DELIVERABLE 3.3

AGILE 4.0 OPERATIONAL & TRADE-OFF PROCESS

Author(s): Francesco Torrigiani
WorkPackage N°: WP3
Due date of deliverable: 30.06.2022
Actual submission date: 19.06.2023
Document ID: AGILE4.0_Deliverable_3.3

Grant Agreement number: 815122
Project acronym: AGILE 4.0
Project title: Towards cyber-physical collaborative aircraft development
Start date of the project: 01/09/2019
Duration: 42 months

Project coordinator name & organisation:
Luca Boggero
DLR – System Architectures in Aeronautics | Aircraft Design and System Integration
Tel: +49 40 2489641-338
E-mail: luca.boggero@dlr.de

Project website address: www.agile4.eu

This project has received funding from the European Union's Horizon 2020 research and innovation framework programme under grant agreement No 815122

DOCUMENT INFORMATION

Document ID	AGILE4.0_Deliverable_3.3		
Version	03		
Version Date	16/06/2023		
Author	Francesco Torrigiani		
Dissemination level	Public		

APPROVALS

	Name	Company	Date	Visa
Coordinator	Luca Boggero	DLR	23/06/2023	
WP Leader	Thierry Lefebvre	ONERA	23/06/2023	

DOCUMENTS HISTORY

Version	Date	Modification	Authors
0.3	19.06.2023	Document finalization	F. Torrigiani

LIST OF AUTHORS

Full Name	Organisation

DISTRIBUTION LIST

Full Name	Organisation

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

GLOSSARY

Acronym	Signification
DSM	Design Structure Matrix
MBSE	Model Based Systems Engineering
MDAO	Multidisciplinary Design Analysis and Optimization
SE	Systems Engineering
SoI	System of Interest

1 EXECUTIVE SUMMARY

1.1 Introduction

The present deliverable is in the form of an OWL model [1]. This document is the companion report providing explanation and introduction to the model itself. The OWL model addresses the entire AGILE4.0 MBSE Development Process, whereas this report focuses only on the last step of the development process: System Design. The other steps are covered, in document form, by the deliverables D1.3 [2] and D2.3 [3] addressing System Definition and System Architecting respectively. The connections among the ontology elements belonging to the different phases of the development process are also highlighted in this report, providing an overall high-level view of the entire architectural framework. The OWL model is publicly available, through the AGILE 4.0 Zenodo page [4]:

- AGILE 4.0 Zenodo Community: https://zenodo.org/communities/agile4
- AGILE 4.0 MBSE Ontology DOI: 10.5281/zenodo.4671895

1.2 Brief description of the work performed and results achieved

The research activities that have brought to the final version of the present deliverable started from a brief literature survey on System Engineering methods or Model Based System Engineering frameworks covering the System Design phase of the development process.
Material from the different application cases concerning MDAO, Trade-Off and Design Scenario activities were provided to the entire consortium at M33 during a physical workshop. Feedback and comments from the different application case working groups were formalized and used to further refine the ontology model and process for the System Design phase of the development process.
A first version of the ontology concerning System Design and the connection between the ontology associated with the System Definition and System Architecting phases have been presented to the whole Consortium at M36 during a virtual meeting. The first version of the process concerning System design has been also presented to the whole Consortium in the same occasion.
After the first version of ontology and process was defined, guidelines concerning the Deliverables 6.5, 7.5 and 8.5 were also provided to the application case working groups.

1.3 Deviation from the original objectives

1.3.1 Description of the deviation

No technical deviation from the original objectives of the deliverable have been found. The only deviation observed is the delay in the writing of the deliverable itself. However, the technical activities documented in the deliverable have been conducted in due time, no negative impact on the connected tasks has been observed, see the following paragraph “Corrective Actions”.

1.3.2 Corrective actions

Material and guidelines affecting other deliverables and activities in the project that are part of this deliverable were provided to the consortium before the actual finalization of the deliverable itself in order to avoid delays in other interconnected tasks.

2 INTRODUCTION

2.1 Aim of the deliverable and organization of the report

The Deliverable D3.3 is presented in the form of an OWL model [4], which represents the conceptual elements involved in the design of the different Application Cases' system of interest. Among several other elements the key concepts of *Parameter*, *Parameter Role*, *Design Problem*, *Design Scenario* and *Trade-off* are introduced and integrated in the development process.
The Application Case Design represents the last step of the Model Based System Engineering (MBSE) process being set up within the AGILE 4.0 project for the development of complex systems of interest (e.g. aircraft). The MBSE process also includes System Definition and System Architecting which were addressed in Deliverable D1.3 [2] and Deliverable D2.3 [3] respectively. This process is schematized in Figure 1, based on [5].

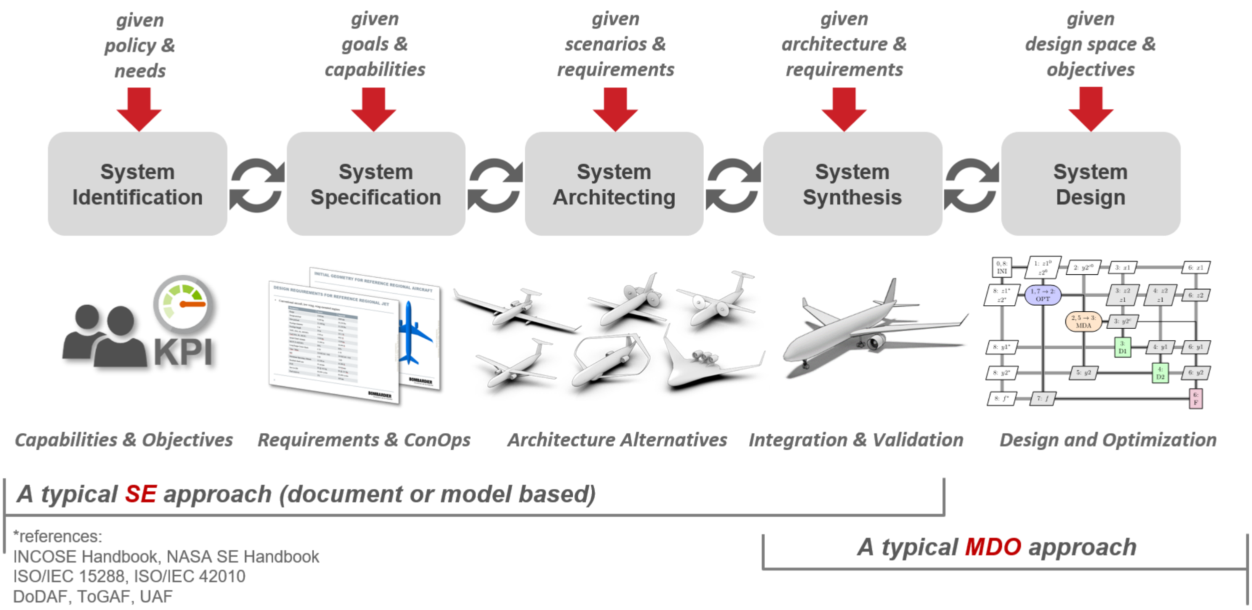

Figure 1: Schema of the AGILE4.0 Development process: a bridge between upstream architecting phase (e.g. a MBSE approach) and the downstream product design phase (e.g. a MDAO process).

