33rd Annual INCOSE international symposium
hybrid event
Honolulu, HI, USA
July 15 - 20, 2023

The AGILE 4.0 Project: MBSE to Support Cyber-Physical Collaborative Aircraft Development

Jasper Bussemaker
German Aerospace Center (DLR)
Hein-Saß-Weg 22, 21129 Hamburg, Germany
jasper.bussemaker@dlr.de

Luca Boggero
German Aerospace Center (DLR)
Hein-Saß-Weg 22, 21129 Hamburg, Germany
luca.boggero@dlr.de

Björn Nagel
German Aerospace Center (DLR)
Hein-Saß-Weg 22, 21129 Hamburg, Germany
bjoern.nagel@dlr.de

Abstract. This paper presents the main output of the EU-funded H2020 AGILE 4.0 project: the AGILE 4.0 MBSE-MDAO Development Framework. The framework aims at connecting upstream Model-Based Systems Engineering (MBSE) activities to downstream Multidisciplinary Design Analysis and Optimization (MDAO) activities, applied to the development of complex aeronautical systems. Upstream activities include the definition of stakeholders, needs, semantic requirements, and modeling the system architecture design space. Downstream activities include defining, implementing, and executing cross-organizational and collaborative MDAO workflows to support system synthesis. The development framework is implemented in an Operational Collaborative Environment (OCE), a web-based platform for project data and modeling tool access. A business jet family design application case demonstrates the use of the development framework and OCE.

1. Introduction

Designing aeronautical systems has become more challenging in recent years due to a combination of increasing demands placed on sustainability and product performance, and a general increase in product complexity. These trends pose challenges to the competitiveness of European aviation industry, and cause projects to be delivered with exceeded cost and time budgets, and problems with quality. To keep the capability for developing innovative complex systems, it should be possible to integrate the highly multidisciplinary development process with requirements and constraints on the system itself, on components and technologies applied in the system, and on interactions with external and supporting systems. Overall, there is a need to streamline and accelerate the development of innovative systems, across diverse disciplines and domains, and throughout the entire system life-cycle (Ciampa & Nagel 2021).

Achieving this requires a move towards a more digital engineering process, leveraging the development of system models across the life-cycle. More precisely, **Model-Based Systems Engineering (MBSE)** approaches should be leveraged to track system development and connect to more detailed product design activities (Ciampa & Nagel 2021). Advantages of using an MBSE approach, compared to document-based systems engineering, include increased traceability between system and information elements, improved communication and clarity among designers and stakeholders, potential for automation of system verification and validation steps, and the re-use of results in future projects (Madni & Purohit 2019).

Data-driven decision-making will play an ever larger role in digital development processes. In the context of designing complex systems, the decision-making process is challenging in the sense that knowledge from engineers and engineering tools from diverse disciplines, backgrounds, and organizations needs to be integrated to come to a consistent, dependable, and optimized system design (Ciampa & Nagel 2020). **Multidisciplinary Design Analysis and Optimization (MDAO)** techniques can support the integration of such data (Sobieszczanski-Sobieski, Morris & Tooren 2015). However, there are several challenges with applying the current generation of MDAO systems, largely related to the efforts required to setup such collaborative MDAO workflows. A new approach developed in the AGILE project (running from 2015 to 2018), called the AGILE Paradigm, demonstrated a 40% reduction in setup time, allowing more time to be spent on running analyses, optimizations, and interpreting results (Ciampa & Nagel 2020).

The follow-up project AGILE 4.0 (running from 2019 to 2022) extends this work towards typical upstream systems engineering phases, in order to enable traceability to requirements and performing architectural trade-offs, while leveraging the AGILE Paradigm for implementing collaborative MDAO techniques (Ciampa & Nagel 2021). This paper presents an overview of the AGILE 4.0 MBSE-MDAO Development Framework. First, more details about the project itself are provided. Then, sec. 2 presents the framework. The implementation of the framework in an online environment is presented after that in sec. 3. Finally, the framework is demonstrated by the design of a business jet family in sec. 4.

1.1. Ambitions of the AGILE 4.0 Project

The AGILE 4.0[1] project (“Towards cyber-physical collaborative aircraft development“), coordinated by the German Aerospace Center (DLR), is funded by the European Commission and aims at improving the design process of complex systems by leveraging MBSE and MDAO approaches. The project extends the scope of the AGILE project running between 2015 and 2018, where cross-organizational collaborative MDAO processes and technologies were developed and improved (Ciampa & Nagel 2020). The extended scope includes MBSE processes and linking those to MDAO processes, in order to establish traceability from stakeholder needs and requirements to collaborative MDAO, via system architecture. The main ambition of AGILE 4.0 in the end is to reduce aircraft development costs and time-to-market throughout the aeronautical supply chain, and at the same time enable the development of more innovative systems incorporating novel and more sustainable technologies (Ciampa & Nagel 2021). In addition to the aeronautical product (e.g. an aircraft) itself, the development process should consider the design of supporting systems at the same time. Several typical aeronautical support systems are considered in the AGILE 4.0 project: the production (including the supply-chain) system, the certification system, and the maintenance system. These ambitions are realized by a project consortium consisting of 16 European, Canadian, and Brazilian partners from academia, research, and industry (see also fig. 1).

The main object of the project is to develop, implement, and demonstrate the effectiveness of the **AGILE 4.0 MBSE-MDAO Development Framework**. This framework covers several product development steps, including development scenario modeling, stakeholder identification, needs and requirements modeling, system architecture definition, setting-up and deploying large-scale analysis and optimization processes, and decision-making and system validation. Particular focus is placed on bridging systems engineering (MBSE) and design and optimization (MDAO) activities. The development of the framework is tackled by three project layers:

---

[1] “4.0” in the project name refers to the fourth industrial revolution.

1. *Specification, modeling, validation*: this layer defines the development framework conceptually, in terms of an architectural framework and several related processes. This layer forms the foundation driving the development of implementing technologies.
2. *Enabling technologies*: in this layer, technologies implementing the MBSE-MDAO framework are developed. This includes product and process models, collaborative development platforms, optimization and decision-making techniques, and various other system design tools.
3. *Application cases*: here, developed methodologies and technologies are applied to design a variety of systems and associated supporting systems. The project includes 7 parallel industry-driven application cases, each one focusing on one specific aspect of the aeronautical development life-cycle, such as manufacturing, assembly, certification, or maintenance. Each application case results in finalized MDAO workflows that perform trade-offs between at least two design objectives. An overview of application cases is shown in fig. 2.

Figure 1: AGILE 4.0 project consortium (left) and main aeronautical pillars addressed in the project (right). Reproduced from (Ciampa & Nagel 2021).

Finally, during the course of the project, the development process is guided by industry partners, and results are disseminated to the scientific and academic communities (available through agile4.eu). For more details on the project objectives and structure, the reader is referred to (Ciampa & Nagel 2021).

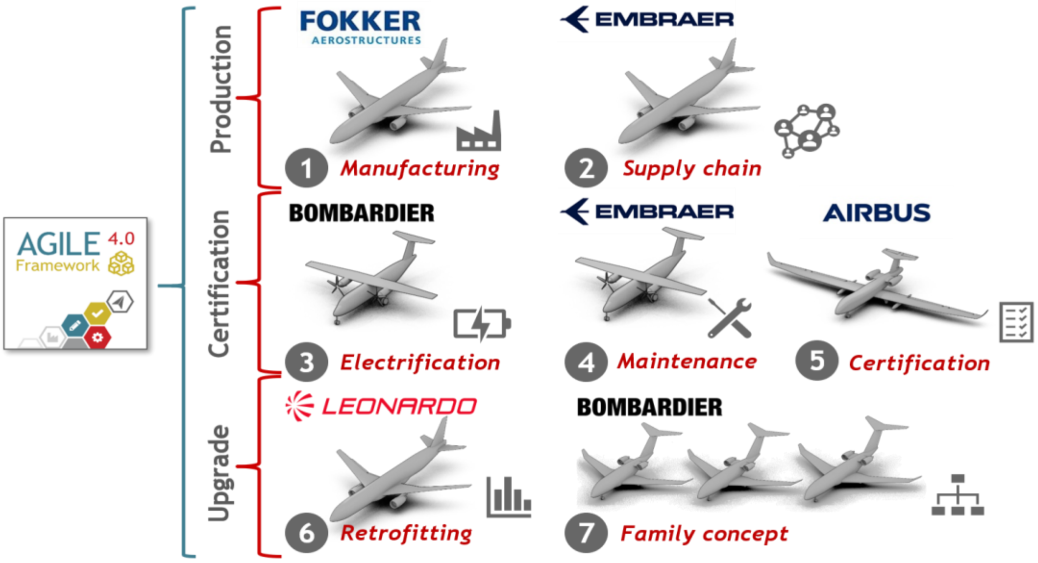


Figure 2: The seven industry-driven application cases supported by the AGILE 4.0 MBSE Framework. Reproduced from (Ciampa & Nagel 2021).

