Optimize architecture by generating N2 Design Structure Matrices with DSM4Capella

Webinar Capella

DUBE Sébastien – sebastien.dube@samares-engineering.com

OJEDA Mirna – mirna.ojeda@samares-engineering.com

Samares-Engineering Overview

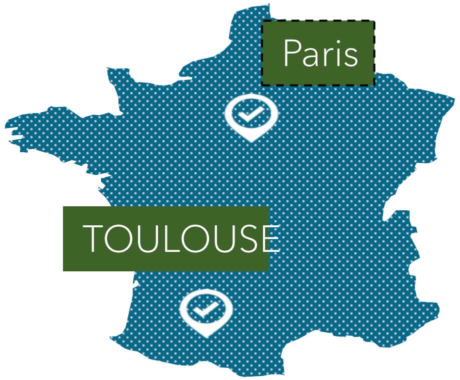
20 Employees

Certified SE Professionals

17 Experts and Consultants in MBSE

90% Europe – 11 countries

10% UK and USA

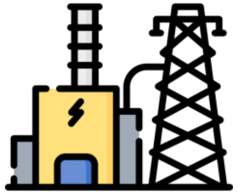


Main Industrial Domains

Defense

Automotive

Space



Energy

Avionics

SAMARES Mission

Deliver Systems Engineering (SE/MBSE) Expertise and services

SAMARES ENGINEERING
Accelerate Systems Design

Scoping & Assessment

SE practices assessment

SE Cost saving identification

SE Tool assessment

SE Improvement plan definition

Delivery & Follow-up

Qualiopi processus certifié
RÉPUBLIQUE FRANÇAISE

SE, MBSE
PLE
Trainings,
coaching,
support

Learning Center

SE tools Customization

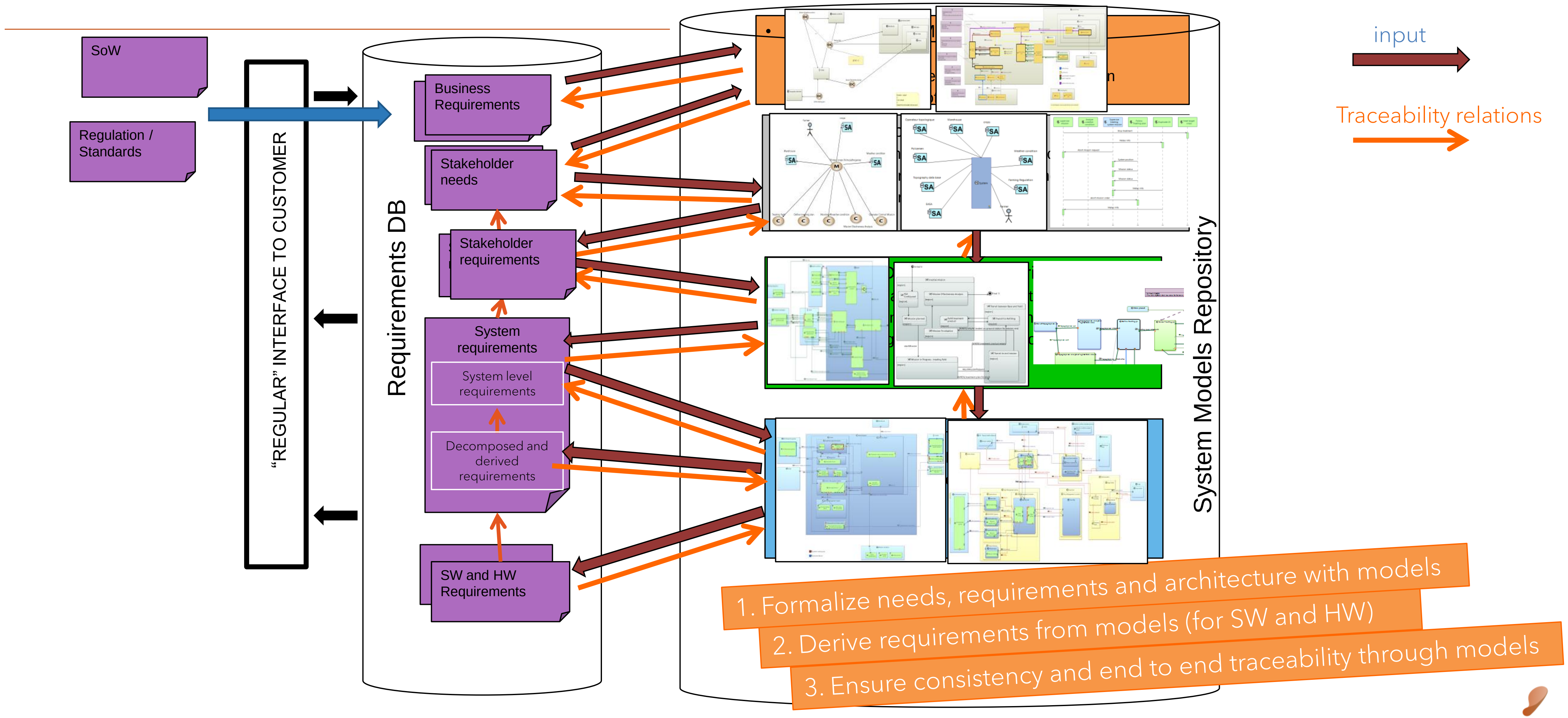
Product Line Engineering

MIC with parameters

Digital continuity

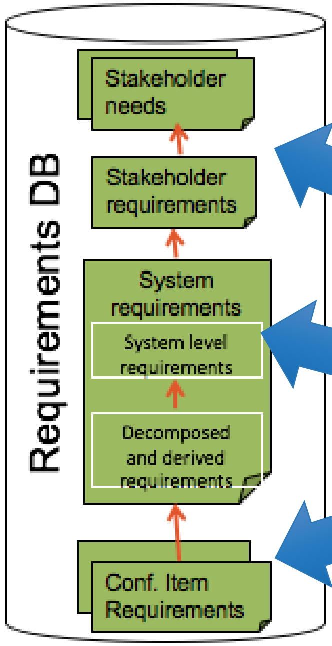
[Samares-Engineering Web Site]

Our vision Requirements and MBSE (1/2)

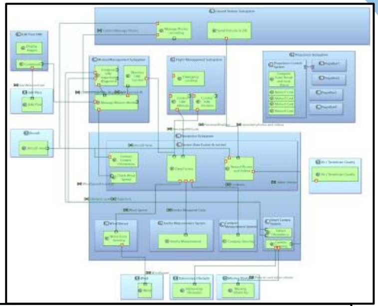


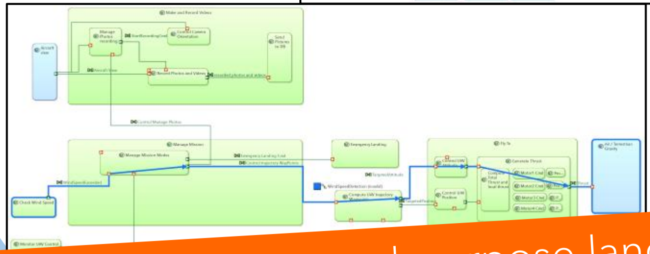
Our vision – digital continuity (2/2)

Capella is a quite good language/tool to formalize requirements and system architecture

