Advancing Cross-Organizational Collaboration in Aircraft Development

Baalbergen, Erik H. ; Vankan, Jos; Boggero, Luca ; Bussemaker, J.H.; Lefebvre, Thierry; Beijer, Bastiaan; Bruggeman, A.M.R.M.; Mandorino, Massimo

DOI
10.2514/6.2022-4052

Publication date
2022

Document Version
Final published version

Published in
AIAA AVIATION 2022 Forum

Citation (APA)
Baalbergen, E. H., Vankan, J., Boggero, L., Bussemaker, J. H., Lefebvre, T., Beijer, B., Bruggeman, A. M. R. M., & Mandorino, M. (2022). Advancing Cross-Organizational Collaboration in Aircraft Development. In *AIAA AVIATION 2022 Forum: June 27-July 1, 2022, Chicago, IL & Virtual* Article AIAA 2022-4052 (AIAA AVIATION 2022 Forum). American Institute of Aeronautics and Astronautics Inc. (AIAA). https://doi.org/10.2514/6.2022-4052

AIAA AVIATION Forum
June 27-July 1, 2022, Chicago, IL & Virtual
AIAA AVIATION 2022 Forum

10.2514/6.2022-4052

Advancing Cross-Organizational Collaboration in Aircraft Development

Erik H. Baalbergen[1] and Jos Vankan[2]
Royal Netherlands Aerospace Centre, Amsterdam, The Netherlands

Luca Boggero[3] and Jasper H. Bussemaker[4]
German Aerospace Center (DLR), Institute of System Architectures in Aeronautics, Hamburg, Germany

Thierry Lefèbvre[5]
ONERA/DTIS, Université de Toulouse, Toulouse, France

Bastiaan Beijer[6]
KE-works, Delft, The Netherlands

Anne-Liza M.R.M. Bruggeman[7]
Delft University of Technology, Delft, The Netherlands

Massimo Mandorino[8]
Università degli Studi di Napoli Federico II, Napoli, Italy

Collaboration is a key enabler for the development of modern aircraft and its systems and components. Because of the highly complex and integrated nature of many aircraft systems, effective collaboration requires well-organized, multi-disciplinary, multi-engineer, and multi-organization development processes. These processes require data-driven and computer-supported tools and methodologies. Collaboration may seem as simple as working together, thereby adopting standards and tools, and freely sharing data, information, and knowledge. However, in the development of complex systems such as aircraft, collaboration is not that straightforward. For example, aircraft engineers across disciplines and organizations commonly face challenges such as firewalls, data and tool heterogeneity, and intellectual property protection. In this paper, we review the collaboration challenges, describe how the EU-funded research project AGILE 4.0 addresses these challenges, and detail the application of, and experiences with, AGILE 4.0's collaboration-enabling technologies.

---

[1] Senior Scientist, Collaborative Engineering Systems, Aerospace Vehicles Division, erik.baalbergen@nlr.nl.
[2] Principal Scientist, Collaborative Engineering Systems, Aerospace Vehicles Division, jos.vankan@nlr.nl.
[3] Head of MDO group, Institute of System Architectures in Aeronautics, Aircraft Design & System Integration, Luca.Boggero@dlr.de.
[4] Researcher, MDO group, Aircraft Design & System Integration, Hamburg, Jasper.Bussemaker@dlr.de.
[5] Research Scientist, ONERA/DTIS, Université de Toulouse, Thierry.Lefebvre@onera.fr.
[6] Senior Project Engineer, KE-works, bastiaan.beijer@ke-works.com.
[7] PhD Candidate, Faculty of Aerospace Engineering, A.M.R.M.Bruggeman@tudelft.nl, AIAA Student Member.
[8] PhD Candidate, Department of Aerospace Engineering, massimo.mandorino@unina.it.

I. Introduction

Aircraft have grown over the past century from experimental and spectacular “flying machines” into large and complex, but safe and reliable transportation vehicles. In line with this growth, development of aircraft has evolved from a trial-and-error activity carried out by one or two persons in a work shed into a large collaborative effort that involves many engineers from multiple disciplines, working at different organizations in various countries, and employing digital tooling for huge and dispersed data sets. This evolution is still ongoing, driven by the expectation that airlines will increasingly demand new aircraft to replace older, less fuel-efficient aircraft [1]. The aircraft industry and its supply chain must constantly innovate to respond to the growing demands from airlines for innovative, cost-efficient and high-performance aircraft, on increasingly shorter development timescales. Meanwhile, the industry has to deal with challenges of global competition, fluctuations in the economy, accruing regulatory oversight, growing scarcity of non-renewable energy sources and materials, and more dynamic staff turnover.

