

ADORE

GUI Quick Reference Manual

Updated: January 2026

German Aerospace Center (DLR)

Institute of System Architectures in Aeronautics, Hamburg, Germany

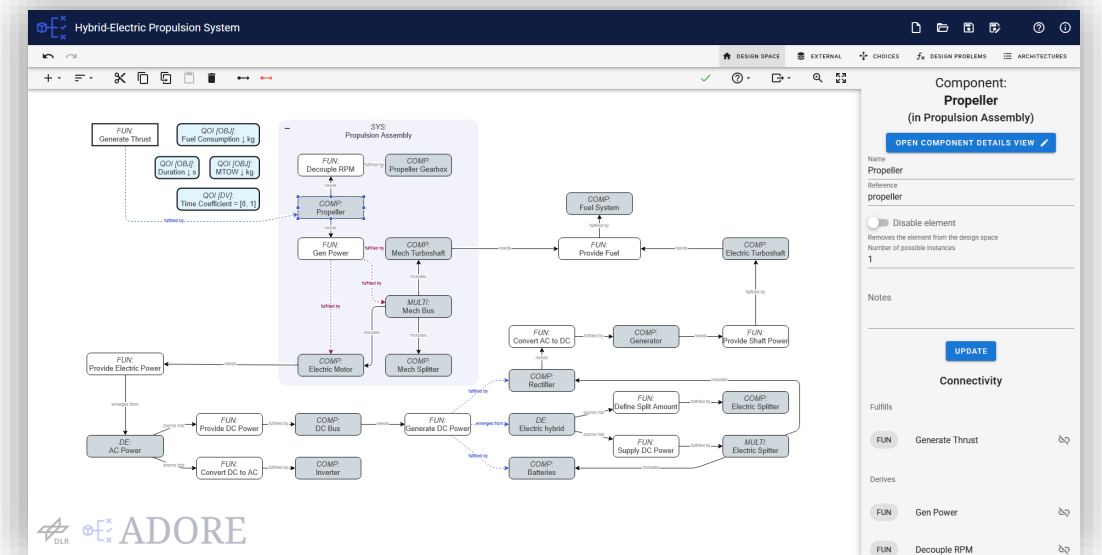
Jasper Bussemaker (jasper.bussemaker@dlr.de), Luca Boggero



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



This allows the formulation and execution of System Architecture Optimization (SAO) problems.



For more background information about system architecture optimization, refer to: [System Architecture Optimization: Function-Based Modeling, Optimization Algorithms, and Multidisciplinary Evaluation](#)