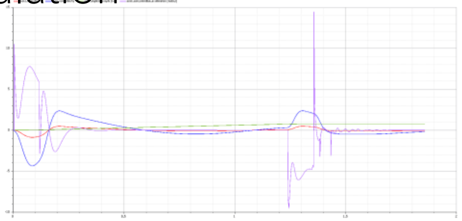


Business, stakeholder and system requirements

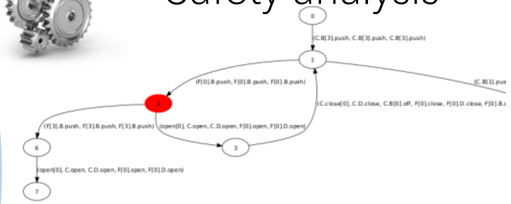




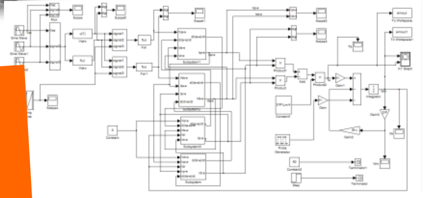
Multi physical analysis and simulation



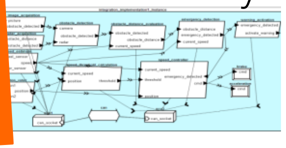
Safety analysis



Control command algorithms



Real time analysis



Mechanical aspects, sizing constraints

But Capella remains a general-purpose language : it must be extended or completed (with other languages and tools) to address the physical world, 3D positioning, detailed behavior, constraint equations between key system parameters, and to perform specific analyses and verifications

Coupling Optimization
(Work presented at [ERTS2024 conference])

Problematic of interfaces coupling

Multiplicity of interfaces and involved physics

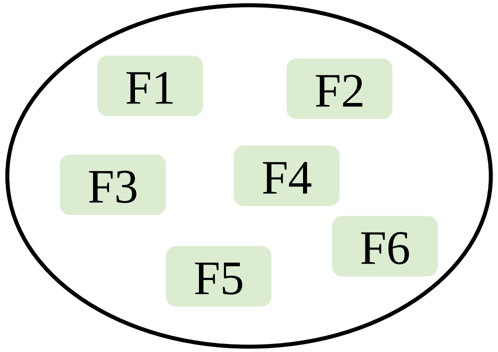
As stated in standards such as ISO26262, it is recommended to

« Avoid unnecessary complexity of interfaces »
ISO26262-4 Product Development at System Level

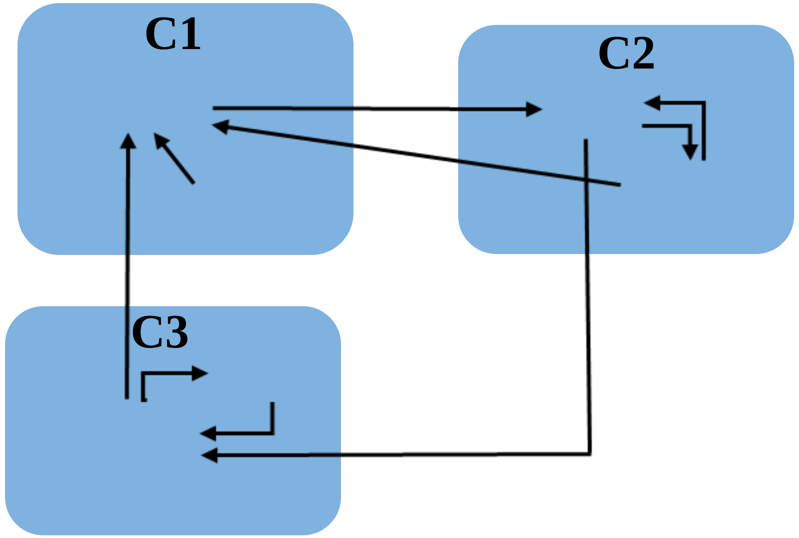
[related paper presented at ERTS2024 conference [0]]

Fundamentals

Context overview



Functions



Logical components

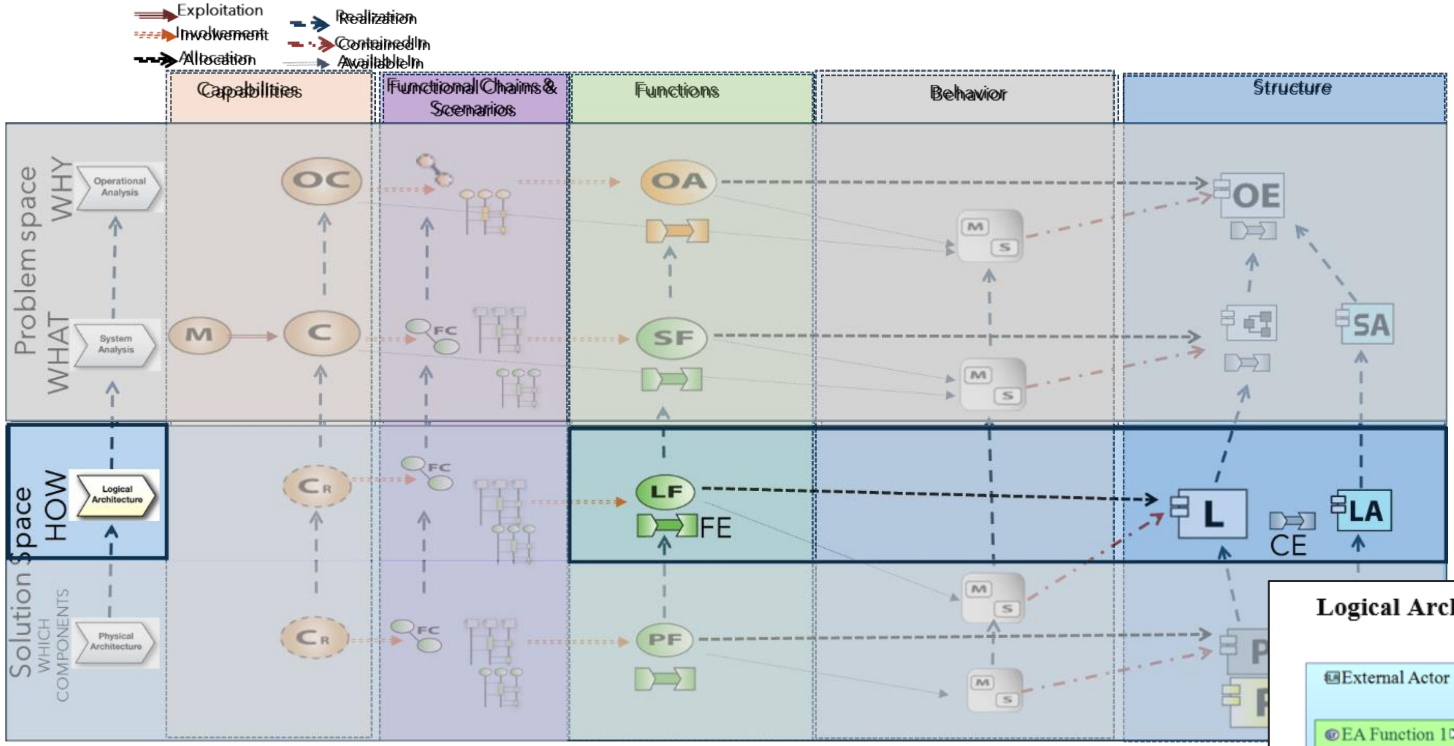
Interfaces

How to manage minimization of interfaces between components for large systems (high number of functions/components) ?