Improved collaboration effectiveness and efficiency between original equipment manufacturers (OEMs), as well as their suppliers and other stakeholders, is seen as an important enabler for dealing with some of the aforementioned challenges throughout the aircraft development lifecycle. Efficient collaboration among multidisciplinary engineers, specialists and parties is a prerequisite in the design, construction, operation, and maintenance of modern and innovative aircraft [2, 3]. Efficient multidisciplinary collaboration sounds trivial but is certainly not straightforward given the aforementioned challenges [4, 5]. Collaboration is often impeded by national, organizational, human and technical barriers [6]. National barriers include national laws and export regulations, such as the US-enforced International Traffic in Arms Regulations (ITAR) [7], and impose heavy security-related restrictions on international collaboration. Organizational barriers evolving from regulations or lack of trust amongst competitors include: complex work instructions, methods, and tooling for the “daily job”; physical and employment contractual impediments and security constraints to protect resources and intellectual property; and having to deal with remotely delegated responsibilities. Human barriers include diversities in culture and language, and heterogeneous ways of working. Technical barriers commonly resulting from national, organizational, and human barriers, include incompatibility and heterogeneity among tooling, data, and information, and IT security measures such as firewalls and proxy servers.

Traditional technical solutions enabling specialists to remotely communicate and exchange information have been available for several decennia, such as telephones, electronic mail, software for file exchanges and sharing, teleconference facilities, and distributed as well as remotely accessible central data sharing and tooling. Despite these traditional solutions, effective collaboration still faces many of the barriers mentioned above. Technical solutions addressing national laws and international regulations include the labelling of data and filtering of mail messages and data exchanges based on the labelling, but raise the additional technical challenge of how to actually apply labelled data for engineering activities. Organizational barriers cannot be avoided nor circumvented, so typically arrangements are made through legal agreements and contracts to allow and enable engineers to collaborate across the organizational borders, thereby adopting collaboratively agreed ways of working using specific collaboration-enabling tools. The arrangements may indeed permit engineers to collaborate, yet engineers still face the technical barriers resulting from the extensive IT protection measures taken by the collaborating organizations. Facing human and technical barriers generally involves dealing with a diversity of specialist languages, tooling, and ways of working, establishing some form of interoperability among remotely dispersed and heterogeneous tools and data, whilst retaining compliance with regulations, organization-specific constraints, and security rules. These combined challenges require specialized attention in setting up successful cross-organizational collaboration schemes – ones that also work fluidly, transparently, and efficiently for engineering staff, with a minimum of organizational or technical overhead.

The topic of collaboration across disciplines, organizations and countries has been addressed throughout the past decades in aerospace and engineering areas, e.g., [2, 8, 9, 10]. Research and developments have resulted in several initiatives, standardization efforts, and partially also envisioned holistic solutions that support collaboration, in addition to contracts, agreements, and information technology (IT) facilities and developments. Examples are the Transglobal Secure Collaboration Program (TSCP) addressing the sharing and exchange of data in collaboration programs in aerospace and defense that span national jurisdictions [10], BoostAeroSpace [11] (”the Standard European Cloud providing secure collaboration services and business process integration throughout the extended value chain of the Aerospace and Defense industry”) and GAIA-X [12] (“the next generation of data infrastructure: an open, transparent and secure digital ecosystem, where data and services can be made available, collated and shared in an environment of trust”). These initiatives are highly focused on the legal aspects of data exchange and generic IT and cloud services. The study described in this paper focuses on other collaboration aspects in several representative application cases from the aeronautical supply chain, that directly improve how effectively and efficiently engineers are able to perform their daily work, alongside – and in spite of – any imposed legal constraints and IT restrictions.