2. The AGILE 4.0 MBSE-MDAO Development Framework

The envisioned model-based conceptual framework aims to provide the capabilities to efficiently generate, evaluate, optimize, and perform trade-offs regarding aeronautical products, accounting for large numbers of architectural and design choices throughout the system life-cycle (Ciampa & Nagel 2021). The basis for this framework is provided by the well-established systems engineering approach. In order to provide the required acceleration and automation of the product development process, in particular the Model-Based Systems Engineering (MBSE) approach is adopted. In its vision for the year 2025, INCOSE expects MBSE usage to mature and support better understanding of complex system behavior earlier in the product life-cycle (INCOSE 2014). In addition to MBSE, Multidisciplinary Design Analysis and Optimization (MDAO) techniques are used to support the selection of optimal parameter values given some system architecture. As MBSE supports the overall engineering process, MDAO is seen as an important enabling tool within the systems engineering process (Bussemaker, Boggero & Ciampa 202AD).

The conceptual framework focuses in bridging *downstream product design* and *upstream architecting* phases. Here, the downstream product design phase roughly refers to activities typically performed in an MDAO context and focusing on the more detailed design of a selected system configuration. The upstream architecting phase refers to more typical systems engineering activities, focusing on requirements identification and the definition and selection of the system architecture. The development phases of the framework are shown in fig. 3.

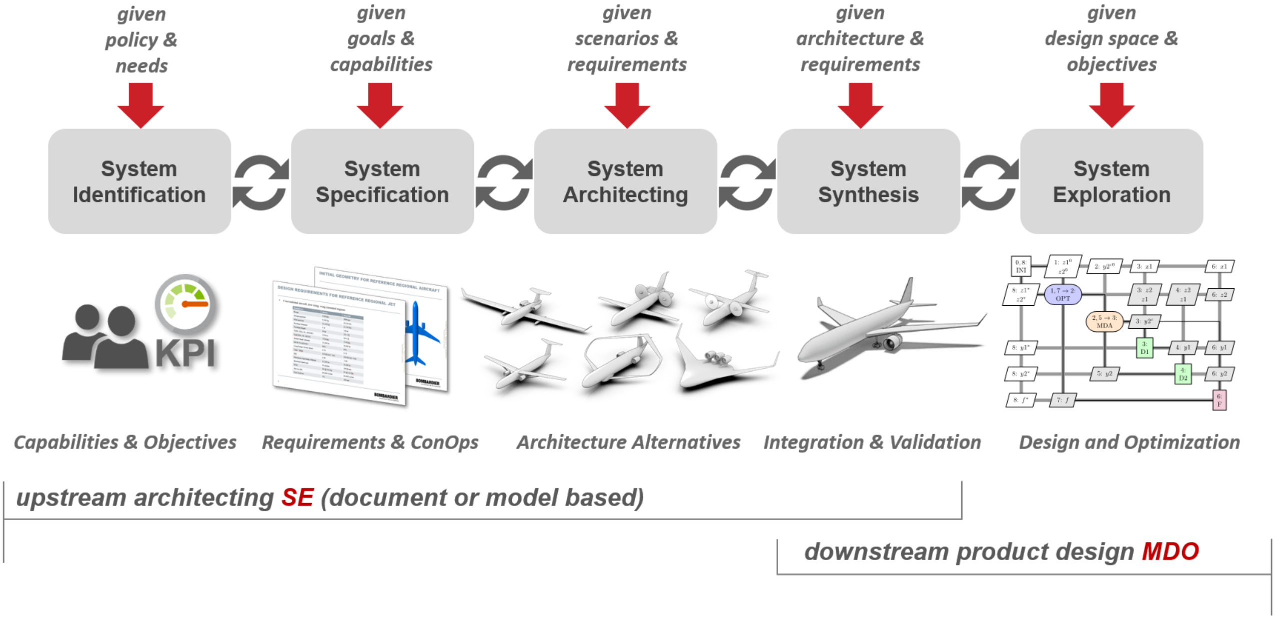


Figure 3: The AGILE 4.0 MBSE-MDAO Development Framework. Reproduced from (Ciampa & Nagel 2021).

Several sub-phases are defined under the general upstream and downstream phases. It should be noted that although in general the upstream phase is executed before the downstream phase, the complete conceptual framework should be seen as iterative in nature. The **upstream architecting phase** consists of:

- *Identification of stakeholders and needs*: for a given policy (e.g. design for sustainability), here system stakeholders are identified and their needs are solicited, resulting in the set of goals and capabilities which need to be met by the system under development. Stakeholders are additionally responsible for validating the system design.
- *Specification of ConOps and requirements*: for a given set of goals and capabilities, here the Concept of Operations (ConOps) is elaborated and described through scenarios, and

requirements are developed which implement the ConOps and stakeholder needs, and are verified by the system under development.

- *System architecting*: for a given set of scenarios and requirements, here architectural alternatives (in terms of function and form) of the system under development are defined. All architectures include functional (defining the functions to be provided), logical (the mapping of function to logical components), and physical (instantiations of logical components that can be evaluated) descriptions. An important outcome is a model of the architecture design space.

The **downstream product design phase** consists of:

- *System synthesis*: for a chosen architecture and given set of requirements, here the system exploration phase is prepared by formally defining the design space from requirements and/or architecture alternatives. Results from the system exploration phase are gathered and prepared for requirements verification and decision-making.
- *System exploration*: for a given architecture and design space, here the design competence required to complete the analysis and optimization of the system under development are selected. The design competences are then connected and integrated in an MDAO workflow, which is finally deployed and executed to yield a set of analyzed or optimized system designs. The system is analyzed in terms of the different engineering disciplines (e.g. aerodynamics, structures, costs, etc.).

The system exploration step is mainly inherited from the previous AGILE project (Ciampa & Nagel 2020). Subsequent sub-sections dive into more details of the upstream and downstream phases.

2.1. Upstream Architecting Phase

The upstream architecting phase consists of activities normally considered part of systems engineering, specifically MBSE in the case of the AGILE 4.0 MBSE-MDAO Development Framework. The development process starts with the identification of **stakeholders** . Stakeholders are individuals or organizations having a right, share, claim, or interest in a system or its characteristics. A hierarchy of stakeholders might exist, for example where different departments of some organization might have different interests in the system. For aeronautical systems, typical stakeholders include passengers, crew, airlines, manufacturers, and regulatory authorities. Stakeholders express **needs**: informal expressions of what the stakeholders expect of, or want to get out of the system, or the development project of the system. For example, airlines might want to maximize profit, whereas passengers are more interested in a safe flight with minimal delays. Needs form the basis for **requirements**: statements that express needs and their associated constraints and conditions in a consistent, unambiguous, and verifiable manner. Requirements can either be derived from one or more needs, or from one higher-level requirement.

Requirements consist of a statement and one or more attributes. *Attributes* define metadata such as requirement ID, author, creation date, and means of compliance, and exist for all requirements. The requirement statement is formulated according to some *pattern* defined for the chosen requirement *type*. Five requirement types are defined based on previous work by (Carson 2015):

- *Functional* requirement: defines what function must be performed to accomplish the system objectives.
- *Performance* requirement: defines how well the system functions must be performed.
- *Design constraint* requirement: limit the options open to the designer by imposing boundaries and limits.
- *Environmental* requirement: defines characteristics of the system when exposed in specific operating environments.

- *Suitability* requirement: defines "-ilities" to adhere to (e.g. safety, survivability, reliability, maintainability, security).

The requirement statement can be automatically generated from the pattern elements. This also makes the requirement statement semantic, enabling the requirement to be verified from quantities calculated in the downstream system design phase. Finally, requirements are also subject to *rules*, for example ensuring that requirements are unambiguous, verifiable, consistent, and comprehensible. The stakeholder, needs, and requirements ontology is shown in fig. 4 and published in (Boggero, Ciampa & Jepsen 2021). For more details on the processes, ontologies, and viewpoints for the definition of stakeholders, needs, and requirements in the AGILE 4.0 framework, the reader is referred to (Boggero, Ciampa & Nagel 2021).

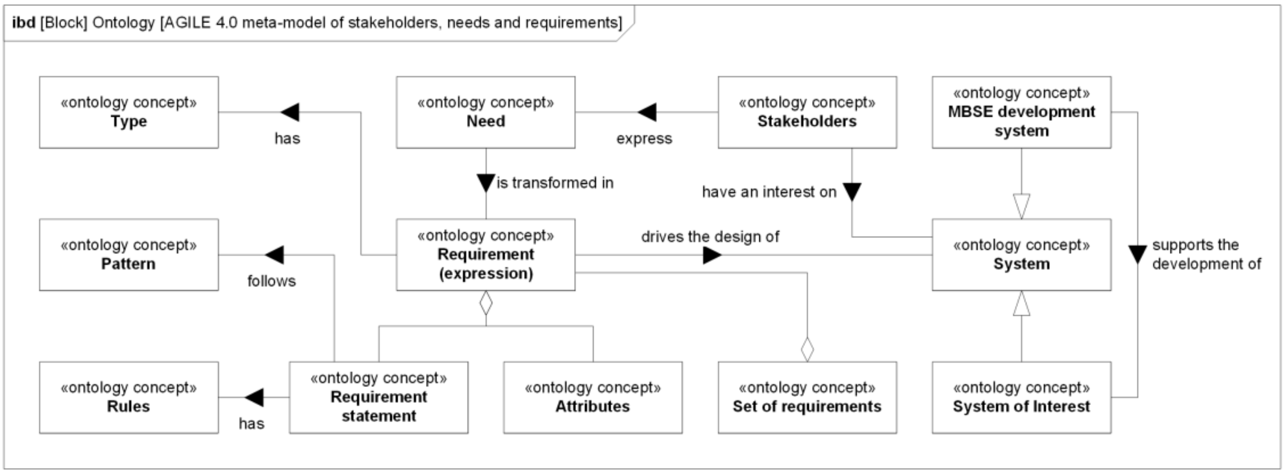


Figure 4: Stakeholder, needs, and requirements ontology. Reproduced from (Boggero, Ciampa & Nagel 2021).