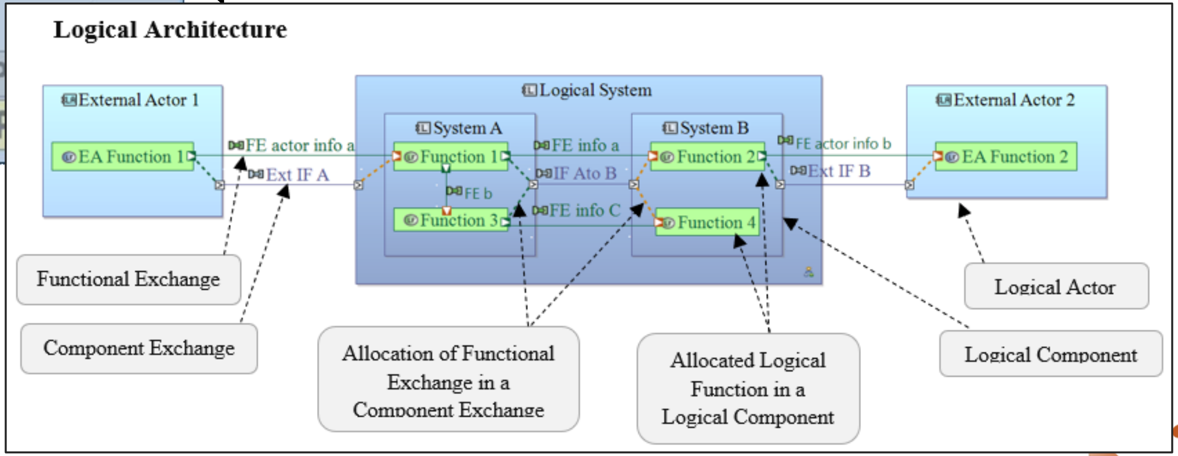
Fundamentals

Architecture Analysis & Design Integrated Approach

ARCADIA method defines systems engineering concepts from needs analysis to architectural solution definition



Concept illustrated

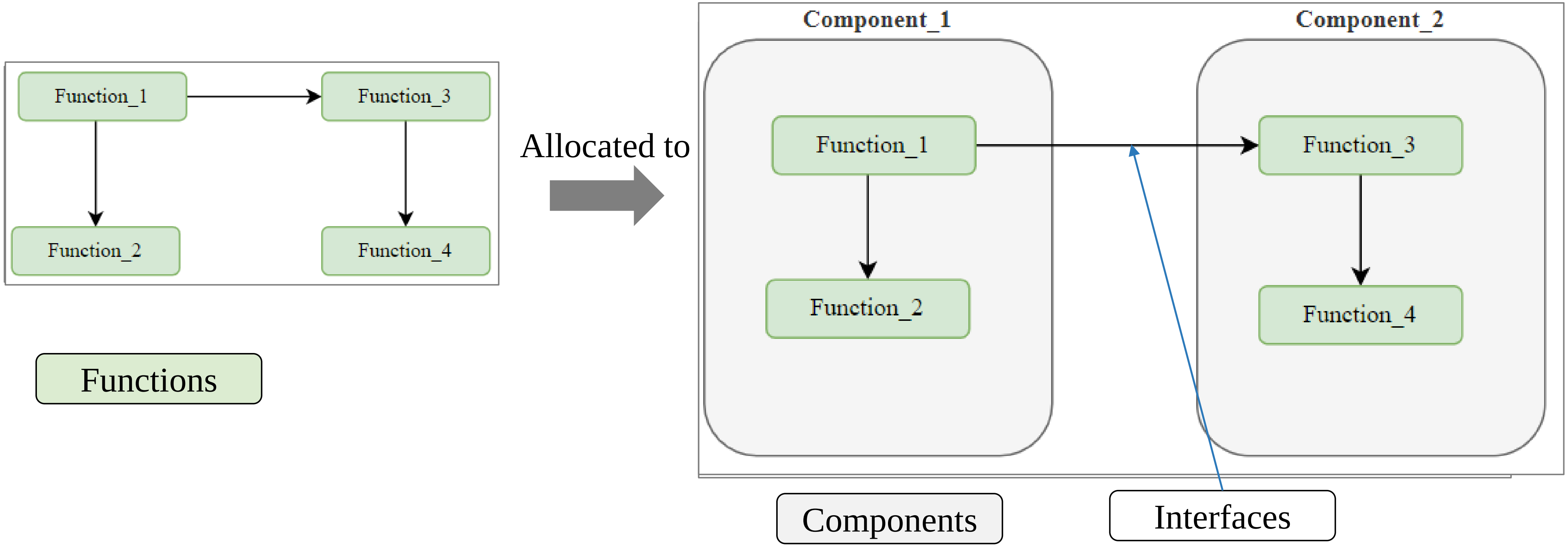


Fundamentals

Logical architecture

Is an intermediate step to define a system architecture from stakeholder needs analysis.

Describes functions closer to their eventual physical implementation, guided by functional dependencies.



Fundamentals

Design Structure Matrix (DSM)/N2

Is a graphical representation used in Systems Engineering to analyse and visualize the relationships and dependencies between different components within a system [1].

Input

Output

	F1	F2	F3	F4	F5	F6
F1	**F1**	1	0	0	0	0
F2	0	**F2**	1	0	1	0
F3	1	1	**F3**	0	0	0
F4	0	0	0	**F4**	1	0
F5	1	0	0	1	**F5**	0
F6	1	0	0	0	0	**F6**

1: there is at least one interface between the functions (column, row)
0: no interaction

Fundamentals

Design Structure Matrix mechanism for optimization as defined in Systems Engineering Handbook [1]

Output \ Input	F1	F2	F3	F4	F5	F6
F1	F1	1	0	0	0	0
F2	0	F2	1	0	1	0
F3	1	1	F3	0	0	0
F4	0	0	0	F4	1	0
F5	1	0	0	1	F5	0
F6	1	0	0	0	0	F6

1: there is at least one interface between the functions (column, row)
0: no interaction

Output \ Input	F1	F2	F3	F4	F5	F6
F1	F1	1	0	0	0	0
F2	0	F2	1	0	1	0
F3	1	1	F3	0	0	0
F4	0	0	0	F4	1	0
F5	1	0	0	1	F5	0
F6	1	0	0	0	0	F6

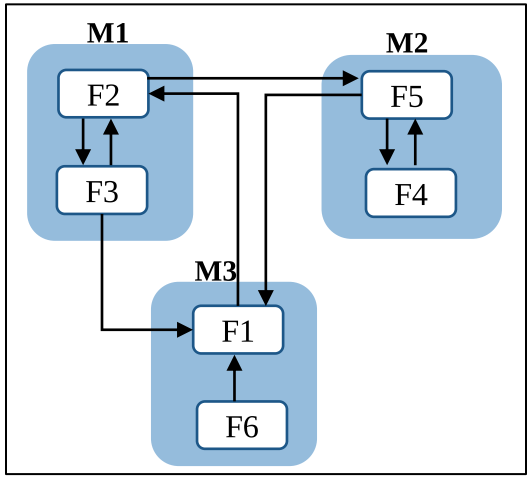
M1 M2 M3

Total 8 interactions between modules

Output \ Input	F2	F3	F5	F4	F1	F6
F2	F2	1	1	0	0	0
F3	1	F3	0	0	1	0
F5	0	0	F5	1	1	0
F4	0	0	1	F4	0	0
F1	1	0	0	0	F1	0
F6	0	0	0	0	1	F6

M1 M2 M3

Total 4 interactions between modules



The metrics of interaction complexity is known as coupling value

Approach Proposed

The coupling value

The coupling value serves as an assessment of the complexity of coupling between logical components, derived from a formula based on software coupling metrics.