Barriers impeding collaboration have been experienced, and therefore also addressed, in the context of several EU-funded aerospace research projects throughout the past decades, such as VIVACE[9] [13], CESAR[10] [14], CRESCENDO[11] [15], TOICA[12] [16], and IDEaliSM[13] [17]. The EU-funded research project AGILE[14] (2015-2018) focused on setting up and executing collaborative, distributed, Multi-disciplinary Design Analysis and Optimization (MDAO) paradigms for aircraft design. A detailed overview of the collaboration challenges addressed in AGILE is included in [6, 18]. The EU-funded research project AGILE 4.0 continues the research and work started in AGILE with an extended scope of the development process. AGILE focused mainly on the system design and optimization, i.e., the MDAO part of the development process. AGILE 4.0 has introduced additional upstream activities inherent to a typical (Model-Based) Systems Engineering approach, such as stakeholders and needs identification, definition of technical requirements, and generation of system architectures. This extension in the original AGILE scope entails additional challenges affecting the collaborative system development. Those challenges are addressed in section III.

In this paper, we start with a short introduction to the AGILE 4.0 project. Next we describe how the project responds to the various aforementioned collaboration challenges in terms of a framework that provides teams of remotely dispersed engineers with an integrated set of specific technologies to facilitate efficient collaboration. We finally describe the application of the framework and the experiences gained in several application cases in AGILE 4.0 context. The application cases use the AGILE 4.0 framework implementation to facilitate collaboration amongst various partners from academia, research institutes, and aircraft OEMs and their suppliers.

II. Introduction to AGILE 4.0

As stated in the Introduction, the challenges of collaborative aircraft development have been researched and addressed by the authors for several years, in the context of multiple projects. One of the projects, the EU-funded H2020 research project AGILE [19] has introduced a novel paradigm and new technologies for cross-organizational collaborative MDAO processes, with the main result being acceleration of the setup time of MDAO development systems, reducing the time needed by more than 40% compared to conventional MDAO approaches.

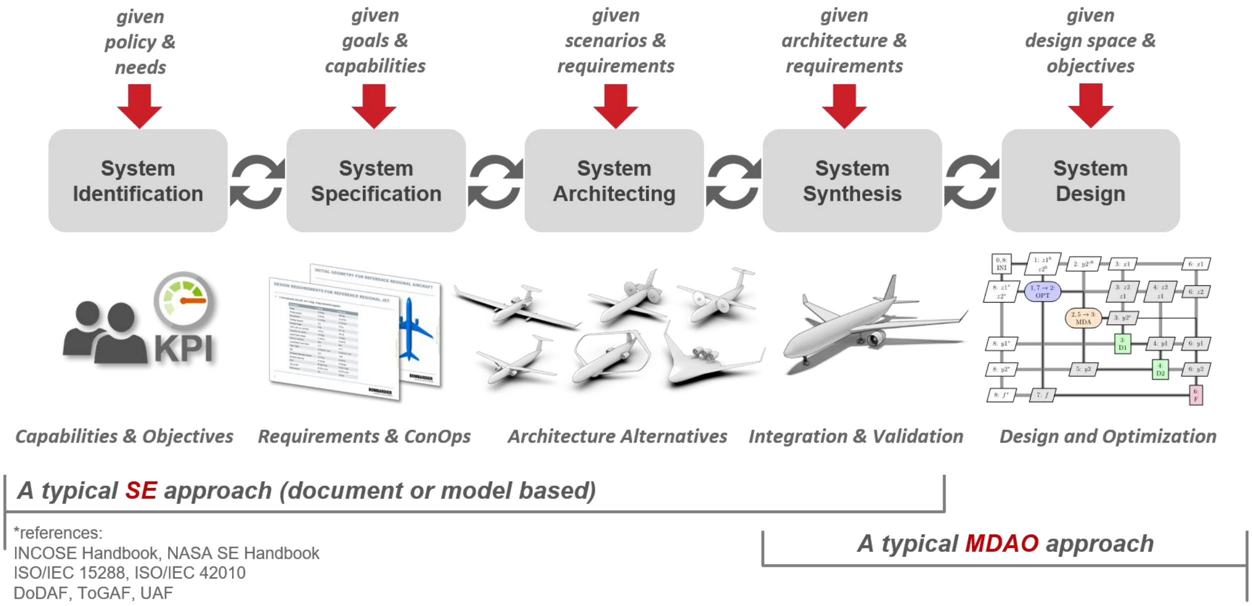


Fig. 1 Systems Engineering Product Development process set up in AGILE 4.0 for the collaborative development of complex aeronautical systems (adapted from [20]).