After requirements have been defined, the system architecting step is performed. Here, functional, logical, and physical architecture alternatives are defined, as input to the downstream product design phase. The **functional architecture** defines all functions that the system should perform, and is derived from functional requirements. The functional architecture is constructed from *use cases*: combinations of high-level functions and operational conditions. Use cases are then decomposed into one or more *boundary functions*: solution-neutral functions that the system should perform in order to meet the design objectives. The ontology for the architecting step is shown in fig. 7.

The **logical architecture** is the first step towards finding the solution for how to fulfill the system objectives: here *logical components* are defined and allocated to boundary functions. In addition, components can *induce* functions, that then in turn also need to be fulfilled. For example, the boundary function "provide propulsive power" is fulfilled by a "turbofan" component, which in turn induces the "supply fuel" function, fulfilled by "fuel system". By allocating multiple components to functions, architecture alternatives can be defined: for example, the "provide propulsive power" can either be fulfilled by a "turbofan" or by a "turboprop" component. In addition, component-specific decisions, such as the selection of properties or number of instances, are used to define architecture alternatives. The set of all possible architecture is called the **architecture design space**, and can be searched to find the best architecture for fulfilling the system design objectives. The architecture design space viewpoint is shown in fig. 5. More details about the theory behind the definition of architecture design spaces, the Architecture Design Space Graph (ADSG), can be found in (Bussemaker, Ciampa & Nagel 2020).

Finally, the **physical architecture** represents a specific instantiation of a logical architecture, enriched with analysis results from the downstream MDAO process. Analysis results can for example represent component performance data, dimensions, or material and mass properties. For more details on the processes, ontologies, and viewpoints for the definition of system architectures in the AGILE 4.0 framework, the reader is referred to (Boggero, Ciampa & Nagel 2022).

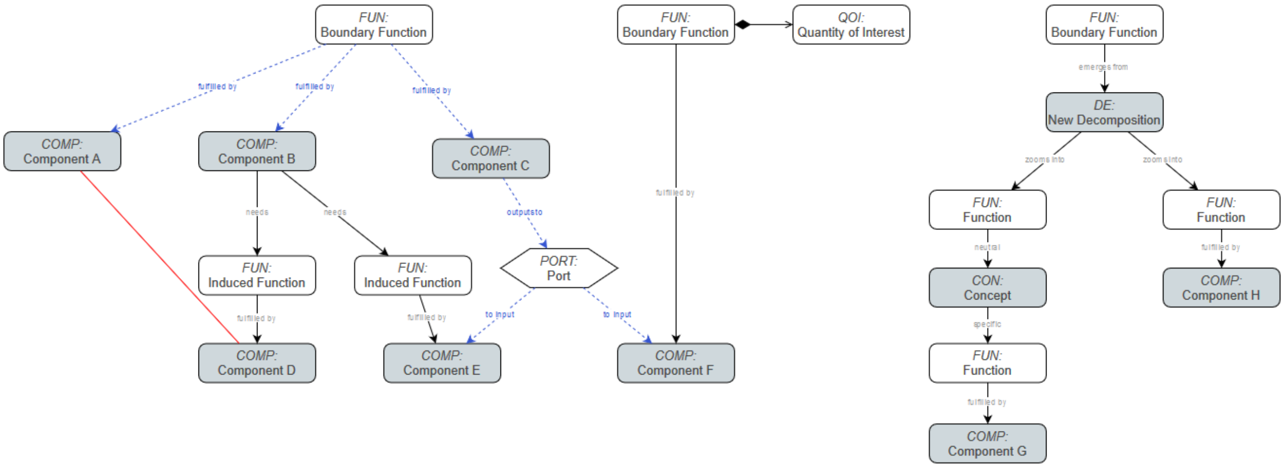


Figure 5: Viewpoint representing the architecture design space, refer to (Bussemaker, Ciampa & Nagel 2020) and (Bussemaker, Boggero & Ciampa 202AD) for more information. Reproduced from (Boggero, Ciampa & Nagel 2022).

2.2. Downstream Product Design and Optimization Phase

The design of complex systems involves the integration of many engineering disciplines, interacting in non-trivial ways including mutual dependency and feedback loops. It must be assumed that any part of the system can depend on and influence any other part of the system, leading to highly-coupled behavior. To design such systems, the product design and optimization phase leverages Multidisciplinary Design Analysis and Optimization (MDAO) techniques. MDAO offers the possibility for integrating disciplinary analysis tools, solving the mutual interdependencies, and resulting in consistent designs that are optimal on the system-level, rather than on the component- or discipline-level (Sobieszczanski-Sobieski, Morris & Tooren 2015).

The central concept in the downstream product design phase is the **design problem** (Torrigiani, Boggero & Nagel 2023): a formalization of the design space in terms of design parameters and design workflow. **Design parameters** are defined from the architecture design space and non-functional requirements, and are assigned roles in the design problem. A parameter either is an input to the design workflow if it is a design variable (a parameter that can be changed in order to improve the design) or static input parameter (e.g. representing assumptions or top-level requirements). Parameters can also be outputs of the workflow, in which case they might play the role of objective to be minimized or maximized, constraint to be satisfied, or generic quantity of interest to be tracked. The **design workflow** is an implementation of an MDAO process for automated and coupled exploration and analysis of the design problem. It includes design driver elements, such as an optimizer or a design-of-experiments generator, and design competences, disciplinary analysis tools that evaluate some aspect of the design. Design competences are connected to each other through a central data schema that represents the physical instantiation of the system under development. In addition, design competences can be executed remotely to meet intellectual property right constraints. The methodology for the definition, modeling, implementation, and execution of such collaborative cross-organization MDAO workflows is formalized in the AGILE Paradigm (Ciampa & Nagel 2020), and is reused in the AGILE 4.0 project.

The main contribution to the downstream design phase in the AGILE 4.0 project is the connection between MBSE and MDAO. Specifically, the definition of how design problems and parameters are derived from requirements and architecture design spaces, and how MDAO results can be used to verify requirements. The connection from MBSE to MDAO is established in the following ways (see also fig. 7):

1. Design parameters are specified either from logical components in the architecture design space, or directly from non-functional requirements.
2. Design competences are selected based on the need to analyze the influence of logical components, or by explicit definition as part of a test case associated to a requirement.
3. In addition to design variables derived from parameters, the design problem also contains design variables derived from architectural decisions in the architecture design space, which typically are categorical design variables.
4. Design solutions represent physical architectures, where all logical components have been instantiated and parameters assigned to them.

The ontology for the system design process including the four previously described connections between the system design phase and upstream phases is presented in fig. 7. For more details on the processes, ontologies, and viewpoints for the definition and execution of design problems in the AGILE 4.0 framework, the reader is referred to (Torrigiani, Boggero & Nagel 2023).

3. Implementation in the Operational Collaborative Environment

The AGILE 4.0 MBSE-MDAO Development Framework has been implemented by various technologies and made available to project participants through the Operational Collaborative Environment (OCE) (Baalbergen et al. 2022). The OCE is a web-based platform where projects and related design studies can be created, user access can be managed, and models can be collaboratively created for each design study: stakeholder, needs, and requirements models, as well as architecture design spaces and MDAO workflows. Then, the link between MBSE and MDAO models can be established, and used to provide input to the MDAO workflow, and use its outputs to verify requirements. An overview of technologies included in the OCE is shown in fig. 6.

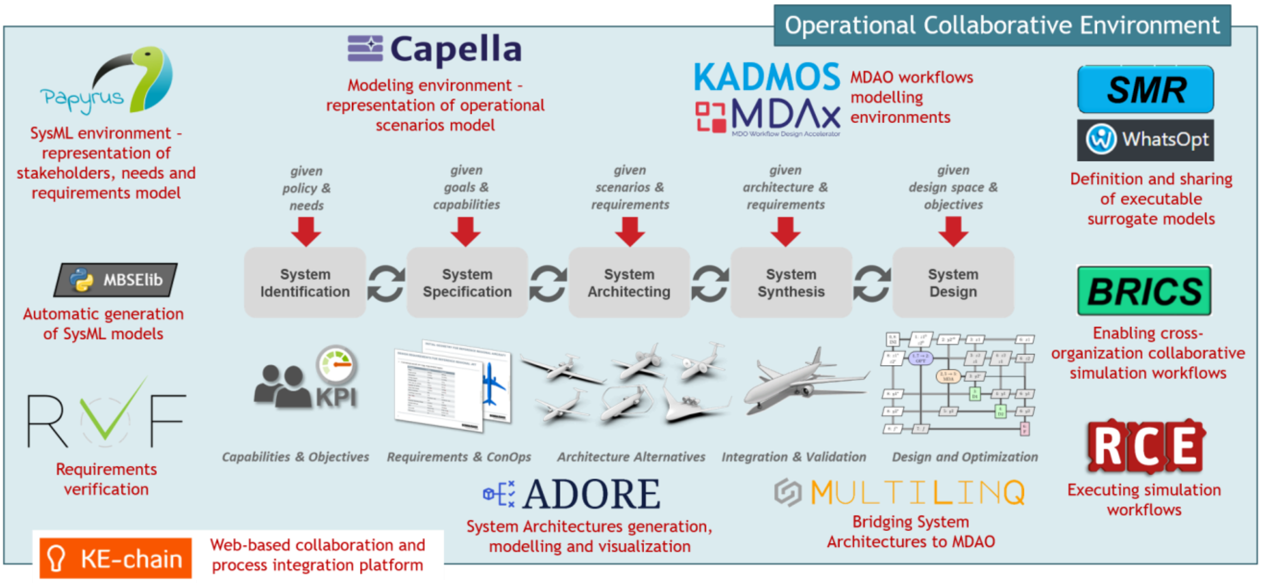


Figure 6: Overview of technologies integrated in the Operational Collaborative Environment (OCE) implementing the AGILE 4.0 MBSE-MDAO Development Framework. Reproduced from (Boggero et al. 2022).