Coupling value formula

$$Coupling(C_{M_k}) = 1 - \frac{1}{d_i + 2 \cdot c_i + d_o + 2 \cdot c_o + \omega + r}$$

Equation 1 - Coupling Value of a Logical Component

$$CouplingValue(C_v) = \sum_{k=1}^{n} [C_{M_k}]$$

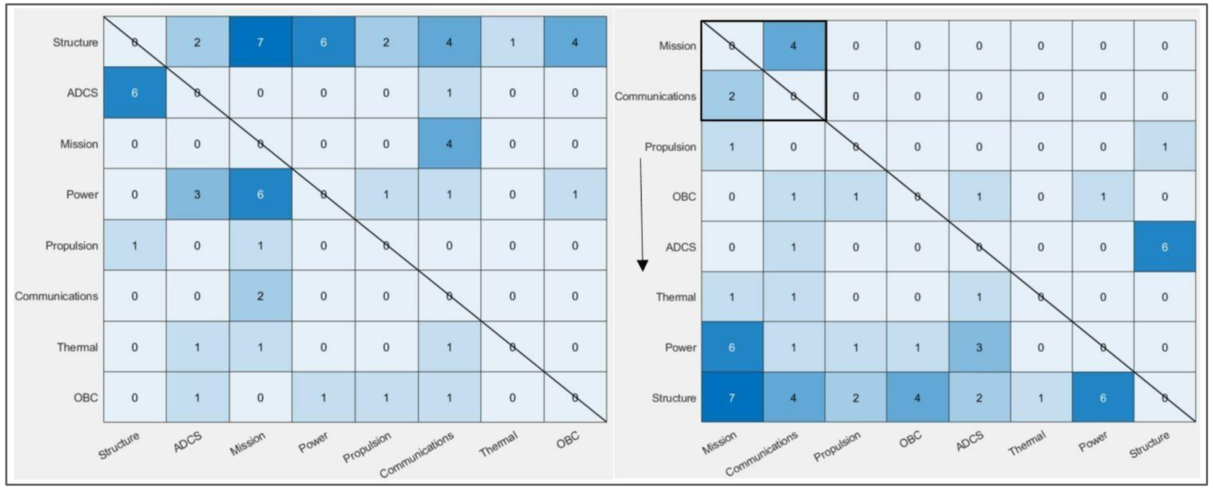
Equation 2 - Coupling Value of the Complete Logical Architecture

M_k is the logical component under consideration
d_i is the number of input data parameter
c_i is the number of input control parameters
d_o is the number of output data parameters
c_o is the number of output control parameters
ω is the number of modules called (fan-out)
r is the number of calling the module under consideration (fan-in)

Source: [4]

Related work

- Integration of N2 matrices within MBSE environments [2]



Focus on extraction of N2 matrice from MBSE model without optimization

- Integration of DSM generation with genetic algorithm for detailed physical architectures [3]

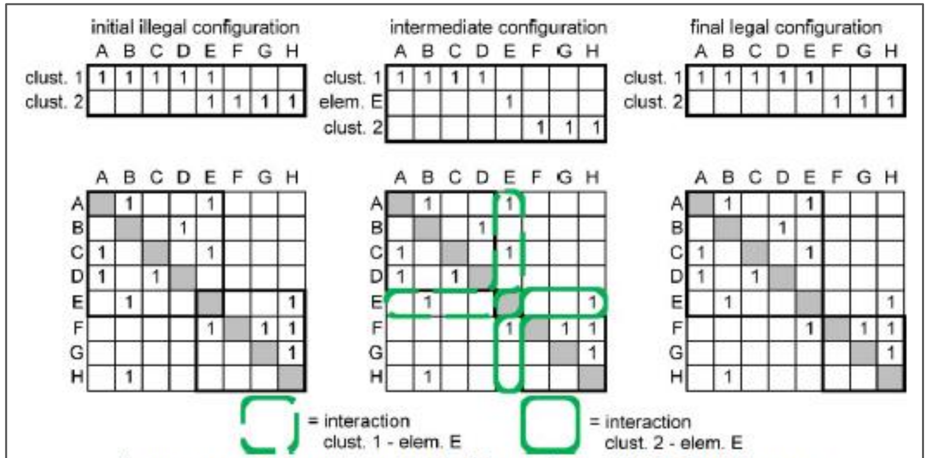
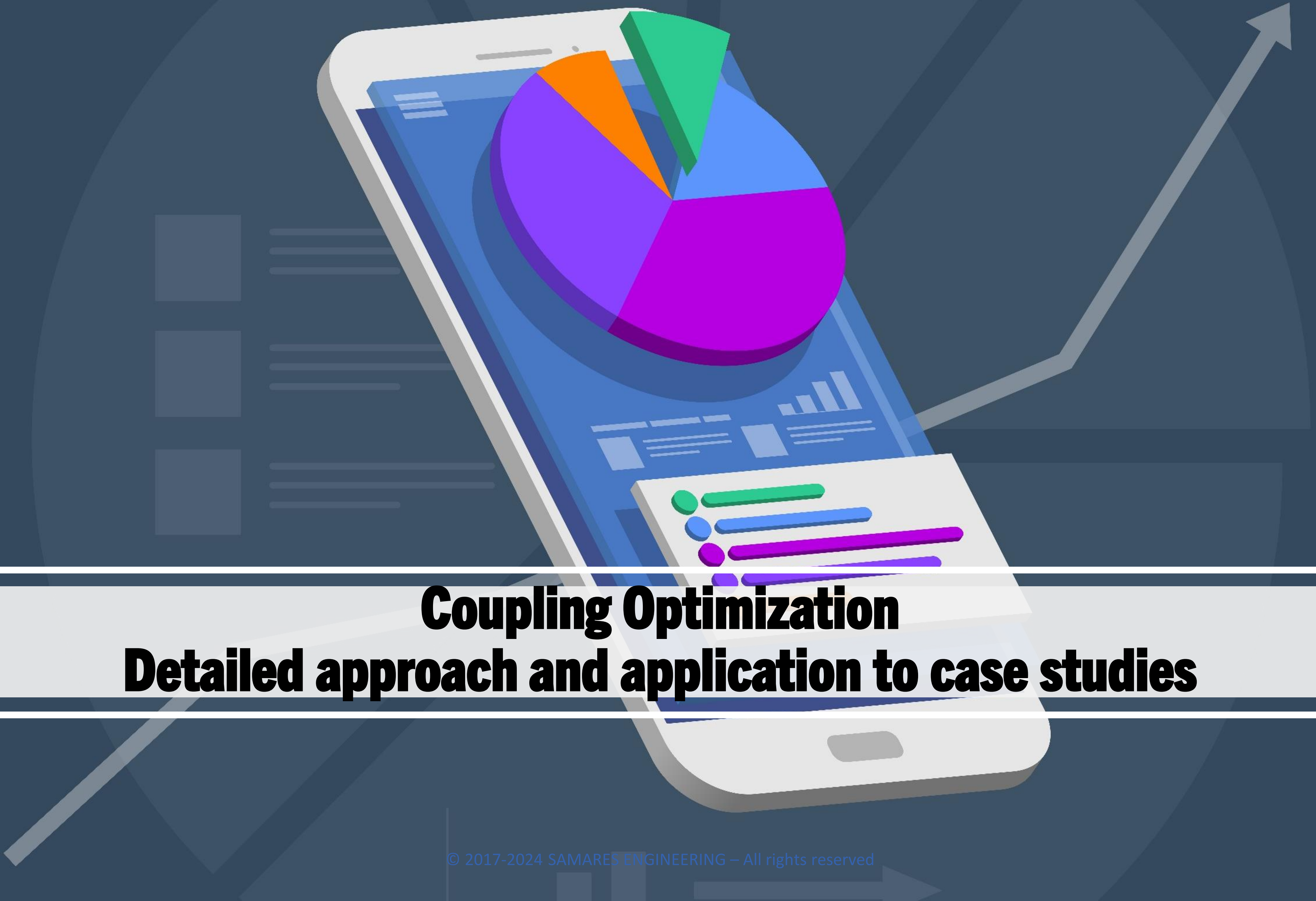