---

[9] Value Improvement Through A Virtual Aeronautical Collaborative Enterprise

[10] Cost Effective Small AiRcraft

[11] Collaborative and Robust Engineering using Simulation Capability Enabling Next Design Optimization

[12] Thermal Overall Integrated Conception of Aircraft

[13] Integrated & Distributed Engineering Services framework for MDO

[14] Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts

The scope of the original AGILE project [21] was limited to the design and optimization of aircraft systems, for a given set of design requirements, and for a given system architecture. That scope is being expanded in the follow-up EU-funded H2020 AGILE 4.0 project, where upstream Systems Engineering phases are included in the system development cycle, in addition to the usual design and optimization activities. This extension in scope is represented in Fig. 1, which depicts the Systems Engineering Product Development process set up in the AGILE 4.0 project [20]. The last two steps of this process represent the development activities investigated in the AGILE project and deal with the formulation and execution of MDAO processes for the design and optimization of aeronautical products. The other preceding steps of the Systems Engineering approach represent the extension in scope introduced by AGILE 4.0. Those steps include the definition of stakeholders, needs, and requirements [22], the generation of multiple alternative system architecting to be designed and optimized [23], and the automatic verification of requirements [24].

The ambition of AGILE 4.0 is to realize a significant reductions in aircraft development costs and time-to-market through the implementation of an integrated "cyber-physical" aeronautical supply chain, from integrators and high-tier suppliers to SMEs, leading to innovative and more sustainable aircraft products. In particular, AGILE 4.0 targets the digital transformation of the main pillars of the aeronautical supply-chain, including design, production, certification, and maintenance. In order to achieve that target, fifteen partners from industry, research organizations and academia – located in Europe, Canada, and Brazil – are involved in the project. More information about the AGILE 4.0 project can be found in reference [25].

AGILE 4.0 builds upon the MDAO technologies developed in the AGILE project, but introduces new methodologies and implementations that leverage a Model-Based Systems Engineering (MBSE) approach. The MBSE approach supports the main activities of a Systems Engineering Product Development process, through modelling. Under the MBSE paradigm, the model *is* the specification. Therefore, it entails modelling of development scenarios, stakeholders involved, needs and requirements; modelling of the aircraft architecture; modelling of the requirement verification methods; and modelling of the decision-making, verification, and validation processes. The MBSE and MDAO technologies developed under AGILE 4.0 are implemented in an Operational Collaborative Environment (OCE). The OCE enables the collaborative development of complex aeronautical products through the support of modeling and a Systems Engineering approach in all development phases. A schema of some of the MBSE and MDAO technologies as part of the OCE and used during all steps of a Systems Engineering Product Development process, is depicted in Fig. 2.