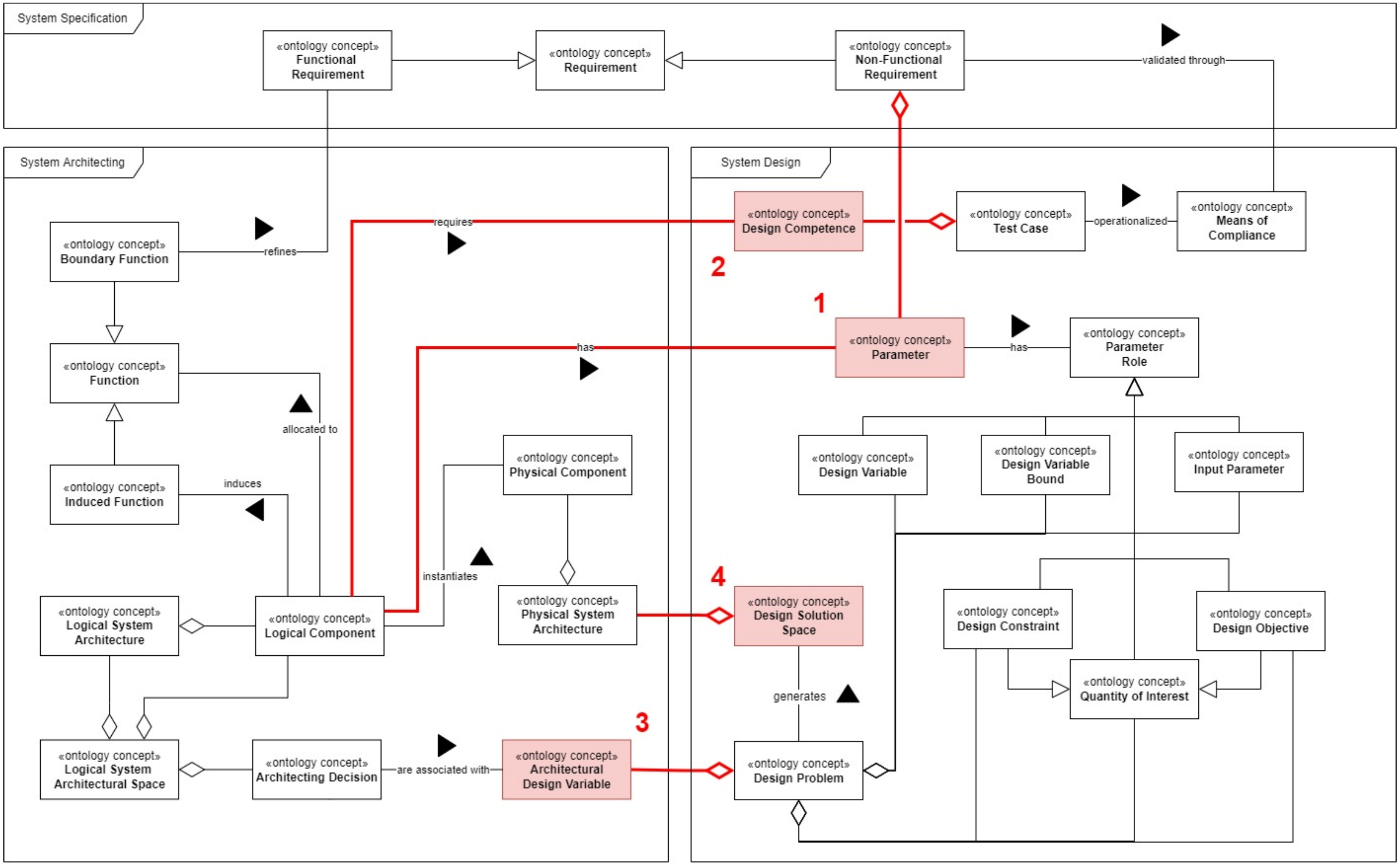


Figure 7: System architecting and design ontologies, and their link to requirements. Reproduced from (Torrigiani, Boggero & Nagel 2023).

The OCE is implemented using KE-chain[2] (developed by KE-works), a web-based process modeling platform for collaborative data management. As a platform, KE-chain enables the definition of projects and associated design studies, and manages user authentication and data access. The definition of stakeholders, needs, requirements, scenarios, and functional architectures is done directly from the KE-chain interface. To inspect the defined models, MBSElib (Boggero et al. 2022) (developed by the DLR) is used to automatically generate SysML models from this data, for display and inspection in Eclipse Papyrus[3]. In addition, operational scenarios can be modeled using Capella[4]. Modeling, visualizing, and inspecting the architecture design space is done using ADORE (Bussemaker, Boggero & Ciampa 2022) (DLR), a web-based editing environment accessible through the OCE. ADORE architecture design space models are initialized from boundary functions defined in the OCE design study, thereby establishing a link between the requirements definition and system architecting steps.

Selection of design competences and connecting MBSE models to MDAO workflows can be done using the Requirements Verification Framework (RVF) (Bruggeman et al. 2022) (Delft University of Technology) and MultiLinQ (Bussemaker, Boggero & Ciampa 2022) (DLR). RVF is embedded in the OCE user interface and enables the system engineer to link requirement parameters to the central data schema used in the MDAO workflow. Additionally, the RVF selects analysis tools based on test cases assigned to requirements, thereby providing input to the MDAO workflow formulation step. RVF can also use analysis output to extract parameter values and verify requirements. MultiLinQ is a web-based tool for linking architecture design space elements to the central data schema, and for identifying which architecture elements are linked to which disciplinary tool using a Component-Tool (CT) matrix.

The formulation of MDAO workflow from tool interface specifications and the selection of design competences can be done using KADMOS (Gent & Rocca 2019) (TU Delft) or MDAx (Page-Risueño et al. 2020) (DLR). Tools communicate data using the CPACS data schema (Alder et al. 2020), an open-source, XML-based format for the parameterization of aircraft designs. Executable workflows are automatically created from the workflow models, either via direct export or through the CMDOWS (Gent, La Rocca & Hoogreef 2018) (TU Delft) data format, and are executed in the RCE environment[5] (DLR). In addition to disciplinary analysis tools, workflows can also include surrogate models of tools, made available to project partners through WhatsOpt (Lafage, Defoort & Lefebvre 2019) (ONERA) and the Surrogate Model Repository (SMR) (Moerland et al. 2020) (NLR). To enable cross-organization data exchange and thereby protect the intellectual property of each disciplinary expert, Brics (Baalbergen et al. 2017) (NLR) is used.

4. Showcase: Designing a Family of Business Jets

The AGILE 4.0 MBSE-MDAO Development Framework is demonstrated using application case 7 (see fig. 1.2), the design of a business jet family. Part of the development of this application case has already been published in (Bussemaker et al. 2022), however here it is extended with new optimization results. The application case is driven by Bombardier, and deals with the concurrent design of three business jets for various design ranges and cabin lengths, see fig. 8 for a visualization. The main trade-off to be evaluated is between recurring Direct Operation Costs (DOC) and OEM (manufacturer) non-recurring costs (NRC), both to be minimized. A trade-off between these two costs can be achieved by varying the amount of commonality between the three aircraft: sharing

---

[2] https://ke-chain.com/

[3] https://www.eclipse.org/papyrus/

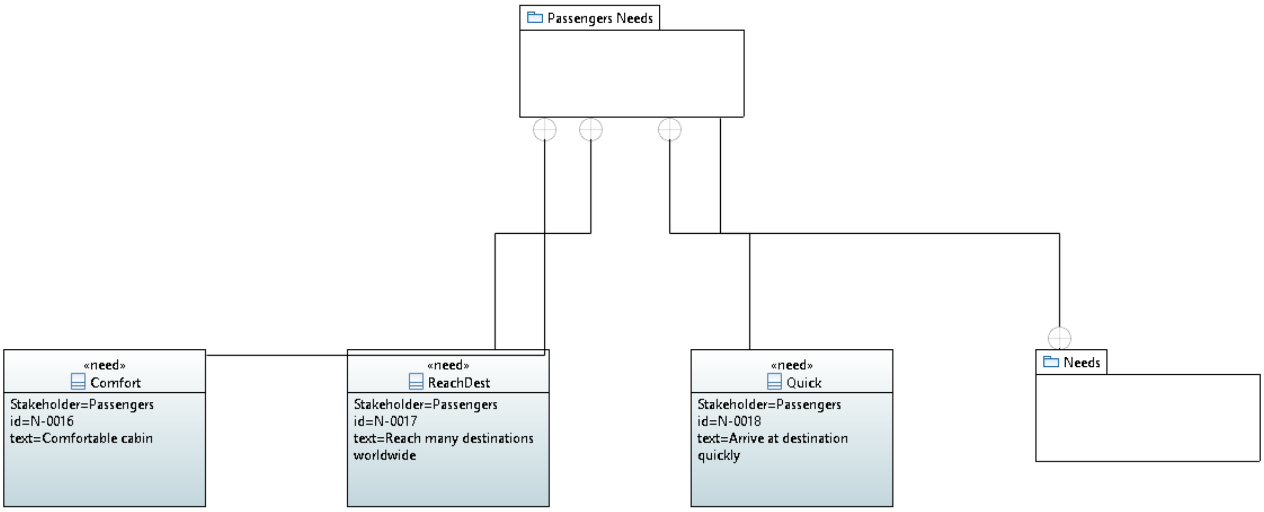
[4] https://www.eclipse.org/capella/

[5] https://rcenvironment.de/

components can reduce development and manufacturing costs, however might lead to an increase in operating costs due to usage of components not sized for the operating point that the aircraft is designed for (e.g. an oversized wing or engine).