Figure 5. Heuristic attempts to find best cluster match for element E

Focus on optimization and fast algorithm on complex physical architecture

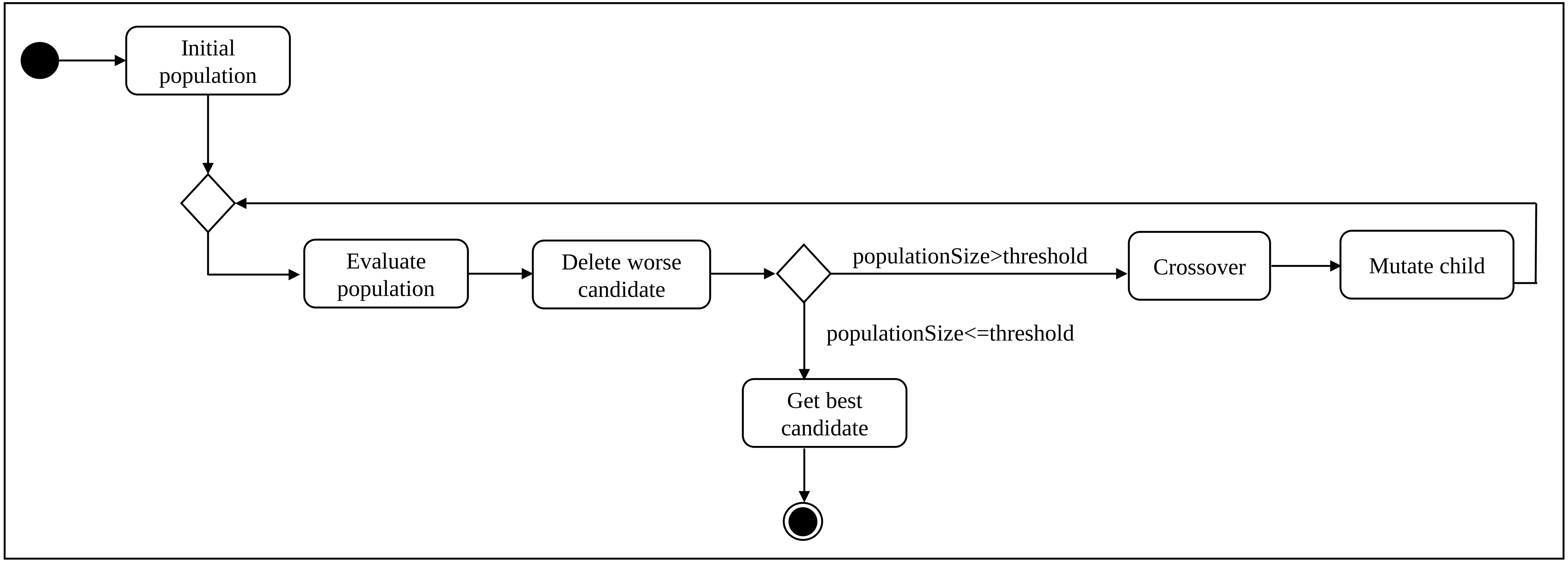


Coupling Optimization
Detailed approach and application to case studies

Approach Proposed

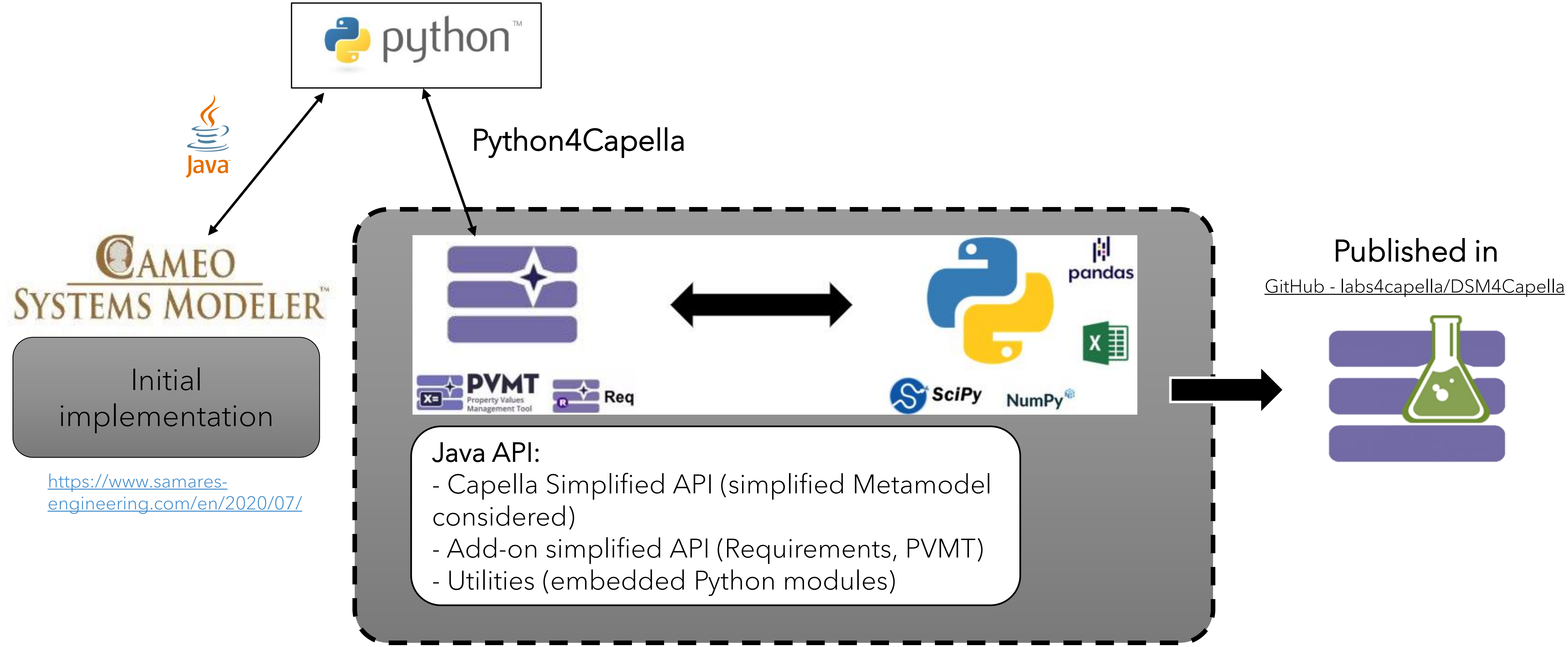
Use of DSM principle and apply the permutation principle to the Genetic algorithm

Genetic algorithms, aim to explore the solution space of a given problem to meet predefined criteria