The process of Figure 1 aims at designing and optimizing aeronautical systems (i.e. system synthesis and system design steps), starting from upstream system development activities. The first step aims at defining the systems' required capabilities and objectives. System stakeholders are identified, and their expected needs about the system are collected. Then, stakeholder needs are evaluated through operational scenarios and transformed into system requirements during the system specification step. System architecting aims at defining all the possible components that should be part of the system architectures in order to provide all the system functions as demanded by the functional requirements. The present deliverable concerns the final step of the development process, the system design phase.
The development process can also be divided in 2 main blocks[1]: the first one concerning System Identification, System Specification and System Architecting steps which are covered by a typical System Engineering (SE) either model- or document-based approach, and the second one consisting of the System Design step which is the considered in a typical Multi-Disciplinary Analysis and Optimization (MDAO) approach. One of the focus of the activities described in this deliverable is also the identification and formalization of the connecting element between the upstream SE part of the development with the downstream MDAO part of the

[1] The System Synthesis step can be either part of both SE and MDAO approaches

development process. Subsection **Error! Reference source not found.** present a brief literature review on the topic.

In this deliverable, the AGILE 4.0 MBSE Architectural Framework is introduced in Section 3, then the 3 elements of the architectural framework: ontology, viewpoints, and process are presented in Sections 4, 5 and **Error! Reference source not found.** respectively. Finally, Section **Error! Reference source not found.** presents some overall visualization capabilities covering the entire database generated during the AGILE4.0 development process.

2.2 Short Literature Review on MBSE-MDAO Connection

Many of the studies on SE and MBSE focus on the initial phases of the development process: stakeholders and needs identification, requirement specification, system architecting and synthesis. The last step of the development process, the system design phase, is often considered as a black-box or in a simplified manner lacking the formalization and the level of details necessary to effectively integrate and deploy MDAO methodologies in the system development process. In recent years, pushed by the challenges outlined in the introduction, few works tried to establish a formal connection between MBSE and MDAO, and investigate the application to the development of complex products.

Reference [6] is a survey on MBSE and MDAO methods and tools in the context of the French project Concorde. One of the project's main goals is the population of part of the MDAO models directly from the MBSE framework and the application of the developed methodologies to a UAV case study. In the survey, both MBSE and MDAO approaches are described, including an historical prospective and comments on the respective acceptance in the design engineering community. In the survey, it is also observed that one of most common MDO graphical representation, the Design Structure Matrix (DSM) [7] or the extended version proposed in [8], is derived from the N2 coupling matrix [9] which is a standard System Engineering approach for the analysis of interfaces in the logical architecture.

In [5], Ciampa et al. propose an MBSE architectural framework composed by ontological concepts and perspectives driving the development of MDAO system models. The paper introduces a distinction between the "complex system" under development (e.g. an aircraft) and the "design system" (e.g. the set of MDO tools used to design the system under development), pointing out that the latter can also be the System of Interest (SoI) for a development process. The reference focuses on the formalization of the MDO system model and therefore, from the product development point of view, concerns only the last part of the product development process, the system design phase. Although the SoI considered in this work is the "complex system" (e.g. an aircraft) and therefore different from the one considered in by Ciampa, reference [1] provides the theoretical foundation of the MBSE approach followed also in the AGILE 4.0 project and therefore in the current paper.

Reference [10] presents a link between MBSE and MDAO based on re-usage of already existing models generated in the upstream phases of the development process, in order to minimize the effort for the integration of MDAO methods in an MBSE framework. The proposed approach is applied to the design of an automotive coolant pump. Although the overall idea supporting the implemented methods is general enough to by applied to any engineered system, the approach relies on several manual operations and no formalization is provided to support the integration of the typical numerical models used in aeronautical MDO processes. Accordingly, the disciplines integrated in the MDAO example consist only of analytical equations.

In the framework of the AGILE 4.0 project, the connection between MDO and system architecting is investigated in [11]. The link between the two phases is established assigning quantity of interest to the system logical architecture and then mapping them to the product data schema which is used in the MDO workflow formulation. The focus on the paper is on implementation aspects and strongly relies on technologies available in the MDAO-MBSE framework under development in the AGILE 4.0 project.

3 AGILE 4.0 MBSE ARCHITECTURAL FRAMEWORK

One of the goals of the AGILE 4.0 project is the definition of a new MBSE Architectural Framework. The ISO/IEC/IEEE 42010 standard defines an architectural framework as a set of:

conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders [12]

In other words, an architectural framework is a guideline that can be used to represent system architectures, where an architecture is a formal description of a system, of its behaviour and of the relationships among all the entities composing the system. The MBSE Architectural Framework developed in AGILE 4.0 is structured in four layers, as shown in Figure 2.

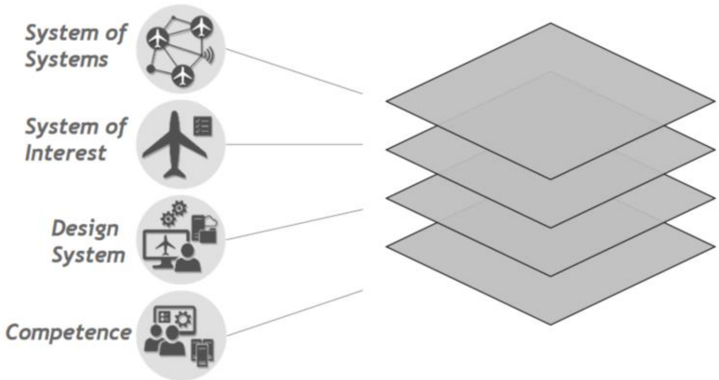

Figure 2: Four-layer structure of the MBSE Architectural Framework addressed in AGILE 4.0 project.

One layer focuses on the System of Interest, which is the system being designed and optimized. The System of Interest can be for example an aircraft, a subsystem (e.g. ATA 24 – Electrical System) or a component e.g. an electric generator), but also an enabling system, like a Supply Chain or a production system. The MBSE approach schematized in Figure 1 is employed for the development of the System of Interest. In AGILE 4.0 project, seven Systems of Interest (SoI) are developed in WP6-7-8.
The System of Interest can be developed through a Design System, which is represented by a different layer. This layer encompasses all the Systems Engineering and MDAO processes required to develop the System of Interest. Again, an MBSE approach can be employed for the definition, architecting and development of this layer.
The development of a System of Interest needs several competences, which can be for example formalized through disciplinary modules. Therefore, the lower layer Competences is part of the MBSE Architectural Framework.
As mentioned before, the ultimate aim of the MBSE Architectural Framework is the development of Systems of Interest. However, the same framework can be employed for the development of multiple systems operating together according to the concept of System of Systems, which defines the fourth layer, the highest one.
The present deliverable focuses on the Design System layer, although the proposed guideline is used to represent the architectures of a System of Interest.

Several architectural frameworks are available in literature (e.g. Zachman's Framework [13], DoDAF [14], TOGAF [15] and UAF [16]). They are all characterized by the following elements:

- *Ontology*: definition of the concepts composing the architecture and their relationships.
- *Viewpoint*: convention for the construction, interpretation and use of architectures from the perspective of specific system concerns.
- *Process*: logical sequence of tasks performed to achieve a particular objective.