ADORE Graphical User Interface



Project title

Export project/view

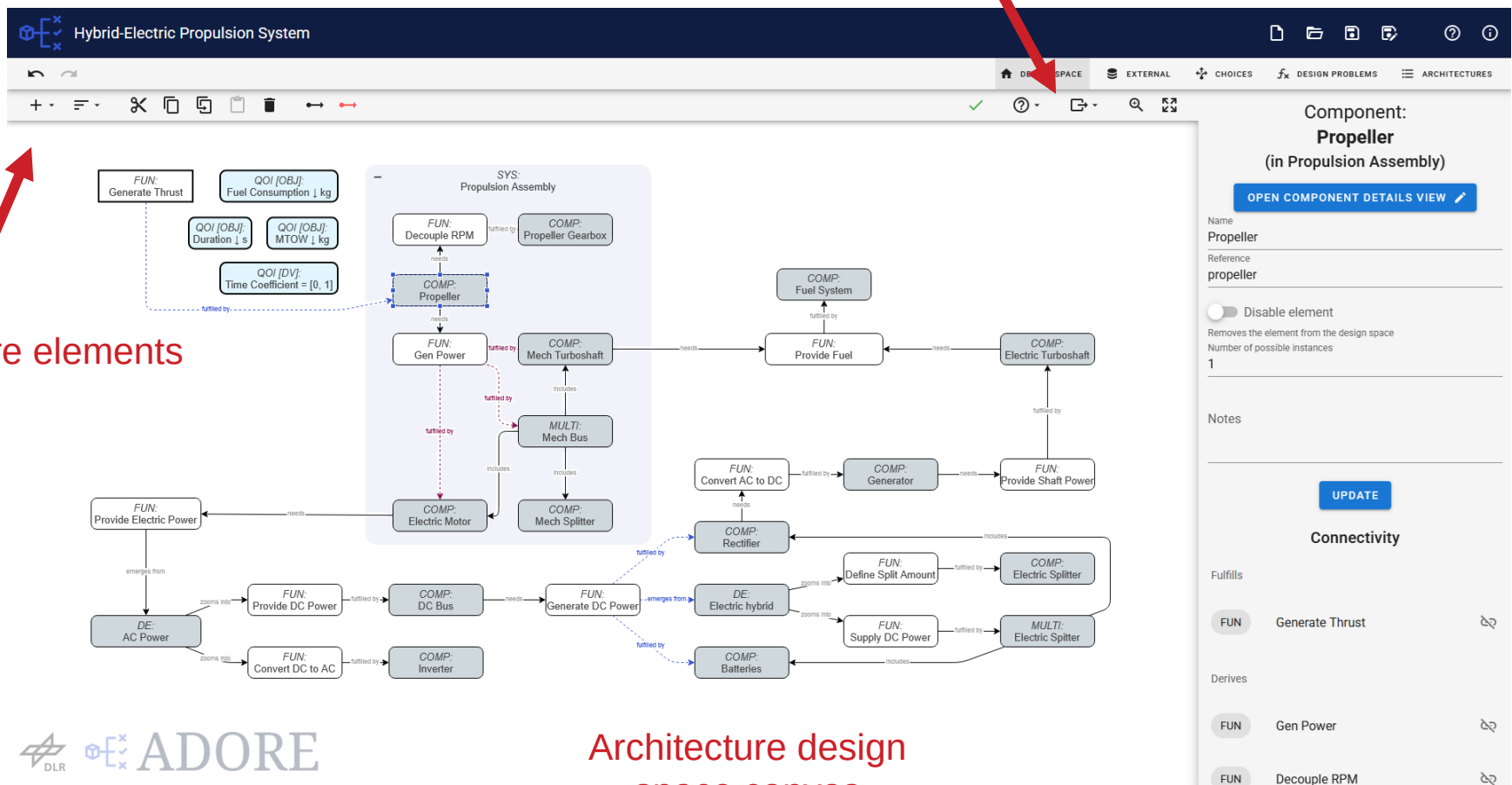
Project buttons

Undo/redo

Navigation

Add architecture elements

Element editing pane



Architecture design
space canvas

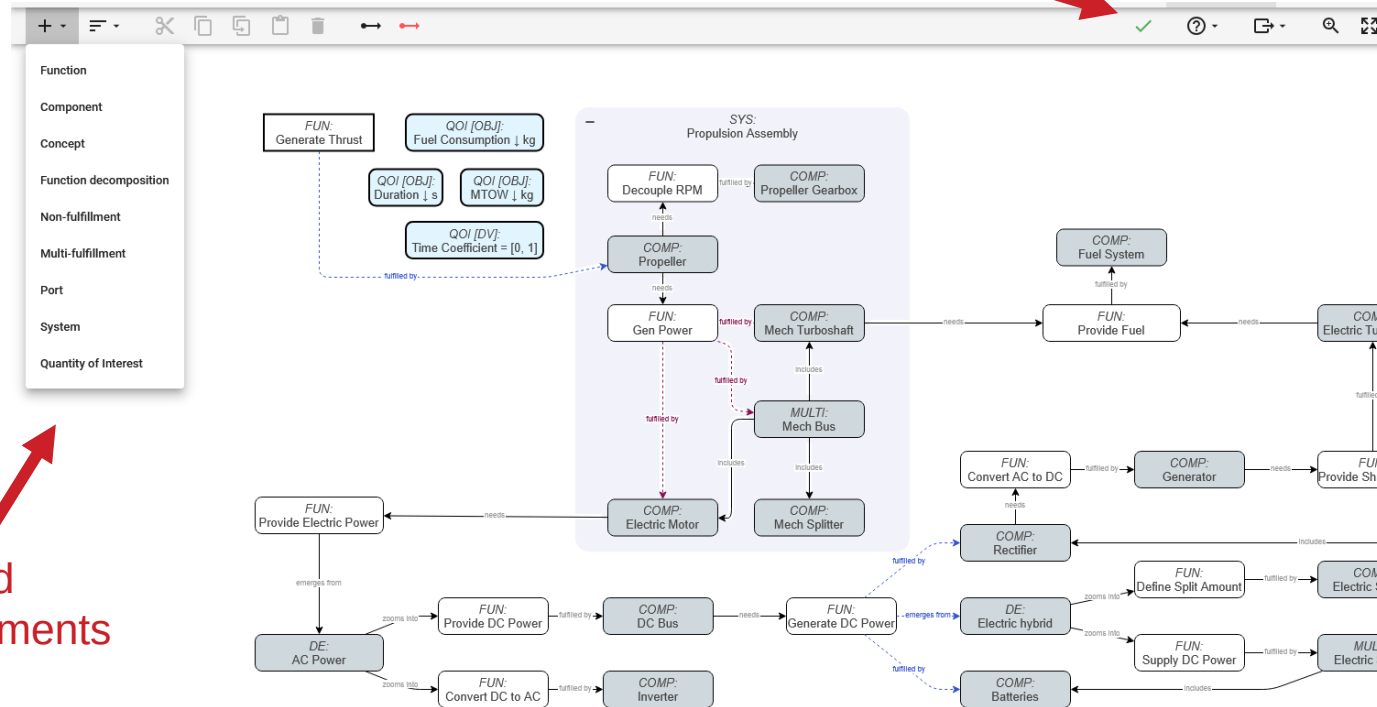
Editing Design Space Elements

Auto layout

Cut/copy/duplicate/delete selected element

Model status indicator

Element quick reference



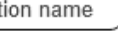
Draw connection

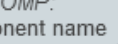
Draw incompatibility constraint

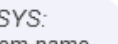
Drag from source to target

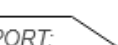
The highlight color indicates if a connection is allowed

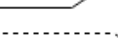
Legend

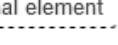

Function or quantity of interest


Component, concept, decomposition or non/multi-fulfillment


Subsystem


Port


Element in system context


Element with issue(s)


Element exists unconditionally


Component-specific node


Derivation or connection


Composition ("has")


Architectural choice


Constrained choice


Incompatibility constraint

Element Types

Symbol	Element Type
ATTR	Attribute
COMP	Component
CON	Concept
DE	Function decomposition
EXT	External element
FUN	Function
INST	Component instance
MULTI	Multi-fulfillment
NOF	Non-fulfillment
PORT	Port
QOI	Quantity of Interest
SYS	System
VAL	Attribute value

Advanced Mode

Advanced mode allows you to edit component "attributes", and change the number of port connector instances.

Advanced mode

Linking External Elements

External elements (e.g. requirements, metrics, components) can be connected to using **external databases**.

It depends on your design problem and integration environment how these databases are made available.