Figure 8: Notional render of the business jet family consisting of three aircraft designed for various ranges and cabin sizes. Reproduced from (Bussemaker et al. 2022).

The system of interest is the family of business jets, with each individual aircraft represented as a component within the system. The system development starts with the identification of stakeholders and needs. Stakeholders include OEM, Operator, Engine OEM, Passengers, Pilots, and Regulatory Authorities. The stakeholders and needs are entered in the OCE and correctly linked to each other. An export to SysML of the needs of the "Passengers" stakeholder, generated with MBSElib (Boggero et al. 2022), is shown in fig. 9.



Stakeholder=Passengers
id=N-0016
text=Comfortable cabin

Stakeholder=Passengers
id=N-0017
text=Reach many destinations worldwide

Stakeholder=Passengers
id=N-0018
text=Arrive at destination quickly

Figure 9: Needs model for the needs of the Passengers stakeholder of the business jet family. Reproduced from (Bussemaker et al. 2022).

Stakeholder needs are transformed into requirements, to make them unambiguous, consistent, and verifiable. Requirements are entered through the OCE user interface, including the definition of parameters, systems, and other requirement pattern elements. The requirement statement text is generated automatically. A subset of the requirements model, also exported to SysML using MBSElib, is shown in fig. 10.

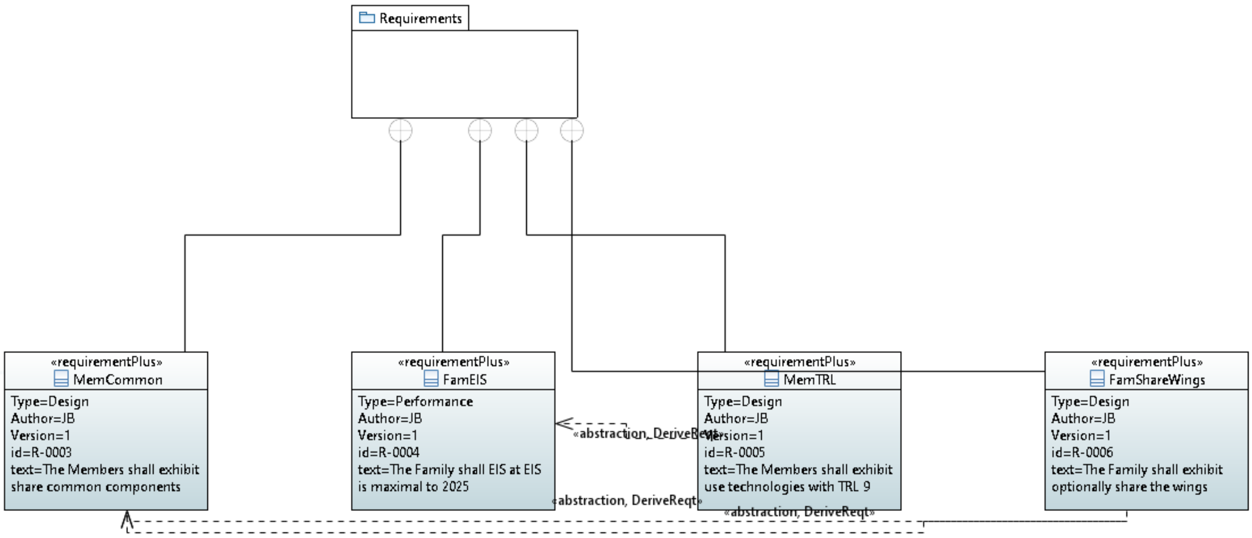


Figure 10: Requirements set showing several requirements related to family size and introduction into service for the business jet family. Reproduced from (Bussemaker et al. 2022).

After requirements have been specified, the functional architecture is defined. This is done by elaborating on functional requirements by defining use cases, and from that extracting boundary functions, such as “Contain Passengers”, “Regulate Temperature”, and “Transport Passengers”. The functional architecture forms the basis for the logical architecture design space model created using ADORE (Bussemaker, Boggero & Ciampa 2022). This design space model assigns aircraft components to aircraft-level functions, and models component sharing choices, such as whether wings, engines, landing gear, on-board systems, or the empennage should be shared between aircraft or not. In addition, design parameters are specified, defining inputs to (e.g. wing sweep, cruise altitude, cruise speed) and output from (e.g. weights, costs, fuel burn) the design workflow. An overview of the architecture design space model is presented in fig. 11, showing how components are linked to the top-level function, and aircraft-level components are allocated to aircraft-level functions. More details can be found in (Bussemaker et al. 2022).

Each family architecture will be sized and evaluated using an MDAO workflow, integrating several disciplinary analysis tools through the CPACS central data schema (Alder et al. 2020). Several tools available from partners within the project consortium are used, including ASTRID for on-board system sizing developed by Politecnico di Torino, tools for flap CL_{max} estimation and tailplane sizing developed by Università di Napoli Federico II (UNINA), the composite wing sizing tool PROTEUS developed by Delft University of Technology, mission analysis and overall aircraft design tools developed by the DLR, and cost estimation tools developed by Rheinisch-Westfälische Technische Hochschule (RWTH). Each of these tools is executed at the location of the respective partner, except for several tools that are integrated using surrogate models.

The MDAO workflow is created using MDAx (Page-Risueño et al. 2020) and consists of two levels: an aircraft-level workflow that sizes one specific aircraft considering commonality sharing decisions, and a family-level workflow that connects the aircraft-level workflows and calculates family-level costs. The aircraft-level workflow including associated partners is shown in fig. 12. The workflow is implemented in RCE after being exported from MDAx. Data files are transferred between partners using Brics (Baalbergen et al. 2017).