The focus of this deliverable is on the last step of the Systems Engineering approach being adopted in the AGILE 4.0 project, namely System Design. It should be finally noted that an MBSE architectural framework is employed in the frame of the present deliverable to address the definition of Systems Of Interest. However, an MBSE architectural framework can be similarly adopted to address all the layers depicted in Figure 2, therefore targeting also the development of the Design System.

4 ONTOLOGY

The ontology depicted in Figure 3 defines the main conceptual elements involved in the System Design phase and the relation among them. The conceptual element of the ontology concerning the System Identification and Specification phases are also included in the AGILE4.0 Architectural Framework Ontology, which is built using Protégé [17] and is published as open access. Therefore, it can be freely downloaded from the Community page of the project on the Zenodo website [4] and re-used by any user inside and outside the project

Consortium. The available files represent the meta-models, rendered by OWL, supporting the development of any complex system in any domain.

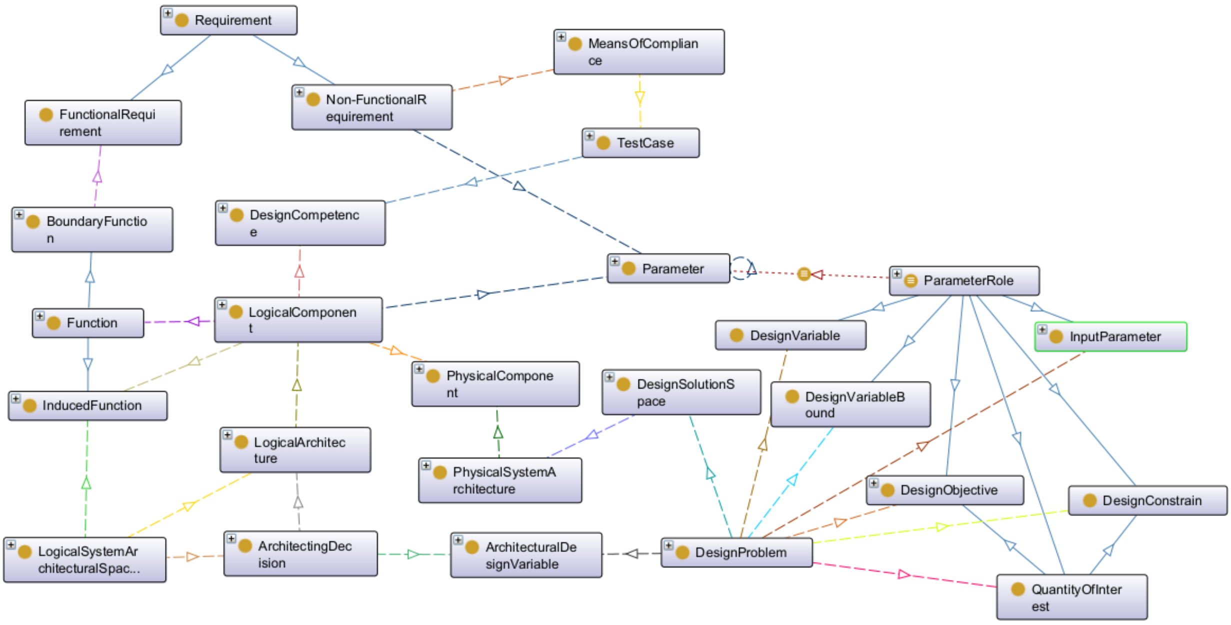

Figure 3: Key elements of AGILE4.0 Architectural Framework Ontology, as represented by the "OntoGraf" plugin of Protégé [17]

In this deliverable, the AGILE4.0 Architectural Framework Ontology is often represented as a SysML Internal Block Diagram (IBD) in order to facilitate the reading, like in Figure 4. However, the complete ontology is available on Zenodo.

In Figure 4, four main cluster of conceptual elements can be identified in the ontology of the system design phase. Conceptually the central entity of the ontology is the Design Problem and the associated cluster of elements. The *Design Problem* generates a *Design Solution Space* which as a different dimension according to the chosen *Design Driver*. For instance, if a single objective optimizer is used as the driver of the design process, then the *Design Solution Space* consists of a single system configuration, i.e. a single *Design Solution Vector*. Instead the use of a *Design of Experiment* (DOE) driver will lead to the definition of multiple *Design Solution Vectors*. After the *Design Solution Space* definition, the chosen *Decision Maker* strategy compares the different design solutions and chooses a single *Design Solution Vector*.
The *Design Problem* is composed by two main entities: *Design Workflow* and *Parameters* with the associated roles. The *Design Workflow* cluster is composed by *Workflow Components*, which can be either a *Design Driver*, like DOE, single or multi-objective optimization algorithm; or *Design Competence*, meaning a disciplinary design competence like aerodynamic performance analysis or structural sizing. The Design Competence element provides one of the connections between the system design phase and two upstream phases: the system architecting and the system specifications. *Design Competences* compose on one hand the *Test Cases* which are used to validate non-functional requirements, and on the other hand the set of *Available Design Competences*.