The screenshot displays the ADORE External Database Demo interface. On the left, a sidebar shows a list of external databases: 'Boundary functions' (updated 09-02-2022 08:22) and 'Design constraint requirements' (updated 09-02-2022 08:22). The main area is titled 'External Database: Boundary functions' and shows a table of 'External Elements (2/6 linked)'. The table has columns for '#', 'Name', 'Element types', and 'Linked to'. The first row shows 'Boundary function: Contain passengers' linked to 'Contain passengers'. The second row shows 'Boundary function: Accomodate passengers' linked to 'Accomodate passengers'. The third row shows 'Boundary function: Regulate pressure' with a red 'X' in the 'Linked to' column. The fourth row shows 'Boundary function: Regulate temperature' with a red 'X' in the 'Linked to' column. The fifth row shows 'Boundary function: Regulate humidity' with a red 'X' in the 'Linked to' column. The sixth row shows 'Boundary function: Transport passengers' with a red 'X' in the 'Linked to' column. On the right, a panel shows the details of the selected element, 'Boundary function: Contain passengers', including its reference, type, and overridden properties.

#	Name	Element types	Linked to
1	Boundary function: Contain passengers	Function	Contain passengers
2	Boundary function: Accomodate passengers	Function	Accomodate passengers
3	Boundary function: Regulate pressure	Function	X
4	Boundary function: Regulate temperature	Function	X
5	Boundary function: Regulate humidity	Function	X
6	Boundary function: Transport passengers	Function	X

External element: **Boundary function: Contain passengers**
Ref: 5e9f453d-3192-401d-a58a-96e2a52554be

Can be linked to
Function

Overridden properties

Property	Value
Name	Contain passengers
Reference	contain-passengers
Type	Solution neutral
Boundary function	Yes

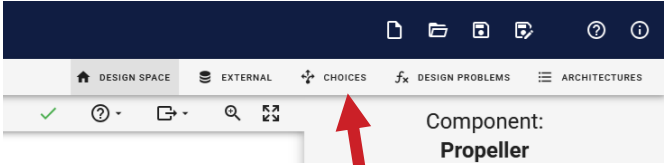
External elements may **override** properties of the elements they are linked to.
The **types** of elements they can be linked to might be restricted.

Inspecting Architectural Choices



Gives an overview of defined architectural choices. Also allows defining constraints between choices.