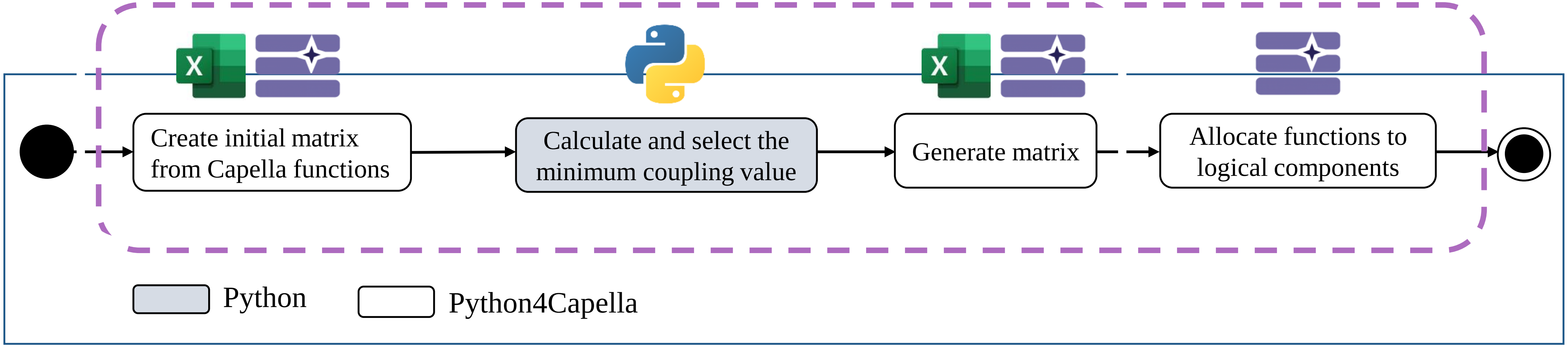
Genetic algorithm processes

Implementation



Implementation

Algorithm structure and adaptation to Capella

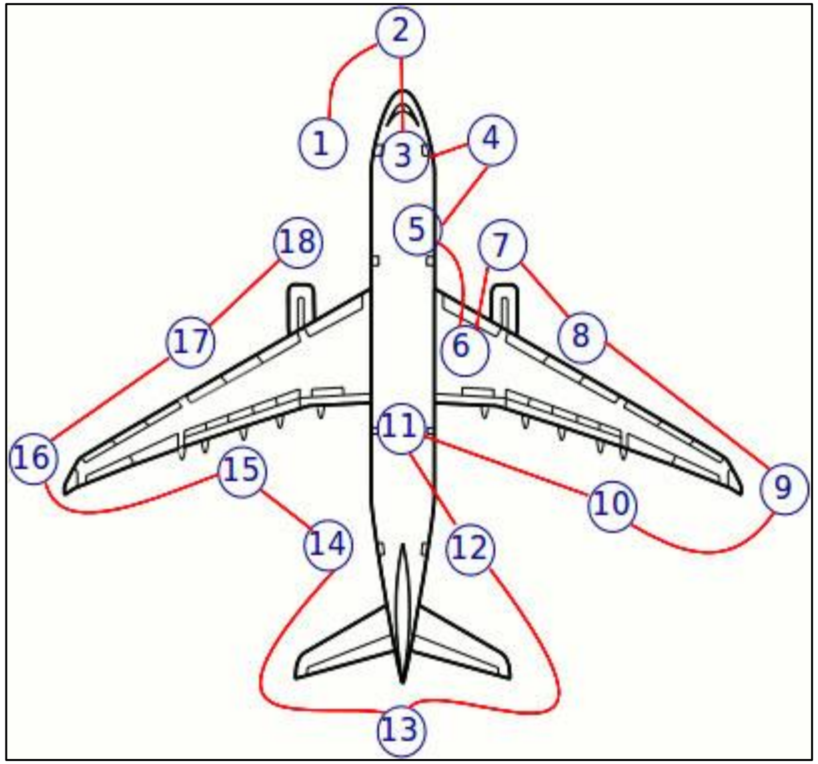


Analysis + Optimization

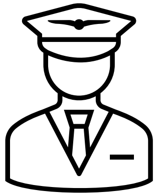
Implementation – 1st case study

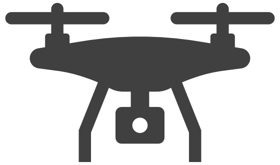
AIDA use case

The AIDA system is a Remotely Piloted Aircraft System (RPAS) that it is composed of a quadcopter drone, which performs an autonomous inspection around the aircraft before take-off [5].



Source: [5]