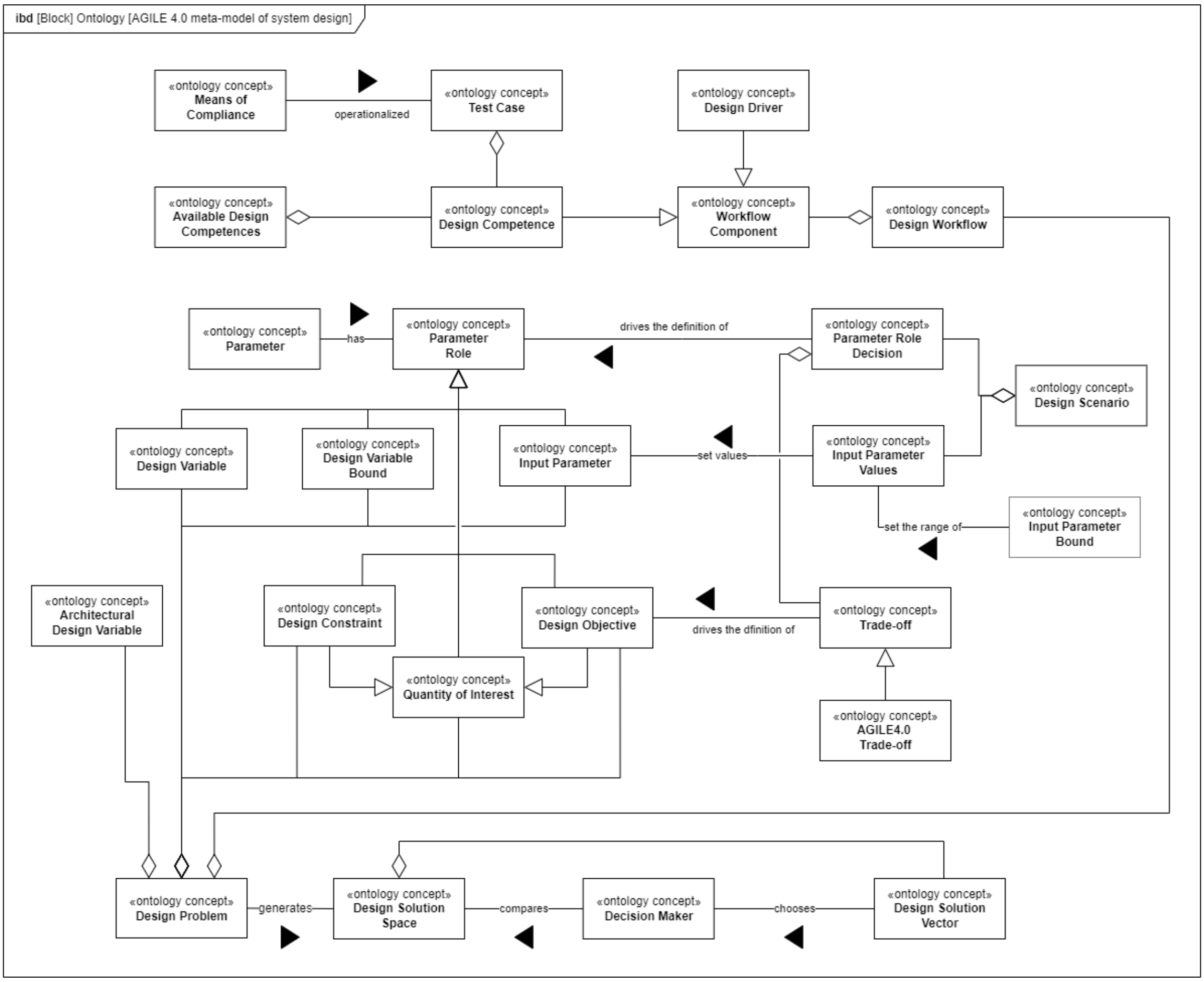

Figure 4: The AGILE4.0 Architectural Framework Ontology as SysML Internal Block Diagram representing key concepts and their relationships for the system design phase.

The third cluster originates from the *Parameter* element which has a *Parameter Role*. Following the nomenclature proposed in [17], six different roles are defined:

- *Design variables* are the parameters defining the Design Solution Vector and are directly defined by the Design Driver.
- *Design Variable Bounds* are the upper and lower limits of the Design Variables and together with the Design Constraints define the design space of which the Design Solution Space is a subset. They are constant during the execution of the design process.
- *Quantity of Interests* are a subset of the output parameters generated by the different Design Competences, and need to be monitored during the execution of the design process.
- *Design constraint* parameters are a sub-class of the Quantity of Interest parameters and, together with Design Objectives, are inputs of the Design Driver. Together with the Design Variable Bounds define the design space.
- *Design Objective* parameters are Quantity of Interest of which the limit values (either minimum or maximum depending on the design problem definition) is sought within the design space.
- *Input Parameters* are necessary for the execution of some of the Design Competences but are not provided as output neither by the Design Driver nor by other Design Competences, therefore they need to be provided initially and are constant during the execution of the design process.

The remaining conceptual elements belong to the *Design Scenario* cluster. In the proposed ontology, the *Design Scenario* is composed by the *Parameter Role Decision* and the *Input Parameters Values*. As already mentioned, the values of the *Input Parameters* are not defined during the design process and need to be provided

externally. For instance, in a classical aircraft design setup the fuel price is a typical Input Parameters, it influences the final design solution (e.g. by minimizing for the Direct Operative Cost which are highly affected by the fuel price), it is not computed by any design competence, and needs to be provided as input of the design process. The designer sets the fuel price value or a set of values according to the required scenario.
The assignment of parameter roles is represented by a dedicated element within the Design Scenario concept, the Parameter Role Decision. In fact, many different combinations of roles are usually compatible with the given requirements and each of them leads to a different Design Solution Space. Among the different parameter role decisions, the ones concerning the design objectives are associated with the Trade-off element. In the proposed ontology the following definition of trade-off is used: "*a balancing of factors all of which are not attainable at the same time*". The generic *Trade-off* concept is further specialized in the *AGILE 4.0 Trade-off* element, which poses and additional condition on the trade-off factors: they must belong to different domains of the aircraft development process (e.g. design, certification or manufacturing domain).

4.1 Connections among the Ontologies of the Entire Aircraft Development Process

Figure 5 highlights the connections among the ontologies of the different phases of the development process, namely the system specification, the system architecting and the system design. Note that the ontologies in Figure 5 are not complete since only the element involved in the connections are represented. The following connections are identified:

1. Independently from the roles, the system design Parameter element connects both system specification and system architecting with the system design. Parameters are derived directly from Non-Functional Requirements and also from architectural Logical Components, which in turn are generated from Functional Requirements.
2. It is also observed that most of the times Input Parameters and Design Variables are associated to architectural Logical Components, whereas Quantities of Interest derive from Non-Functional Requirement.
3. The Design Competence element connects both system specification and system architecting with the system design. Design Competences compose Test Cases which derive from Non-Functional Requirements, and are also associated with the system architecting Logical Component element.
4. In addition to the Design Variables derived from the Parameters element, the Design Problem also includes Architectural Design Variables which are associated with Architecting Decisions and are typically categorical variables.
5. Finally, it is observed that the design solution is in fact a Physical System Architecture where all the Logical Componenta have been instantiated assigning a value to the associated parameters and all the Architecting Decision have also been taken by the Design Driver.