Statistics about the design space can be calculated. The “valid” design space size represents the number of valid (discrete) architectures in the design space.



Click to refresh statistics

Architecture Choices

Search

# ↑	Operation	Subject	Component Instance	System Instance	Options	Constraints
1	Fulfill function	Generate DC Power			Electric hybrid, Batteries, Rectifier	
2	Fulfill function	Gen Power		1	Electric Motor, Mech Bus, Mech Turboshaft	
7	Instantiate system	Propulsion Assembly			1, 2, 3, 4 or 5 times	
11	Continuous design variable	COMP: Electric Splitter -> elec_DoH_climb				
12	Continuous design variable	COMP: Electric Splitter -> elec_DoH_cruise				
13	Continuous design variable	COMP: Electric Splitter -> elec_DoH_descend				
14	Continuous design variable	COMP: Electric Splitter -> elec_DoH_loiter				
15	Continuous design variable	COMP: Electric Splitter -> elec_DoH_reserve_climb				
16	Continuous design variable	COMP: Electric Splitter -> elec_DoH_reserve_cruise				
17	Continuous design variable	COMP: Electric Splitter -> elec_DoH_reserve_descend				
18	Continuous design variable	COMP: Mech Splitter -> mech_DoH_climb				
19	Continuous design variable	COMP: Mech Splitter -> mech_DoH_climb				
20	Continuous design variable	COMP: Mech Splitter -> mech_DoH_climb				

Choice Constraint Types

Choices can be linked by a constraint, which limits the options that can be selected when taking all choices within a constraint into account.

As an example, consider two choices with three options (A, B, C) each:

Constraint	Description	Possible Combinations
None	All combinations of options	AA, AB, AC, BA, BB, BC, CA, CB, CC (9)
Linked Values	All choices get the same option	AA, BB, CC (3)
Permutations	All combinations without repeating values	AB, BA, AC, CA, BC, CB (6)
Unordered Combinations	All combinations when selection order is not relevant	AA, AB, AC, BB, BC, CC (6)
Unordered Non-replacing Combinations	Unordered, without repeating values	AB, AC, BC (3)

Statistics

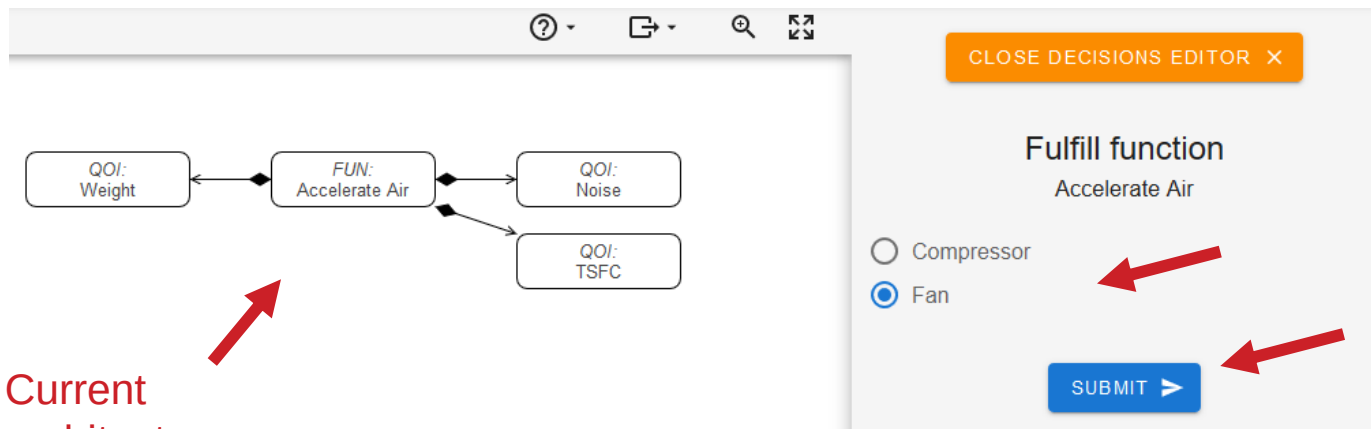
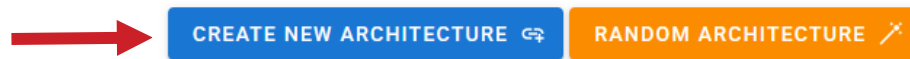
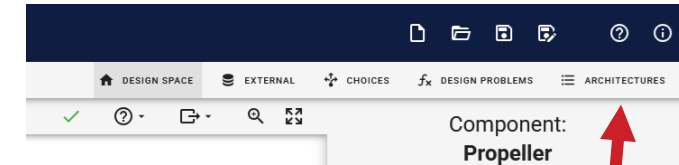
Click to refresh statistics

