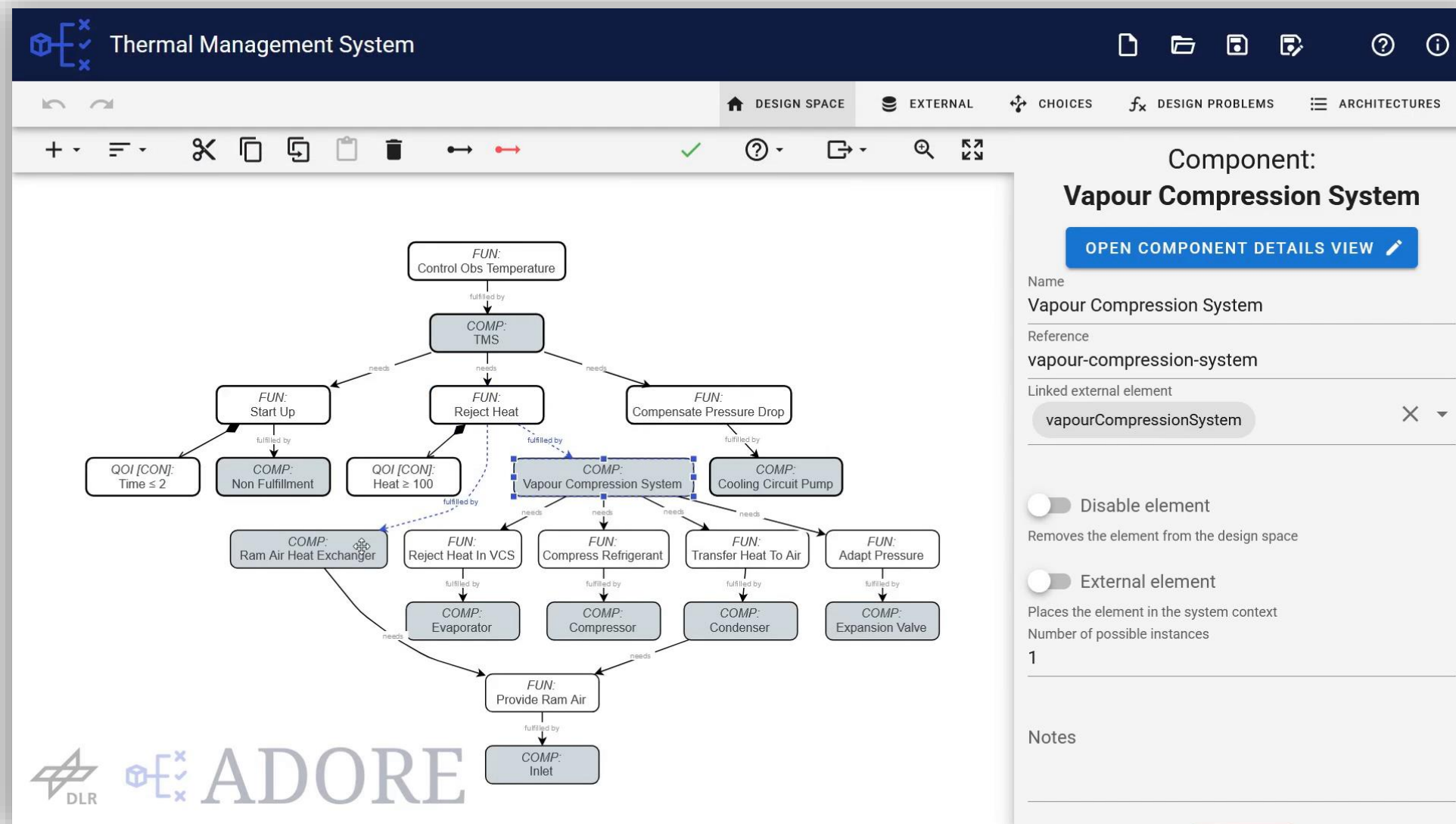
Architecture **optimization** framework

- Define design variables, objectives, constraints
- Connect to optimization algorithms



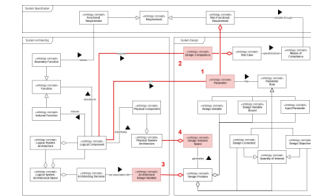
<https://adore.mbse-env.com/docs/>

ADORE SysML v2 Demo



Conclusions

- Next generation digital engineering platform based on **SysML v2**
- Automated architecture design space exploration with **ADORE**
- Ontology development using git, **Syside Editor** and **Sysand**
- Tool connections to SysML v2 using **Syside Automator**



Ontology



```
1 package Sample {
2   import SI1;
3   import NumericalFunctions::sum;
4
5   enum def Colors {black,gray,red};
6
7   enum def DiameterChoices {> ISO::LengthValue{
8     enum = 60 (mm);
9     enum = 80 (mm);
10    enum = 100 (mm);
11  }
12
13  part def Vehicle {
14    doc:/* This is a doc about vehicle */
15    /* This is a comment */
16    attribute partMasses : ISO::MassValue[4];
17    attribute totalMass = sum(partMasses);
18  }
```

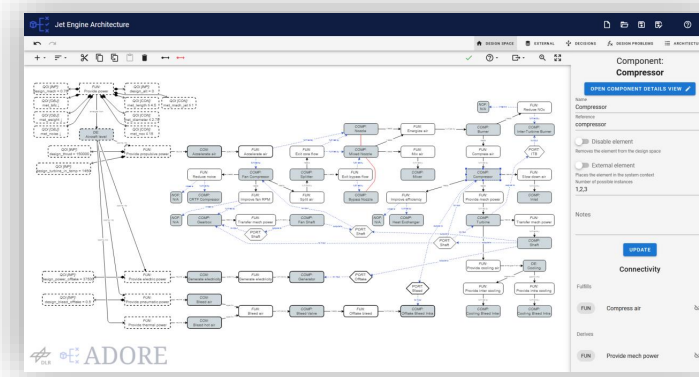
SysMLv2: native + extension library

Connect
authoring tools



Syside
Automator

 **ADORE**



<https://adore.mbse-env.com/docs/>