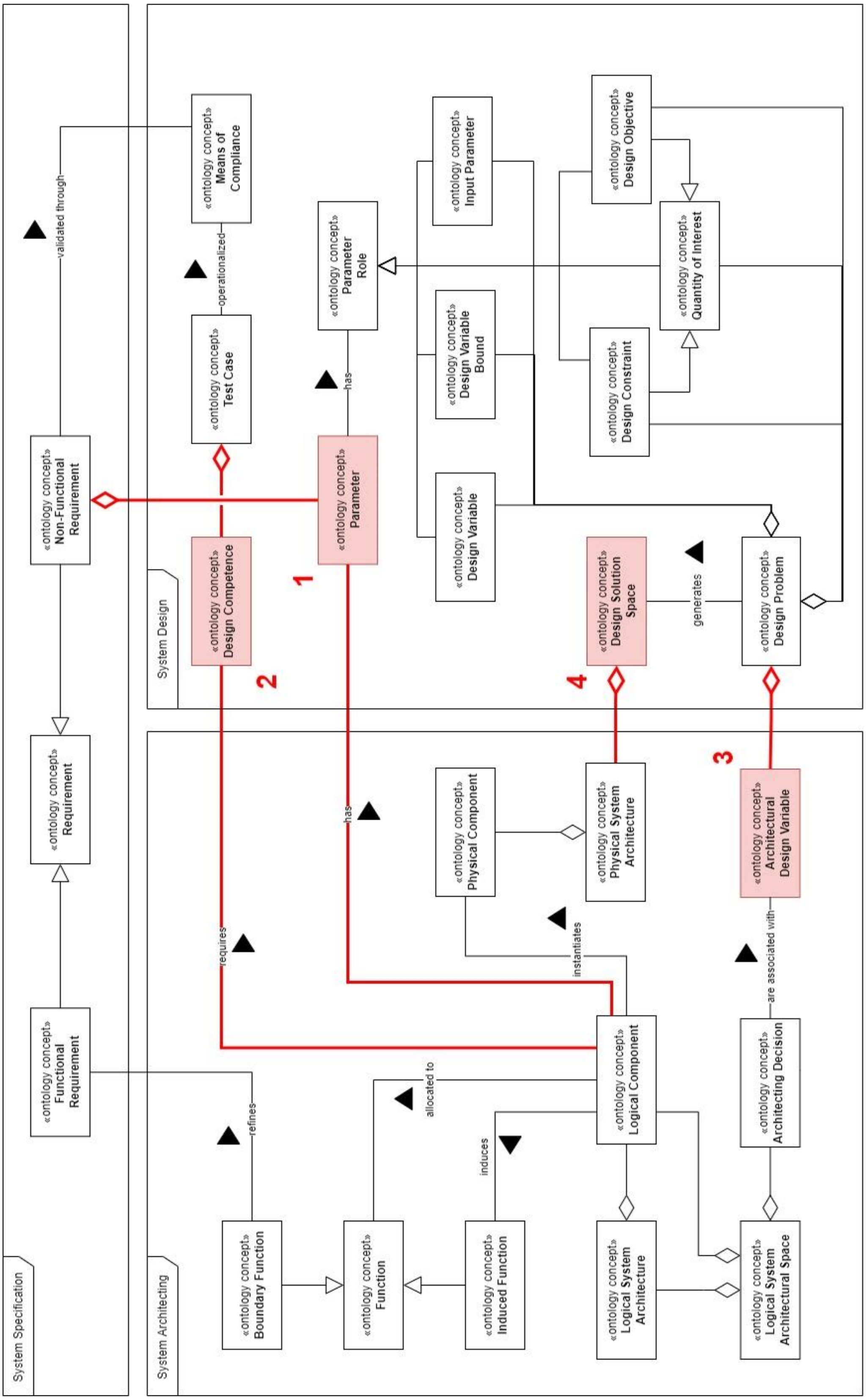

Figure 5: Connections among the ontologies of the System Specifications, System Architecting and System Design phases

5 VIEWPOINTS

Seven different novel viewpoints are proposed for the representation of the system design phase, each one of them focuses on specific aspects of the system. The SysML Package Diagram depicted in Figure 6 collects all the viewpoints together with the associated attributes.

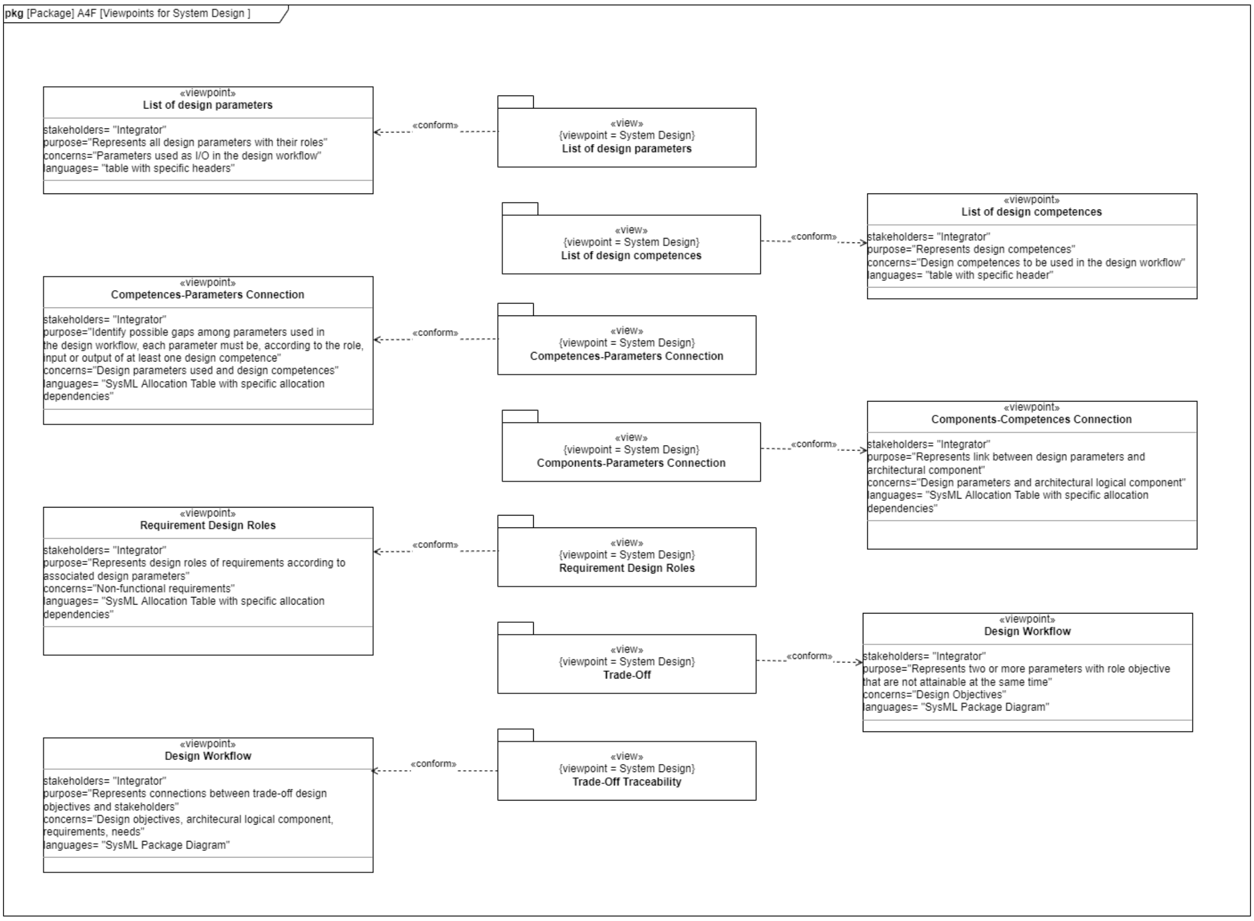

Figure 6: Viewpoints associated to the system design phase as proposed by the AGILE4.0 Architectural Framework

The following subsections present the modelling guidelines prescribed by the viewpoints recommended in the AGILE4.0 architectural framework. When necessary, a modified profile derived from SysML has been created to represent specific element of the system design viewpoints.

5.1 Viewpoints "List of design parameters" and "List of design competences"

Two different viewpoints are proposed to represent the model of design parameters and design competences. The viewpoints "List of design parameters" provide guidelines to represent parameters involved in the design process of the system of interest with the specification of their role in the design workflow. The language for the "List of design parameters" is tabular with specified headers.

Table 1: Template for viewpoint "List of design parameters", template table is filled with an example from AGILE4.0 AC5

Parameter Name	*Parameter Role*	*Parameter Description*	*Value type*
WST	Design Variable	Wing Spar thickness	Continuous
OEM	Design Objective	Operative Empty Mass	Continuous

string	*string*	*string*	*[Continuous; Discrete]*

The viewpoints “List of design competence” provide guidelines to represent design competences, either for analysis or optimization, involved in the design process of the system of interest with the specification of the ownership of the tool. The language for the “List of design competence” is tabular with specified headers.