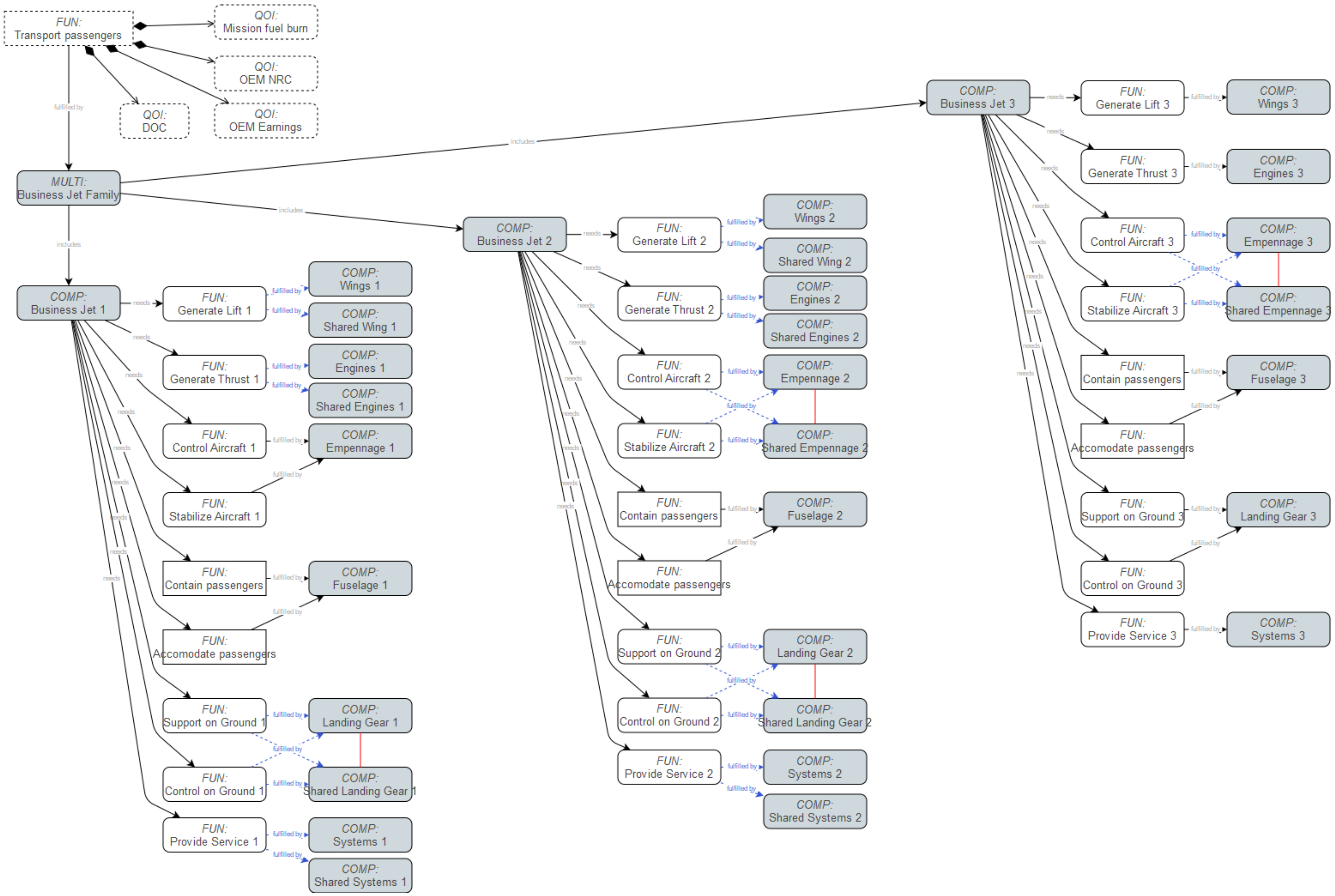


Figure 11: Architecture design space model of the business jet family, showing three aircraft-level architectures with related component sharing decisions. Reproduced from (Bussemaker et al. 2022).

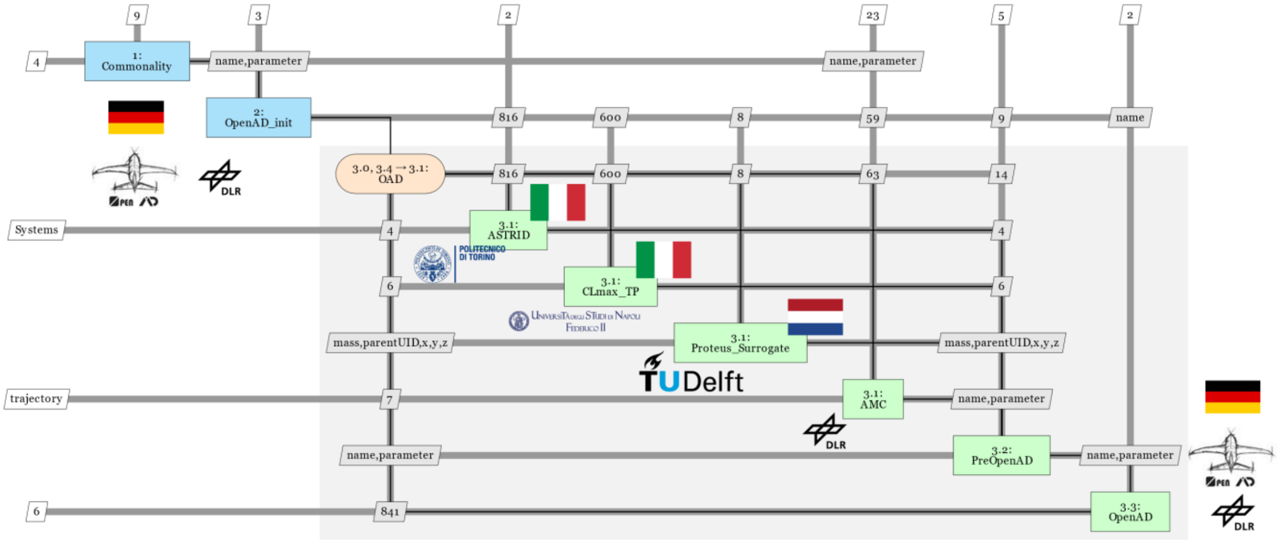
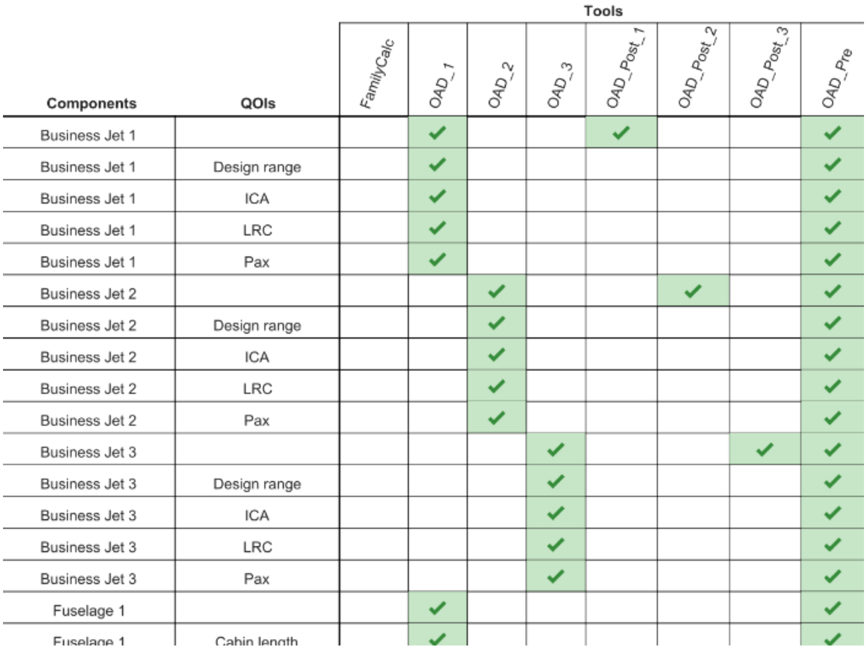


Figure 12: XDSM representation of the aircraft-level MDAO workflow, showing how tools of different partners residing in different countries are connected. Adopted from (Bussemaker et al. 2022).

The design problem itself is formalized by the ADORE architecture design space model, as that model contains the architectural decisions and other design parameters. The connection between the architecture model, which is represented as an Architecture Design Space Graph (ADSG), and the MDAO workflow, which represents data using CPACS, is made using MultiLinQ (Bussemaker, Boggero & Ciampa 202AD). From this, a Component-Tool (CT) matrix can be created, see fig. 13, showing how architecture elements and associated design parameters are mapped to disciplinary analysis tools.



		Tools							
Components	QOIs	FamilyCalc	OAD_1	OAD_2	OAD_3	OAD_Post_1	OAD_Post_2	OAD_Post_3	OAD_Pre
Business Jet 1			✓			✓			✓
Business Jet 1	Design range		✓						✓
Business Jet 1	ICA		✓						✓
Business Jet 1	LRC		✓						✓
Business Jet 1	Pax		✓						✓
Business Jet 2				✓			✓		✓
Business Jet 2	Design range			✓					✓
Business Jet 2	ICA			✓					✓
Business Jet 2	LRC			✓					✓
Business Jet 2	Pax			✓					✓
Business Jet 3					✓			✓	✓
Business Jet 3	Design range				✓				✓
Business Jet 3	ICA				✓				✓
Business Jet 3	LRC				✓				✓
Business Jet 3	Pax				✓				✓
Fuselage 1			✓						✓
Fuselage 1	Cabin length		✓						✓

Figure 13: Component-Tool (CT) matrix created using MultiLinQ, showing how architecture components and QOIs (rows) are mapped to disciplinary analysis tools of the family-level workflow (columns). Reproduced from (Bussemaker et al. 2022).

The optimization problem has been executed using SEGOMOE (Bartoli et al. 2016), a surrogate-based optimization algorithm which has recently been extended to also support multi-objective

(Grapin et al. 2022) and mixed-integer (Saves et al. 2022) problems. Access to the optimization algorithm is provided by WhatsOpt (Lafage, Defoort & Lefebvre 2019) through an ask-tell interface. At each optimization loop, see fig. 14 for a sequence diagram, first a new design vector is requested ("asked" for), from which an architecture instance is generated using ADORE, which is subsequently mapped to a CPACS file using MultiLinQ. The input file is transferred to the MDAO workflow, running in RCE, using Brics. Once the MDAO workflow finishes evaluating the CPACS file, an output CPACS file is returned to ADORE, metrics are extracted using MultiLinQ, and the next optimization loop can start.