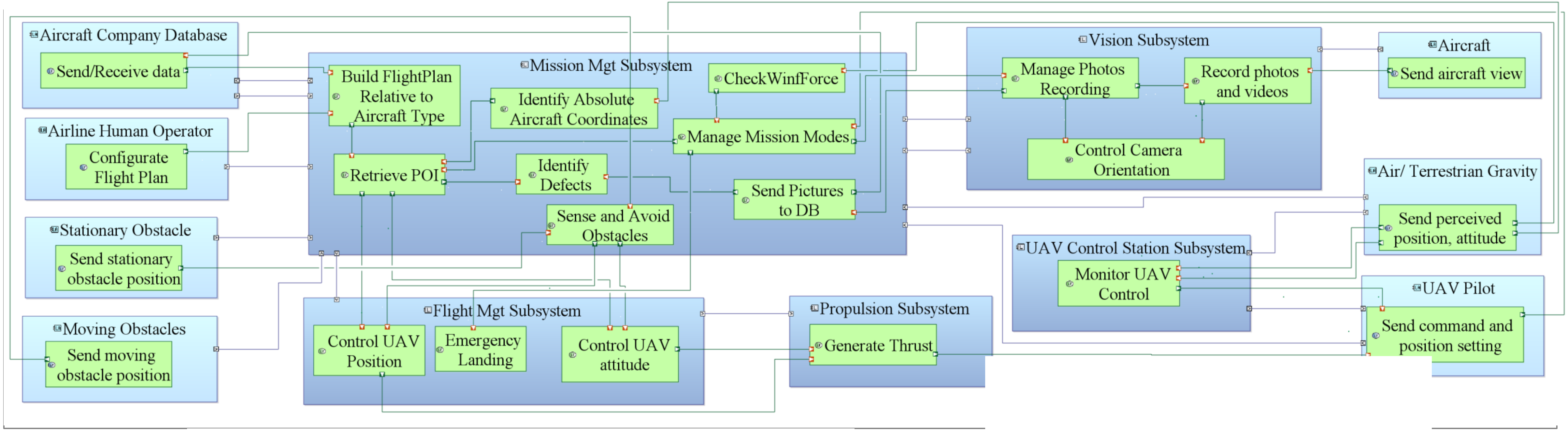




Implementation

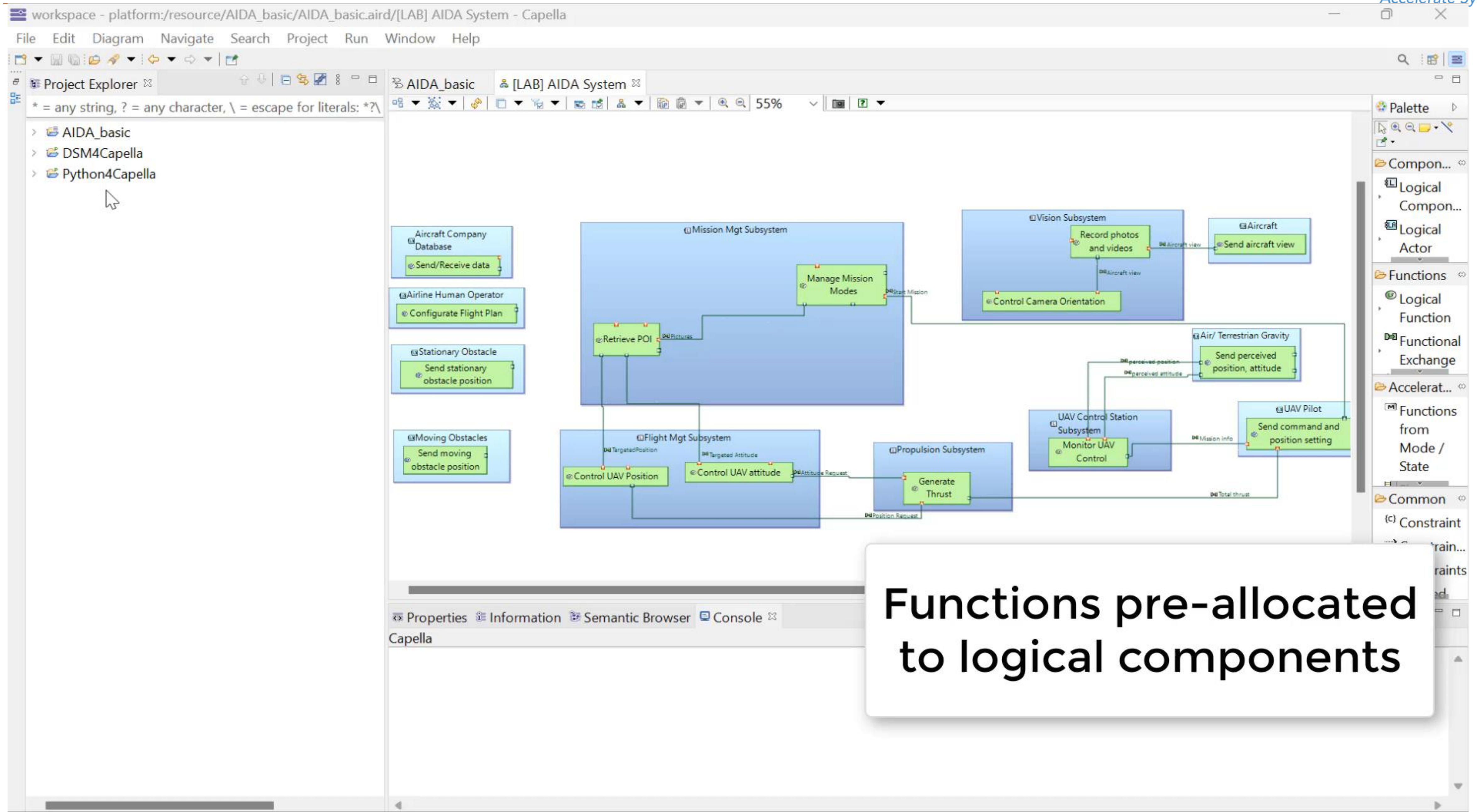
AIDA use case - CAPELLA

Logical architecture before and after algorithm execution



Logical architecture after algorithm execution

Demonstration

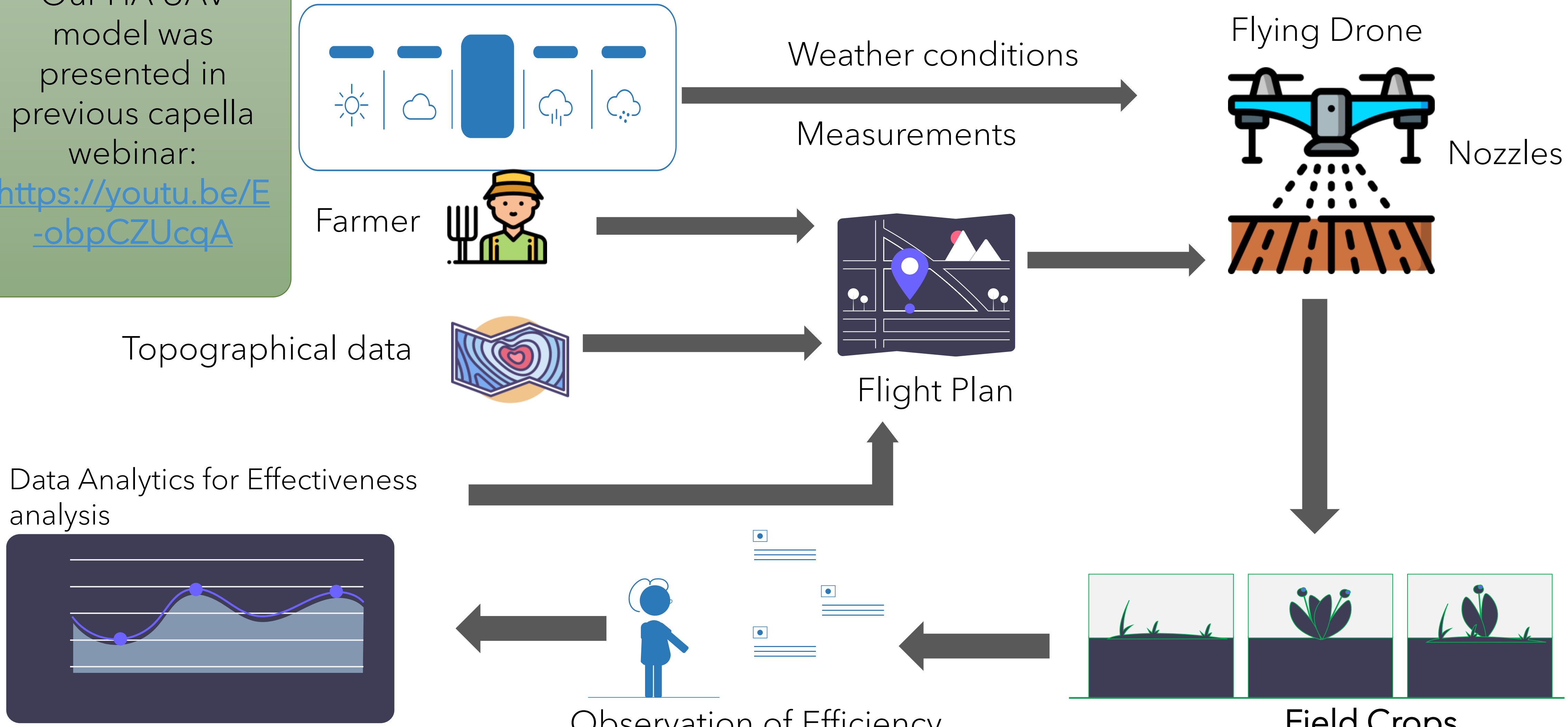


Functions pre-allocated to logical components

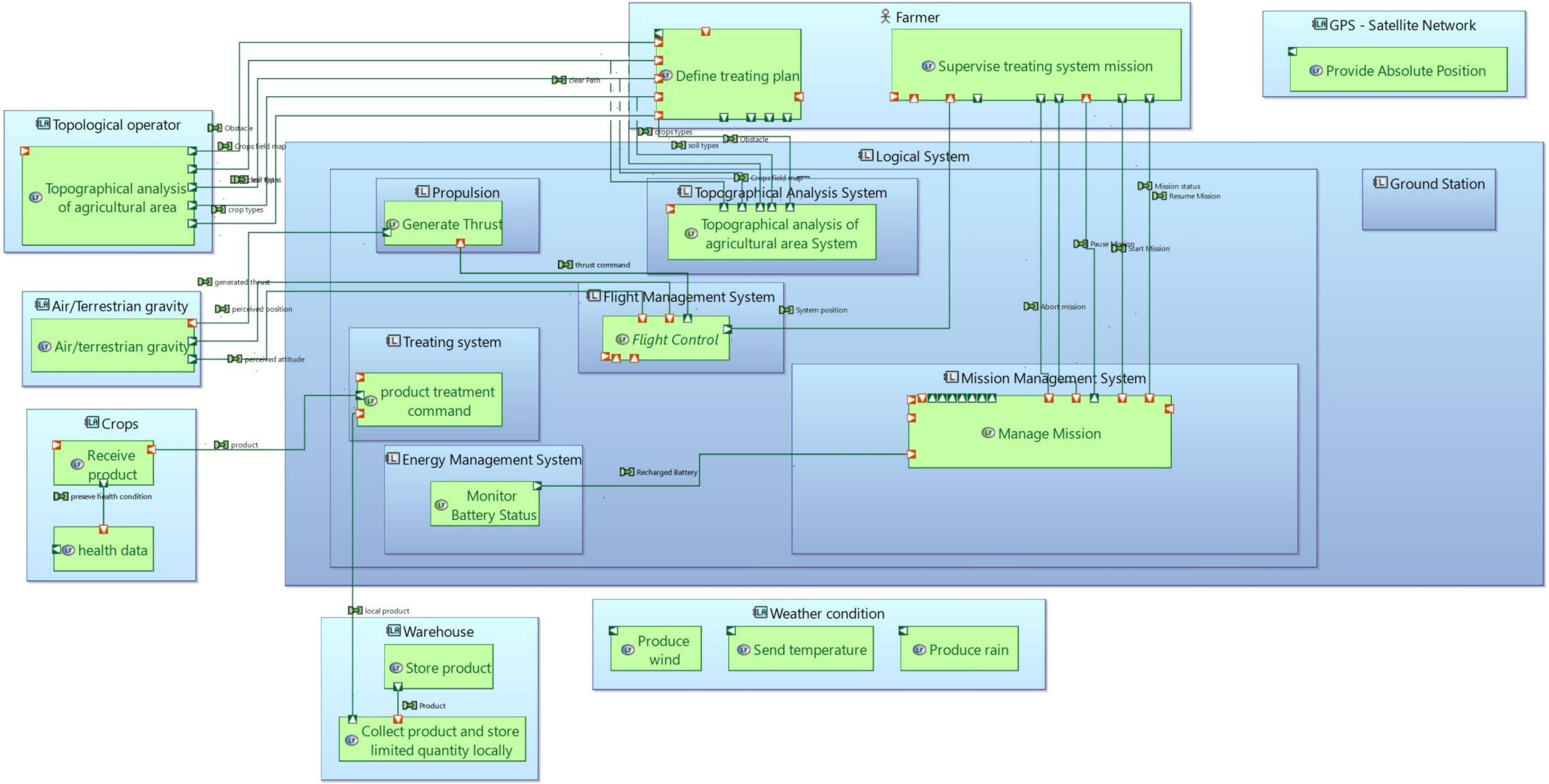
Implementation –2nd case study

Health Agriculture Unmanned Aircraft Vehicle (HAUAV) use case

Our HA UAV model was presented in previous capella webinar: https://youtu.be/E-obpCZUcqA

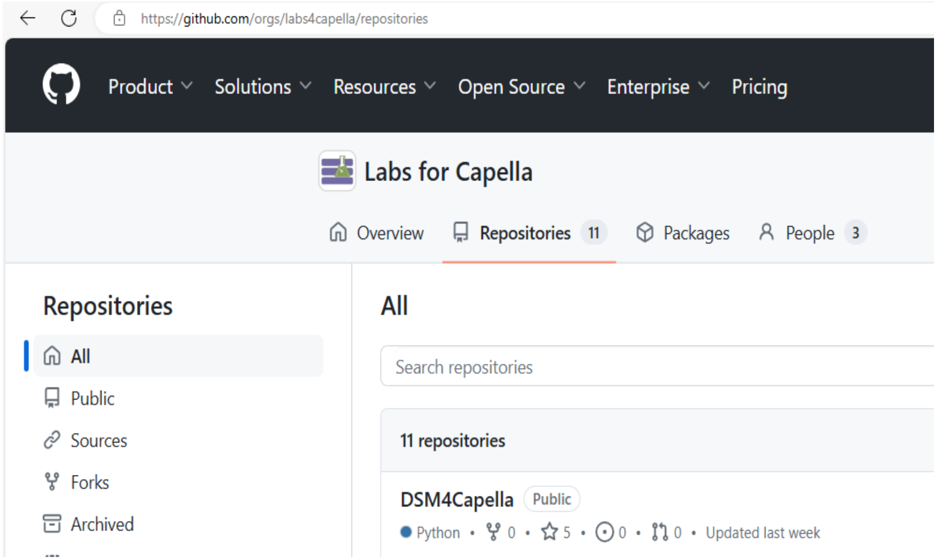


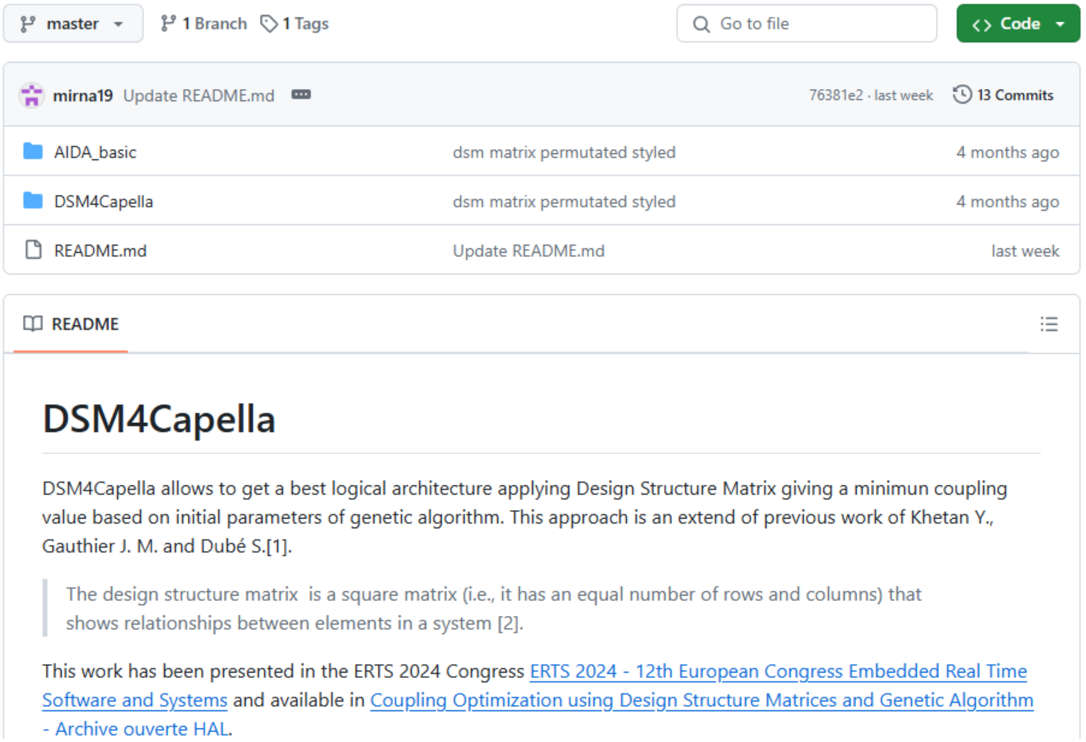
Demonstration - 2nd case study



Conclusion

- Proposed value on Functional and Logical Architectures integrated within MBSE tools
- Enables Decisions making regarding distribution of functions over logical architecture
- Publication **DSM4Capella** (**GitHub - labs4capella/DSM4Capella**) in Capella Community within Labs4Capella [6].





Future works

- Provide implementation of DSM generation within other MBSE Tools
- Extend the concept to introduce consideration of timing constraints and extend the functions and component exchanges with a time delay property and ensure as a constraint that Time budget allocated to the overall functional chain are fulfilled
- Optimize performances of the current algorithm
- Explore other algorithms than Genetic Algorithm and use optimization techniques proposed in related works to handle large matrices.