Table 2: Template for viewpoint “List of design competences”, template table is filled with an example from AGILE4.0 AC5

Design Competence	*Function Description*	*Version*	*Input Description*	*Output Description*	*Level of Fidelity*	*Owner*	*Operator*
Descartes	Structural model generator	2.0	OML, structure layout, material	Structural GFEM ready for optimization	L2		
String	String	2-digit version identifier	String	String	[L0; L1; L2; L3]	String	string

5.2 Viewpoints “Competences-Parameters Connection”, “Components-Parameters Connection” and “Requirement Design Roles”

The two viewpoints “Competences-Parameters Connection” and “Components-Parameters Connection” aims at representing the link of the design parameters with design competences and architectural logical components.

A SysML Allocation table is used for both viewpoints with specific allocation dependencies. In addition, the “Competences-Parameters Connection” is used to identify gaps or inconsistency in the design workflow. In particular the following checks are performed:

- Input type of design parameters (namely Design Variables and Input Parameter) must be input of at least one design competence
- Input type of design parameters (namely Design Variables and Input Parameter) cannot be output of any design competence
- Output type of design parameters (namely Design Constraint, Design Objective, Quantity of Interest) must be output of at least one design competence

To facilitate the above checks two different cell type can be used for input or output parameters of the design competence.

Components	QOIs	AESA	ASTRID	CAESIOM	Descartes	FAEDO	HFL	LIE	LIE-OBS Surrogate	Lagrange Analysis	Lagrange Sizing	MUST	SMA	TRA
		Tools												
3D EM Simulation	ATL							✓						
Aerostructural Analysis	Flutter Speed Margin	✓												
LIE Assessment	ETDL-ATL Margin							✓						
OBS Design Specifications Reading	ETDL							✓						
Surrogate of EM simulation	ATL							✓						
Thermal Risk Assessment (DO160G - SAE AIR1168/6A)	Thermal Risk Scores													✓
	Buckling Reserve Factors									✓	✓			
	Reserve Factors for Damage Tolerance									✓	✓			
Certification guidelines AMC 20-136														
Enviromental Thermal Analysis														
OBS Layout Modelling														
Structural design														

Figure 7: Example of “Competences-Parameters Connection” table obtained with one of the MBSE technologies developed in AGILE4.0. The gaps are automatically identified and highlighted in red.

The purpose of the "Requirement Design Role" is to represent the design roles of requirements according to associated design parameters. In this viewpoint only non-functional requirements are considered since are the only one with that include a parameter according to the specific pattern. In this table the role of the parameter is specified in the table cells.

5.3 Viewpoints, "Trade-Off" and "Trade-Off Traceability"

The last two viewpoints addressed by the system design phase of the AGILE 4.0 architectural framework prescribe guidelines for the representation of trade-off and how the trade-off can be traced back to the stakeholders.
Trade-off is a key element of the AGILE 4.0 project and the following definition is used:

Trade-off is a balancing of factors all of which are not attainable at the same time

In agreement with the above definition the "Trade-Off" view consists of a SysML package diagrams presenting the trade-off element composed by a textual description, the connected design objectives and the associated stakeholders. In addition, the trade-off element is connected to the design parameters facilitating the identification of the design parameters involved in the trade-off.

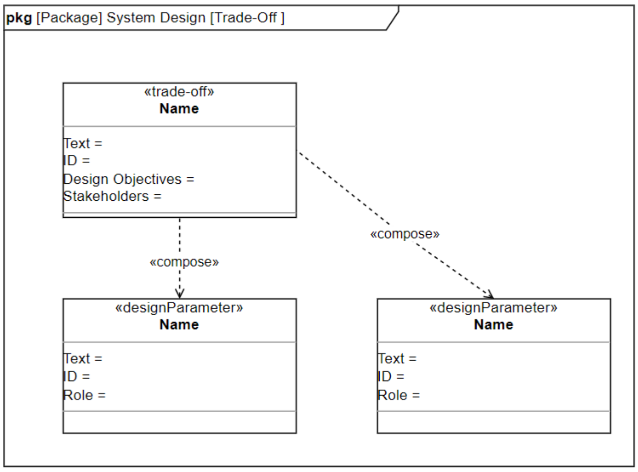

Figure 8: template for viewpoint "Trade-Off"

The "Trade-Off Traceability" view is also a SysML package diagram which further details the information summarized in the previous view. Here the parameter composing the trade-off is further connected to either an architectural logical component or directly to a non-functional requirement. In case the design parameter is connected to the architectural logical component, the latter can be traced back to a functional requirement. Finally, either functional or non-functional requirements are connected to a need which in turns is associated to a stakeholder.
The full chain of data connection, from trade-off to stakeholders is automatically established thanks to the different AGILE4.0 technologies. The next section provides further details and examples of the technologies for automatic visualization of the AGILE4.0 database.

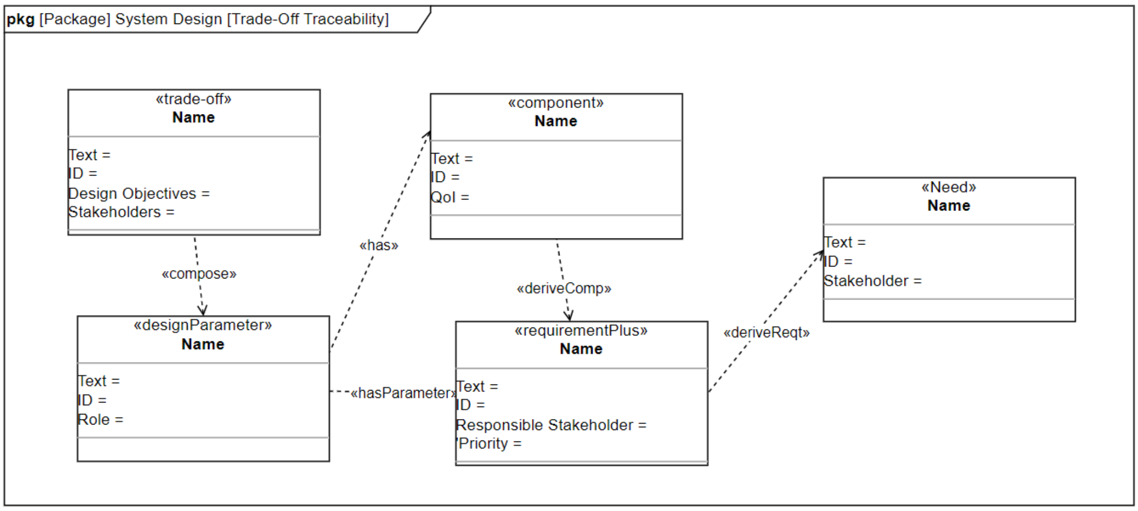

Figure 9: template for viewpoint "Trade-Off Traceability"

6 VISUALIZATION

The starting point for the visualization process is the definition of the graph data model representing the different types of nodes and relationships contained in the graph. The ontology described in the Section 4 can be directly considered as a graph data model. In fact, all the information contained in a graph can be translated without loss to an ontology. In particular:

- Ontology concepts are graph nodes
- Ontology relations are graph edges

Figure 10 represents the graph data model obtained from the AGILE 4.0 ontology as described in Section 4.