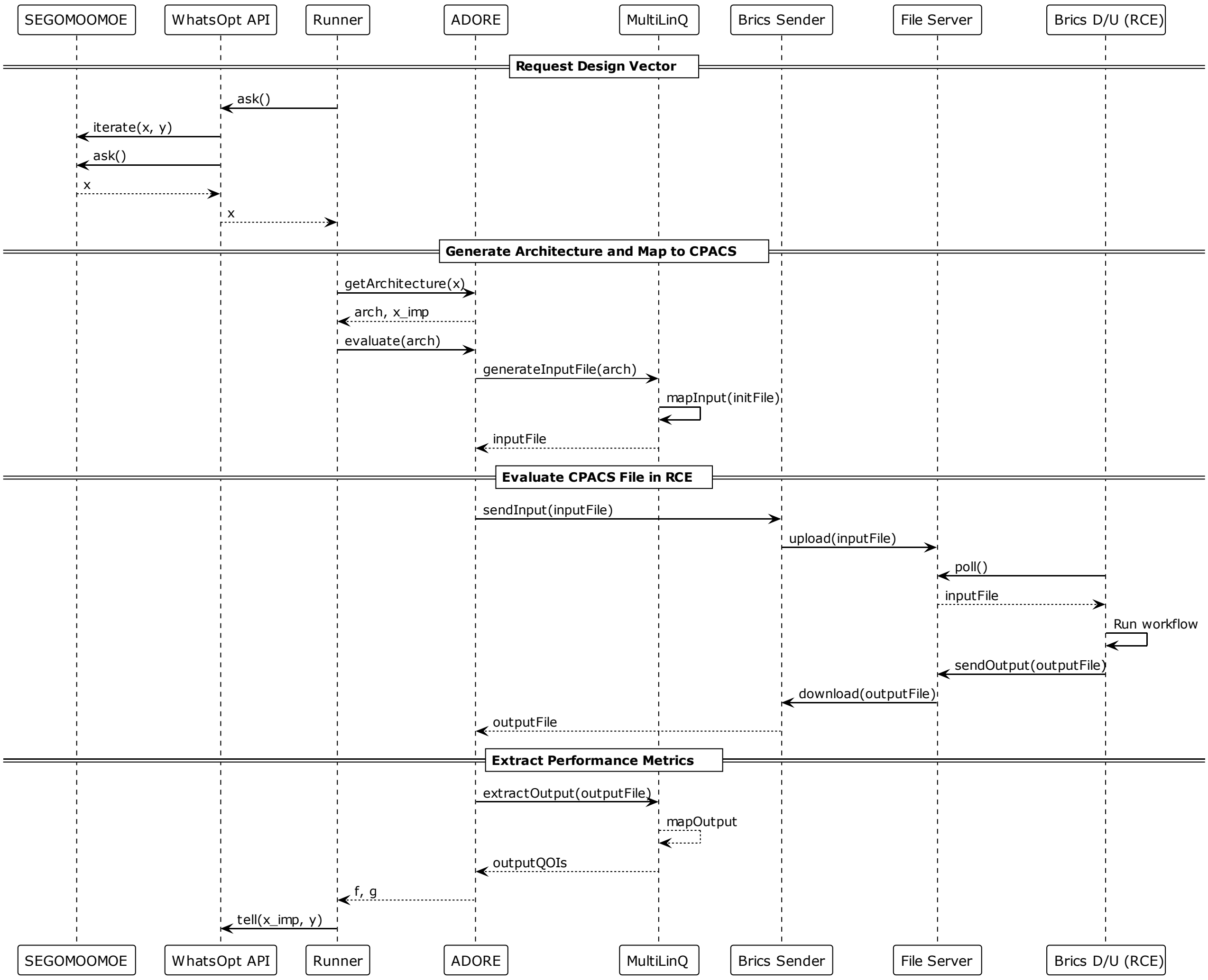


Figure 14: Sequence diagram showing one optimization loop: a design vector is requested from SEGOMOOMOE through WhatsOpt. This is then used to generate an architecture in ADORE, map to a CPACS file using MultiLinQ, and execute the MDAO workflow using Brics and RCE.

The implemented design problem is executed in two phases; first, a 50-point Design of Experiments (DOE) is generated using latin-hypercube sampling to verify correct behavior of the workflow and to provide an initial database of points for the optimizer. Then, the optimizer is executed and additional infill points are calculated to extend the Pareto front. Each of the design points took between 1 and 2 hours to converge. fig. 15 shows the design points and associated Pareto front for the two objectives, DOC and OEM NRC. Also plotting the degree of commonality (i.e. the number of components shared between aircraft) in fig. 16, shows that in general an increase in commonality indeed reduces OEM NRC (i.e. reducing investment and non-recurring costs), however at the cost of increasing DOC (i.e. reduced aircraft efficiency).

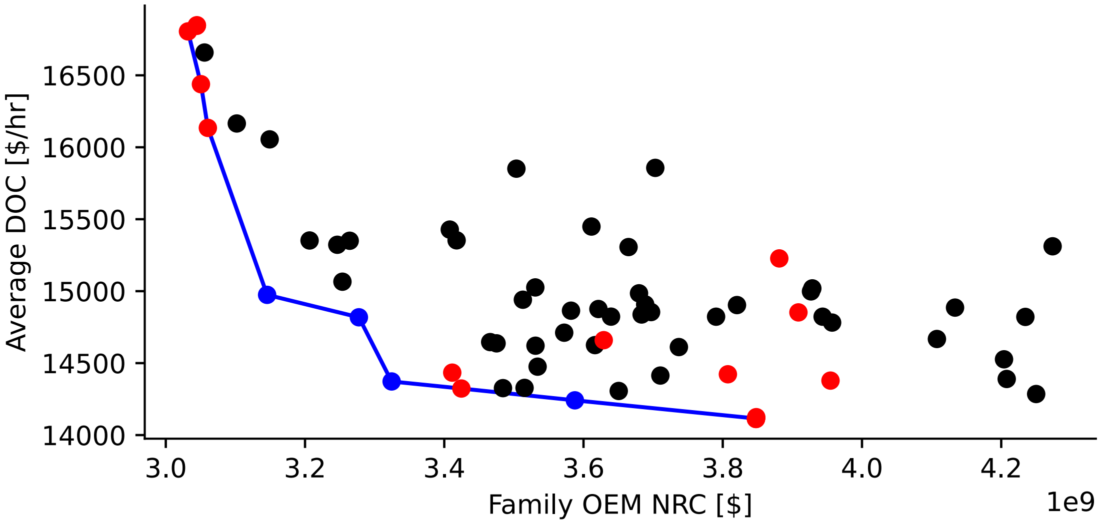


Figure 15: Business jet family analysis results, showing design of experiment points (black), infill points (red) and the Pareto front (blue) between the two objectives.

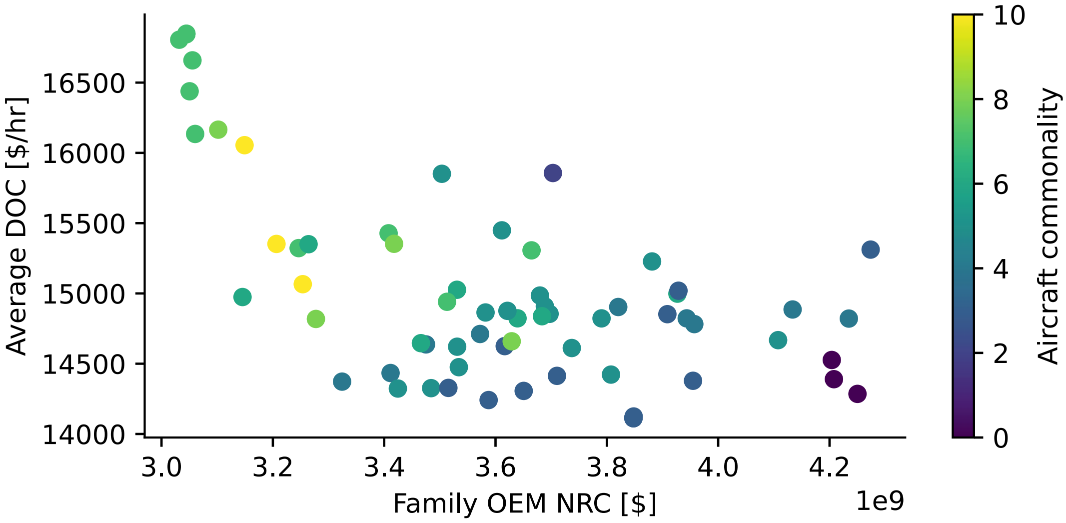


Figure 16: Business jet family analysis results, showing how an increasing level of commonality generally leads to lower investment costs by the OEM, however negatively impacts the aircraft operating costs.

5. Conclusions

As developing aeronautical systems is becoming more challenging due to high demands placed on sustainability and performance, novel digital design methods are needed to keep the European aviation industry competitive. In the EU-funded H2020 AGILE 4.0 project, Model-Based Systems Engineering (MBSE) is applied as a basis for this digital design process, and Multidisciplinary Design Analysis and Optimization (MDAO) techniques are used to provide system-level analysis capabilities. Collaborative MDAO capabilities are inherited from the predecessor AGILE project. MBSE and MDAO are integrated in the **AGILE 4.0 MBSE-MDAO Development Framework**, of which an overview has been presented in this paper. The development framework is tested in seven industry-driven application cases. More details can be found in the many publications produced throughout the project, openly available at the project website: agile4.eu.

The development framework is divided in two phases with several steps:

1. The **upstream architecting phase** (typically MBSE):
    - *Identification of stakeholders and needs*: stakeholders and needs are identified and modeled, resulting in a set of goals and capabilities to be delivered.
    - *Specification of ConOps and requirements*: needs are elaborated in structured and consistent requirements, and the ConOps is elaborated in scenarios, resulting in a system specification that can be verified.
    - *System architecting*: the requirements are used as a basis for creating a logical system architecture design space model, which can be used to generate architecture alternatives.
2. The **downstream product design phase** (typically MDAO):
    - *System synthesis*: the system is integrated/synthesized for a chosen architecture, delivering a sized system for requirements verification. Here the physical system representation is initialized, analyzed, and/or optimized.
    - *System exploration*: for a given architecture and system design space, design competences needed for the design system are developed and selected. These are then used to implement the MDAO workflow, which is then deployed and executed to support system synthesis.