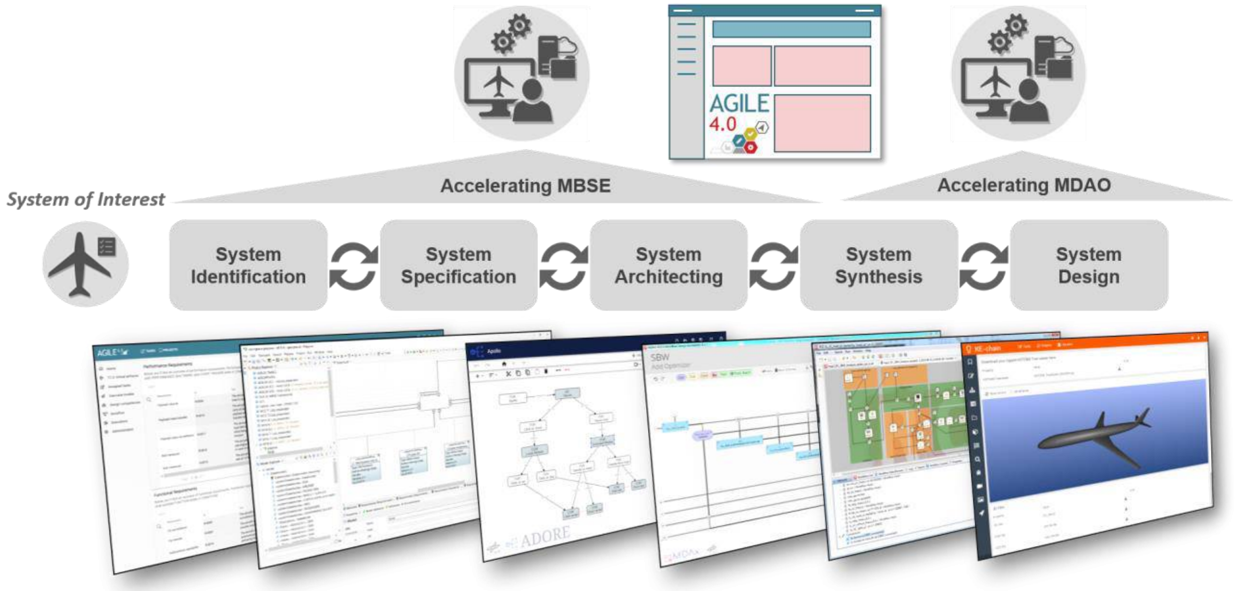


Fig. 2 Collaborative MBSE and MDAO technologies developed in AGILE 4.0 to address all the steps of a Systems Engineering Product Development process (adapted from [20]).

The extensions in scope envisioned by AGILE 4.0 – in particular the inclusion of Systems Engineering upstream activities – exposed new technical challenges affecting collaboration across multiple partners. The main collaboration challenges in MBSE and MDAO encountered in AGILE 4.0 are presented in section III. The solutions provided using the new MBSE and MDAO technologies developed in the project are described in section IV. Integrated application of the solutions through the OCE in example application cases from AGILE 4.0 are described in section V.

III. Collaboration Challenges in AGILE 4.0

This section describes the collaboration challenges in AGILE 4.0 emerging from four key functionalities: (1) integrated access to the OCE; (2) collaborative systems engineering; (3) tool integration in cross-organization collaborative workflows; and (4) surrogate modelling.

A. Integrated Access to the OCE

An important challenge in AGILE 4.0 concerns collaborative access to the OCE, thereby organizing and giving integrated access to expert knowledge, competences, technologies, data, and information dispersed across engineering divisions in different organizations. The main challenge is providing collaboration engineers with access to the OCE throughout the Systems Engineering development process whilst following an MBSE approach, with the aim of achieving (cross-organization) MDAO. This challenge includes providing seamless access to different data standards for modelling information, and managing different expert knowledge domains and domain-specific tooling. Within the team of remote engineers, each engineer must have a dashboard through which he or she can fulfill his or her role within the development process. A graphical user interface must provide each expert with access to the relevant competences, technologies, data, and information whilst hiding complexity traditionally involved with using each domain-specific tool separately. For example, data and tooling connections should be transparent and automated, giving the user an intuitive graphical user interface to steer the design and development process easily and fluidly.

B. Collaborative Systems Engineering

As stated in section II, one of AGILE 4.0's ambitions is to extend the process for the development of complex systems, by including typical upstream Systems Engineering activities such as requirements definition and system architecting before deploying and operating collaborative MDAO processes. This extension entails the generation of new information collaboratively produced by different people with different roles within the dispersed development team, which includes various distinct roles such as systems engineers, requirement engineers and system architects. The elements of the design information are strongly interrelated. For example, system requirements are elicited according to stakeholder expectations, while requirements drive the architecting and design of the system [22]. Therefore, all relevant design information must be available to the entire development team and always be up-to-date, without inconsistencies or duplication. In other words, one of the collaboration challenges tackled by AGILE 4.0 concerns collection and sharing of design information through a Single Source of Truth (SSOT). In this context, an MBSE approach should ideally enhance the collaboration among the experts of the development team. In fact, when the SSOT is in the form of models instead of documents, the design information can indeed be more effectively shared among multiple partners having different tasks in the development process. Therefore, a second significant ambition of the AGILE 4.0 project is the transition from document-based to model-based design, supporting the previously mentioned activities of a Systems Engineering process. This presents an additional challenge to the project: development of technologies that enable and facilitate the concurrent modelling of design information.

A key phase of the Systems Engineering development process is that of system architecting. This phase builds on use cases (how the system is going to be used by actors) and the accompanying functional architecture (system functions and their relations), as defined in the online collaborative environment. System architectures are created by first modelling an architecture design space based on the Architecture Design Space Graph (ADSG) [26], and next using this model to generate architectures in an optimization context. The ADSG represents architectural elements (such as functions, components, decompositions, ports) and their connections, and is created and manipulated using a Graphical User Interface (GUI). Furthermore, to enable analysis and design of generated architectures using collaborative workflows, the elements in the ADSG must be linked to the MDAO workflow. The challenge to remote collaboration lies in how to store and synchronize model data being worked on in a collaborative engineering project (SSOT), how to make the GUI available to all project users with as little extra steps as possible, and how to connect the elements in the ADSG to other upstream (i.e., systems engineering) and downstream (i.e., MDAO) model artifacts.