Decisions	58
Valid Design Space (Discrete)	316
Declared Design Space (Discrete)	7 290
Discrete Choices	8
Discrete Imputation Ratio	23.1
Information Index	59%
Continuous Dimensions	43
Mean Active Continuous Dimensions	42.9
Continuous Imputation Ratio	1
Imputation Ratio	23.1
Encoding Algorithm	Fast
Valid Design Space (Selection-choices)	315
Declared Design Space (Selection-choices)	455 625
Selection-choices	10
Information Index (Selection-choices)	51%
Imputation Ratio (Selection-choices)	1 446.4

Manually Creating Architecture Instances

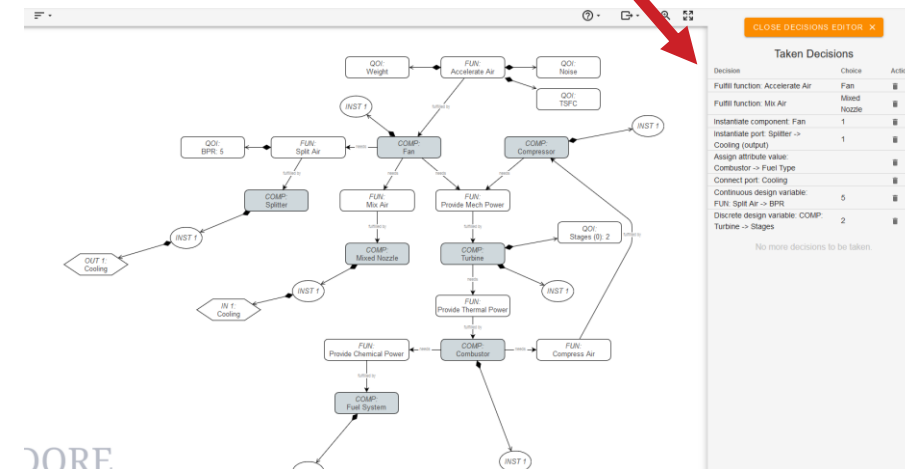


Manually creating architecture can be helpful for validating the design space and testing evaluation interfaces.



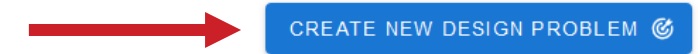
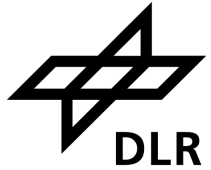
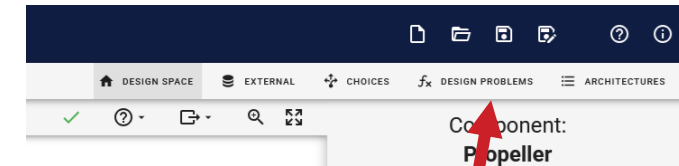
Current architecture instance

Until all choices have been taken



Defining Design Problems

Design problems are a **formalization** of the design space in terms of design variables, objectives, constraints. They are used by optimization algorithms.



Choices and QOIs mapped to design variables

QOIs mapped to objectives and constraints

Design Problem: New Design Problem

Design Variables

#	Name	Type	Source	Options	Fixed Value	Actions
1	func_fulfill	Discrete	Decision #1	2	C: FAN	
2	func_fulfill_2	Discrete	Decision #2	2		
3	comp_instance_FAN	Discrete	Decision #3	2		
4	port_needed_COOLING	Discrete	Decision #4	2		
5	comp_atl_COMBUSTOR_FUEL_TYPE_0	Discrete	Decision #5	3		
6	port_COOLING_0	Discrete	Decision #6	5		
7	port_COOLING_1	Discrete	Decision #6	3		
8	BPR_0	Continuous	QOI: BPR	Between 3 and 12	5	
9	STAGES_0	Discrete	QOI: Stages	4		

Objectives

#	Name	Source	Active	Direction	Actions
1	TSFC	QOI: TSFC	✓	Minimize	
2	Weight	QOI: Weight	✗	Minimize	

Constraints

#	Name	Source	Active	Direction	Reference Value	Actions
1	Noise	QOI: Noise	✓	Lower than or equal to	105	

Generic Output Metrics

#	Name	Source	Actions
No data available			

Design Variable Settings

func_fulfill_2

Type: Discrete

Fixed value

Notes

Optionally fix design variable values

QOI type can be changed to objective, constraint or they might be deactivated