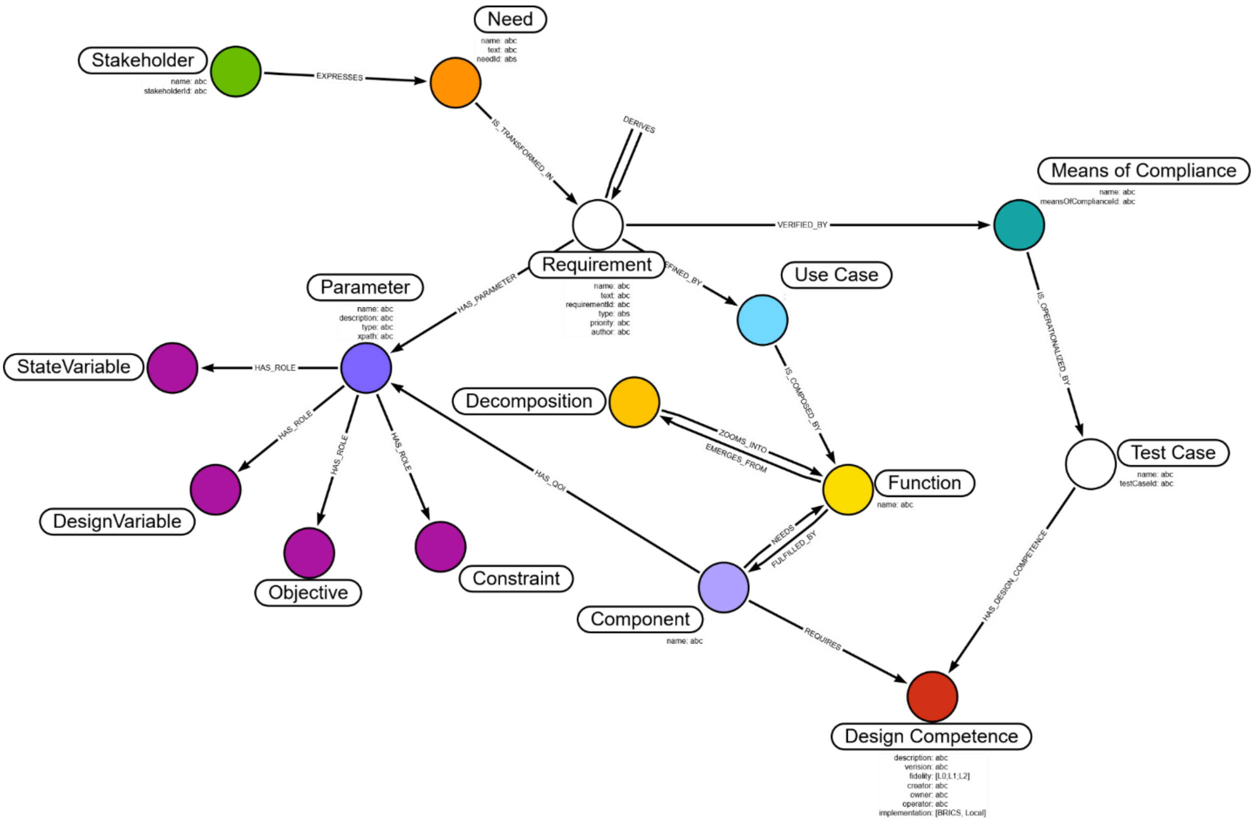

Figure 10: Graph data model obtained from AGILE4.0 ontology

6.1 Graph visualization with NEO4J

The well-established open-source NEO4J platform has been used for graph generation and management, together with ad-hoc python script to extract and pre-process AGILE4.0 data.

The picture below shows the graph created with the developed technology for two AGILE4.0 Application cases. Note that the graphs encompass the entire database from stakeholders to design parameters.

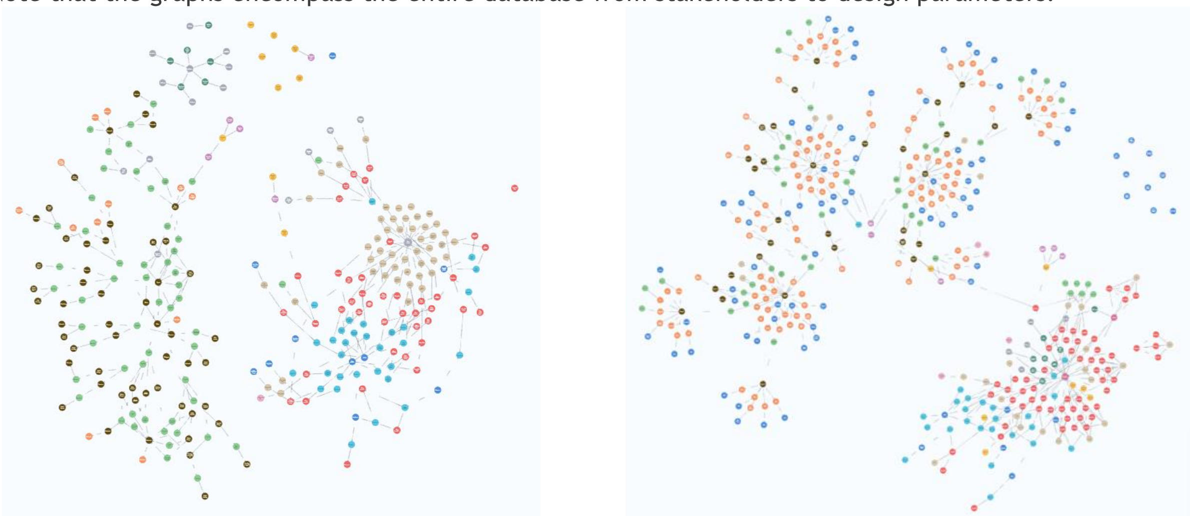

Figure 11: complete graph database for AGILE 4.0 AC1 (left) and AC7 (right)

Exploiting the native NEO4J query language, cypher, knowledge can be extracted from the generated database. For example, the entire graph can be traversed quickly to obtain:

- The influence of a specific stakeholders on the entire database: resulting in all the nodes that originates from that specific stakeholders. See an example in Figure 12
- The unique influence of a specific stakeholders: resulting in all the nodes that would disappear from the database if that stakeholders was removed from the database. This is a subset of the nodes obtained with the previous query. See an example in Figure 13