The development framework is implemented in the Operational Collaborative Environment (OCE), a web-based platform based on the KE-chain framework where all project partners have access to the latest design models and data. Several additional applications are integrated in the OCE, implementing the different steps of the development framework.

For the *upstream* architecting phase: MBSElib automatically generates SysML models for inspection of defined stakeholders, needs, and requirements using Eclipse Papyrus; Capella is used to model scenarios; and ADORE is used to model and inspect architecture design spaces and to generate architecture alternatives. For *connecting* MBSE to MDAO: the Requirements Verification Framework (RVF) is used to select design competences based and verify requirements using product data; and MultiLinQ is used to connect architecture design space elements to product data for synchronization and checking connections to design competences. For the *downstream* product design phase: KADMOS and MDAx can be used to formulate MDAO workflows from a design competence database; CPACS is used as a central data format for representing and exchanging product data; RCE is used to execute the MDAO workflows; WhatsOpt and SMR are used to publish and integrate surrogate models; and Brics is used for cross-organization data exchange.

The application of the AGILE 4.0 MBSE-MDAO Development Framework is demonstrated using the seventh application case: the design of a business jet family. It is shown how stakeholders, needs, requirements, and the architecture design space is modeled, how the cross-organization collaborative MDAO is implemented, and how the architecture design space model is connected to the MDAO product model. Results of an architecture optimization loop are presented, showing the application of a new surrogate-based optimization algorithm, and demonstrating that the development framework can be used to find a multi-objective Pareto front trading-off two system-level design objectives.

Overall, the development framework supports the quick formulation and implementation of cross-organizational MDAO workflows for analysis and optimization of multiple system levels and domains concurrently. Throughout the design process, traceability is ensured, enabling the selection of design competences from requirements, and the verification of requirements based on produced design data. Additionally, the architecture design can be implemented within the design loop, opening up the architectural design space earlier in the design process.

6. Acknowledgments

The research presented in this paper has been performed in the framework of the AGILE 4.0 project (Towards cyber-physical collaborative aircraft development) and has received funding from the European Union Horizon 2020 Programme under grant agreement n. 815122. The authors would like to recognize the entire project consortium, and Pier Davide Ciampa (Airbus DS, former AGILE 4.0 project coordinator) in particular, for their activities performed in the project.

References

Alder, M, Moerland, E, Jepsen, J & Nagel, B 2020, ‘Recent advances in establishing a common language for aircraft design with CPACS’, *Aerospace europe conference*.

Baalbergen, EH, Vankan, J, Boggero, L, Bussemaker, JH, Lefebvre, T & Beijer, B 2022, ‘Advancing cross-organizational collaboration in aircraft development’, *AIAA AVIATION 2022 FORUM*, Chicago, USA.

Baalbergen, E, Kos, J, Louriou, C, Campguilhem, C & Barron, J 2017, ‘Streamlining cross-organisation product design in aeronautics’, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 231, no. 12, pp. 2192–2202.

Bartoli, N, Bouhlel, MA, Kurek, I, Lafage, R, Lefebvre, T, Morlier, J, Priem, R, Stilz, V & Regis, R 2016, ‘Improvement of efficient global optimization with application to aircraft wing design’, *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, no. June, pp. 1–28.

Boggero, L, Ciampa, PD & Jepsen, J 2021, *AGILE 4.0 MBSE ontology*, Zenodo.

Boggero, L, Ciampa, PD & Nagel, B 2022, ‘An MBSE architectural framework for the agile definition of complex system architectures’, *AIAA AVIATION 2022 FORUM*.

Boggero, L, Ciampa, PD & Nagel, B 2021, ‘An MBSE architectural framework for the agile definition of system stakeholders, needs and requirements’, *AIAA AVIATION 2021 FORUM*.

Boggero, L, Lefèbvre, T, Vankan, J, Beijer, B, Saluzzi, V & Nagel, B 2022, ‘The AGILE 4.0 MBSE-MDAO development framework: Overview and assessment’, *33rd congress of the international council of the aeronautical sciences, ICAS 2022*, Stockholm, Sweden.

Bruggeman, A-L, Manen, B van, Laan, T van der, Berg, T van den & Rocca, GL 2022, ‘An MBSE-based requirement verification framework to support the MDAO process’, *AIAA AVIATION 2022 forum*, American Institute of Aeronautics; Astronautics.

Bussemaker, JH, Boggero, L & Ciampa, PD 2022, ‘From system architecting to system design and optimization: A link between MBSE and MDAO’, *32nd annual INCOSE international symposium*, Detroit, MI, USA.

Bussemaker, JH, Ciampa, PD & Nagel, B 2020, ‘System architecture design space exploration: An approach to modeling and optimization’, *AIAA AVIATION 2020 FORUM*, American Institute of Aeronautics; Astronautics.

Bussemaker, JH, Ciampa, PD, Singh, J, Fioriti, M, Cabaleiro, C, Wang, Z, Peeters, DJM, Hansmann, P, Vecchia, PD & Mandorino, M 2022, ‘Collaborative design of a business jet family using the AGILE 4.0 MBSE environment’, *AIAA AVIATION 2022 FORUM*, Chicago, USA.

Carson, RS 2015, ‘Implementing structured requirements to improve requirements quality’, *INCOSE International Symposium*, vol. 25, no. 1, Wiley, pp. 54–67.

Ciampa, PD & Nagel, B 2021, 'Accelerating the development of complex systems in aeronautics via MBSE and MDAO: A roadmap to agility', *AIAA AVIATION 2021 FORUM*, American Institute of Aeronautics; Astronautics, Virtual Event.

Ciampa, PD & Nagel, B 2020, 'AGILE paradigm: The next generation of collaborative MDO for the development of aeronautical systems', *Progress in Aerospace Sciences*, vol. 119.

Gent, I van, La Rocca, G & Hoogreef, MFM 2018, 'CMDOWS: a proposed new standard to store and exchange MDO systems', *CEAS Aeronautical Journal*, vol. 0, no. 0, Springer Vienna, p. 0, viewed <http://link.springer.com/10.1007/s13272-018-0307-2>.

Gent, I van & Rocca, GL 2019, 'Formulation and integration of MDAO systems for collaborative design: A graph-based methodological approach', *Aerospace Science and Technology*, vol. 90, Elsevier BV, pp. 410–433.

Grapin, R, Diouane, Y, Morlier, J, Bartoli, N, Lefebvre, T, Saves, P & Bussemaker, JH 2022, 'Constrained multi-objective bayesian optimization with application to aircraft design', *AIAA AVIATION 2022 forum*, American Institute of Aeronautics; Astronautics.

INCOSE 2014, *Systems engineering vision 2025*, INCOSE.

Lafage, R, Defoort, S & Lefebvre, T 2019, 'WhatsOpt: a web application for multidisciplinary design analysis and optimization', *AIAA aviation 2019 forum*, American Institute of Aeronautics; Astronautics, Reston, Virginia.

Madni, AM & Purohit, S 2019, 'Economic Analysis of Model-Based Systems Engineering', *Systems*, vol. 7, no. 1, p. 12.

Moerland, E, Ciampa, PD, Zur, S, Baalbergen, E, Noskov, N, DIppolito, R & Lombardi, R 2020, *Collaborative architecture supporting the next generation of MDAO within the AGILE paradigm*, vol. 119, Elsevier BV, p. 100637.

Page-Risueño, A, Bussemaker, JH, Ciampa, PD & Nagel, B 2020, 'MDAx: Agile generation of collaborative MDAO workflows for complex systems', *AIAA AVIATION 2020 FORUM*, American Institute of Aeronautics; Astronautics.

Saves, P, Diouane, Y, Bartoli, N, Lefebvre, T & Morlier, J 2022, *A mixed-categorical correlation kernel for gaussian process*, arXiv, viewed <https://arxiv.org/abs/2211.08262>.

Sobieszczanski-Sobieski, J, Morris, A & Tooren, MJL van 2015, *Multidisciplinary design optimization supported by knowledge based engineering*, John Wiley & Sons, Ltd, pp. 1–378.

Torrigiani, F, Boggero, L & Nagel, B 2023, 'An MBSE architectural framework for the agile design and optimization of complex systems', *AIAA AVIATION 2023 FORUM*, San Diego, CA, USA.

Biography

Jasper Bussemaker. Jasper received his MSc in Aerospace Engineering with a focus on aircraft design and MDO from Delft University of Technology in 2018. Currently he researches system architecture optimization and MDO at the DLR Institute of System Architectures in Aeronautics in Hamburg, Germany. He is developing methods and software for modeling architecture design spaces, for using MDO to evaluate candidate architectures, and for solving architecture optimization problems.

Luca Boggero. Luca obtained in 2018 his PhD in Aerospace Engineering at Politecnico di Torino with a dissertation on Multidisciplinary Design and Optimization (MDO), Model Based Systems Engineering (MBSE) and design of aircraft subsystems. He now works as a Research Scientist at the DLR Institute of System Architectures in Aeronautics in Hamburg, and he leads the Digital Development Process Group. He coordinates and is involved in research projects within the context of MDO, Systems Engineering and MBSE**.**

Björn Nagel. Dr. Björn Nagel is director of the Institute of System Architectures in Aeronautics at the German Aerospace Center (DLR). His research is focused on digital engineering methods including MBSE and MDO enabling large and heterogeneous teams to model and optimize aviation as a system-of-systems. Pathways to climate-neutral air transport, co-design for industrialization and military air systems are major application**.**