- Explore the possibility to generate alternatives of architectures in a same model and exhibit the associated properties (timing, performance, costs, …) of each. Then extend this with multi-dimensional optimization techniques.
- Enhance process with Layout facility

References

[0] Sebastien Dube, Mirna Ojeda, Jean-Marie Gauthier. Coupling Optimization using Design Structure Matrices and Genetic Algorithm. ERTS2024, SEE; 3AF, Jun 2024, Toulouse, France. ⟨hal-04632975⟩

[1] INCOSE, Systems Engineering Handbook - V5, 2023.

[2]. S. K. Salas Cordero, C. Fortin et R. Vingerhoeds, «Concurrent Conceptual Design Sequencing for MBSE of Complex Systems through Design Structure Matrices,» chez International Design Conference, 2020

[3] F. Borjesson et U. Sellgren, «Fast Hybrid Genetic Clustering Algorithm for Design Structure Matrix,» chez ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2013.

[4] Y. Khetan, J.-M. Gauthier et S. Dubé, «Part 5 – Coupling optimization of logical architecture using genetic algorithm,» June 2020. [En ligne]. Available: https://www.samares-engineering.com/en/2020/07/. [Accès le 28 05 2024].

[5] IRT St Exupery, «AIDA architecture,» [En ligne]. Available: https://sahara.irt-saintexupery.com/AIDA/AIDAArchitecture. [Accès le 28 05 2024]

[6] «Labs4Capella,» [En ligne]. Available: https://github.com/labs4capella/DSM4Capella/tree/master .[Accès le 23 05 2024].


Questions?