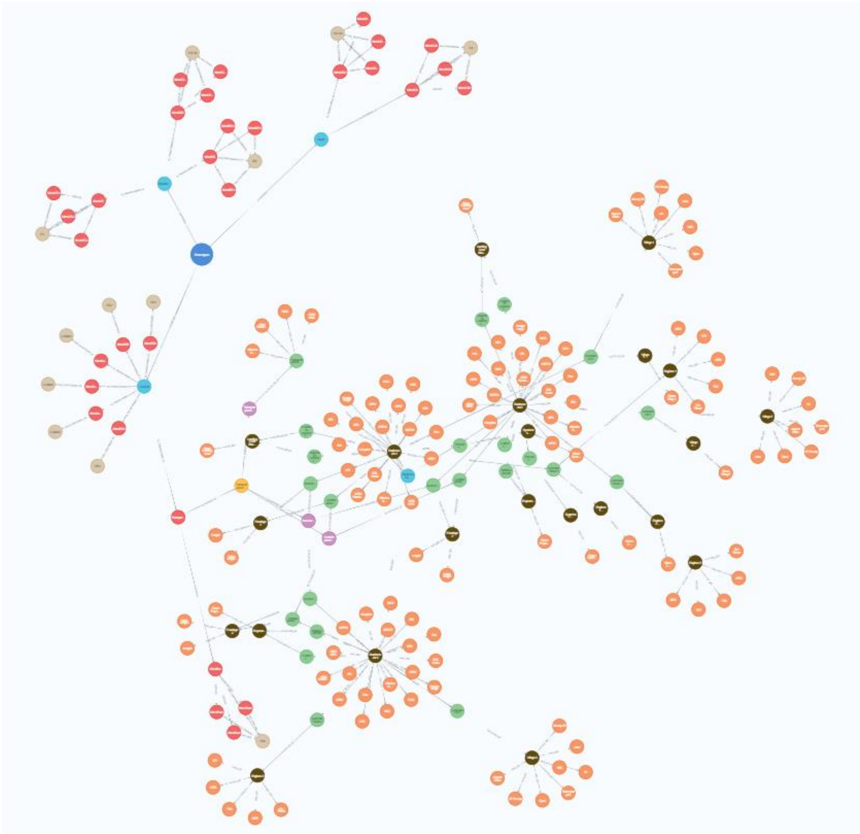

Figure 12: influence of stakeholder "Passenger" for the AGILE 4.0 AC7

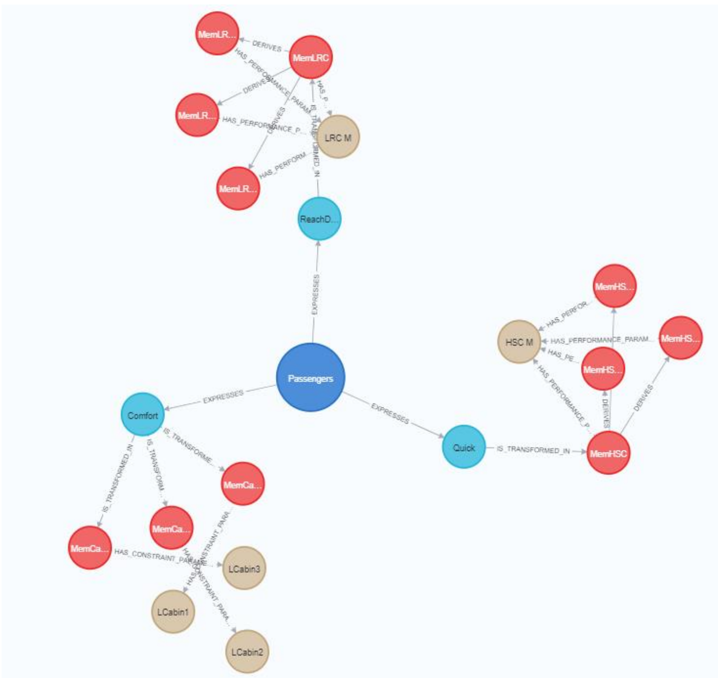

Figure 13: Unique influence of stakeholders "Passenger" for the AGILE4.0 AC7

7 CONCLUSION AND OUTLOOK

In this deliverable, the part of the AGILE4.0 MBSE Architectural Framework concerning System Design is presented in terms of Ontology and Viewpoints. In addition, the connection of the system design ontology with the rest of the AGILE 4.0 ontology is also highlighted.
This document is the companion report providing explanation and introduction to the model itself. The OWL model addresses the entire AGILE4.0 MBSE Development Process, whereas this report focuses only on the last step of the development process: System Design. The OWL model is publicly available, through the AGILE 4.0 Zenodo page [4]:

- AGILE 4.0 Zenodo Community: https://zenodo.org/communities/agile4
- AGILE 4.0 MBSE Ontology DOI: 10.5281/zenodo.4671895

The technical development described in this deliverable have been used in the project to support specifically the activities described in Deliverables D6.5, D7.5 and D8.5. Finally, the last section describes further visualization technology developed specifically to extract knowledge form the entire AGILE 4.0 database.

Future developments concern a refinement and extension of the ontology and viewpoints presented in the previous section. The level of details used in the ontology definition can be increased in other to allow a reasoner to directly extract knowledge form the ontology. The viewpoints can be extended to integrate some of the visualization example provided in the last section, in this case the automatic generation of SysML from the graph database needs to be developed.
Finally, the same process adopted to generate the graph database from the AGILE4.0 data can be adopted to populate the AGILE4.0 MBSE ontology, which is currently solely composed by class elements, with individuals in order to allow specific queries directly on the ontology.

8 REFERENCES

[1] W3C, “OWL 2 Web Ontology Language Primer,” 11 December 2012. [Online]. Available: https://www.w3.org/TR/owl2-primer/. [Accessed 8 February 2023].

[2] L. Boggero, “Process Definition A4F for Application Definition,” AGILE4.0 - Deliverable, 2021.

[3] L. Boggero, “A4F System Architecting Definition,” AGILE 4.0 - Deliverable, 2023.

[4] L. Boggero, P. D. Ciampa and J. Jepsen, “AGILE 4.0 MBSE Ontology, Zenodo,” AGILE4.0, 24 February 2023. [Online]. Available: http://doi.org/10.5281/zenodo.4671896.. [Accessed 24 February 2023].

[5] P. D. Ciampa, G. L. Rocca and B. Nagel, “A MBSE Approach to MDAO Systems for the Development of Complex Products,” in *AIAA AVIATION 2020 FORUM*, 2020.

[6] J.-C. Chaudemar and P. de Saqui-Sannes, “MBSE and MDAO for Early Validation of Design Decisions: a Bibliography Survey,” in *2021 IEEE International Systems Conference (SysCon)*, 2021.

[7] D. V. Steward, “The design structure system: A method for managing the design of complex systems,” *IEEE Transactions on Engineering Management,* Vols. EM-28, p. 71–74, 1981.

[8] A. B. Lambe and J. R. R. A. Martins, “Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes,” *Structural and Multidisciplinary Optimization,* vol. 46, p. 273–284, January 2012.

[9] R. J. Lano, “The N2 Chart,” 1977.

[10] L. Boggero, P. D. Ciampa and B. Nagel, “An MBSE Architectural Framework for the Agile Definition of System Stakeholders, Needs and Requirements,” in *AIAA AVIATION 2021 FORUM*, 2021.

[11] L. Boggero, P. D. Ciampa and B. Nagel, “An MBSE architectural framework for the agile definition of complex system architectures,” in *AIAA AVIATION 2022 Forum*, 2022.

[12] C. Habermehl, G. Höpfner, J. Berroth, S. Neumann and G. Jacobs, “Optimization Workflows for Linking Model-Based Systems Engineering (MBSE) and Multidisciplinary Analysis and Optimization (MDAO),” *Applied Sciences,* vol. 12, p. 5316, May 2022.

[13] J. Bussemaker, L. Boggero and P. D. Ciampa, “From System Architecting to System Design and Optimization: A Link Between MBSE and MDAO,” *INCOSE International Symposium,* vol. 32, pp. 343-359, 2022.