C. Tool Integration in Cross-Organization Collaborative Workflows

As described, the central AGILE 4.0 environment facilitates the definition of collaborative MDAO studies in the form of collaborative workflows involving the competences and resources of the collaborating – but possibly organizationally and geographically dispersed – teams of engineers. The next main challenge is to actually implement collaborative MDAO studies by execution of collaborative workflows. Experiences in past collaboration projects revealed that locating all scattered competences, tools, resources, data and other information required for running such workflows at a central hub, and having all engineers working in, and operating their specialist tools, in the central hub environment is generally not a practical option. Instead, the partners wish – or are even forced – to perform their share of the collaborative study at their own premises, enabling them to protect, and control the usage of, their own resources, intellectual properties, and proprietary tools and data.

The specific and commonly IP-protected resources required for running a partner's share of the collaborative workflow are commonly available on the partner's premises and IT infrastructure only. As a result, a collaborative MDAO workflow usually spans the IT networks of the collaborating engineers, and hence crosses the borders of their organizations. This distributed set-up, however, raises several practical collaboration challenges, including connecting the variety of tools at data level; orchestrating the distributed tools according to the collaborative workflow – thereby keeping the tool and resource owners in control and letting the data flow in a secure way among the tools despite their locations – and supporting successful execution of the collaborative workflow despite offline partners in different time zones, network hiccups, and server overloads due to concurrent use of web-based services.

D. Surrogate Modelling

One drawback of collaborative and distributed MDAO workflows is their natural tendency to increase the time needed to achieve studies compared to fully integrated workflows. Within the AGILE project, several bottlenecks were identified. First, some partners – mainly industrial ones – experience difficulty in sharing their competence because of IP and network accessibility issues. Integrating these competences in an optimization workflow was therefore not achievable. Secondly, to handle more system performance or computation accuracy, more costly competences could be integrated in the workflow, but might lead to intractable increases in simulation time from an optimization efficiency perspective. To overcome such challenges, surrogate modelling technology was applied and developed to enable efficient parametric and optimization studies while protecting IP. This approach relied on a central broker for registration, storage, deployment, sharing, and usage of surrogate models (SMs).

In AGILE 4.0, with the complexity of the workflows that include more than a single design domain, the need of surrogate models and their efficient integration is even more crucial. Development processes should facilitate the creation of surrogate models, their registration for use in the workflows, and access by collaborating engineers. In addition, the technology should allow surrogate modelling competences to be provided by different partners. Increasing the robustness and flexibility of such processes will be an important step towards higher industry adoption.

IV. Collaboration Solutions in AGILE 4.0

The AGILE 4.0 OCE provides an integrated suite of technologies to support the collaborative Systems Engineering and MDAO activities as described in section II. We present in this section the main technologies that support the collaboration aspects, and respond to the challenges presented in section III, organized according to the main functionalities covered in the preceding section.

A. Integrated Access to the OCE

Centralized and integrated access to the OCE is realized using KE-chain [27]. KE-chain features a web-based portal giving heterogeneous teams of experts a user-friendly collaborative platform for organizing and accessing design study data and technologies across the various steps defined in the Systems Engineering Product Development process and among the various collaborating partners. KE-chain provides a central hub for initiating collaborative MDAO studies and for defining MDAO workflows, however it does this without requiring centralization of data storage or domain-specific tooling. KE-chain provides user-authenticated access to design study data, gives access to descriptions of available competences and resources provided by the collaborating partners, and enables integration of engineering services. KE-chain guides engineers through the various steps of the Systems Engineering development process through a dedicated and user-friendly GUI. KE-chain serves as broker for the other OCE technologies and data, thereby hiding as much implementation detail as possible from the engineers so that they can focus on their engineering activities. A schematic overview of KE-chain as front end for the OCE is depicted in Fig. 3. KE-chain gives access to different data standards for the modeling information across the development process (such as SysML, CPACS, CMDOWS), manages expert knowledge and tools such as ADORE, MDAx, KADMOS, and supports the collaborative application of the MBSE methodology with the aim to perform MDAO. These tools are either embedded directly in the KE-chain front end or interact with the KE-chain data model through the Python library Pykechain [28]. OCE tools are integrated as engineering services which can be executed directly on the KE-chain server using Docker containers hosting dedicated Python computational environments.

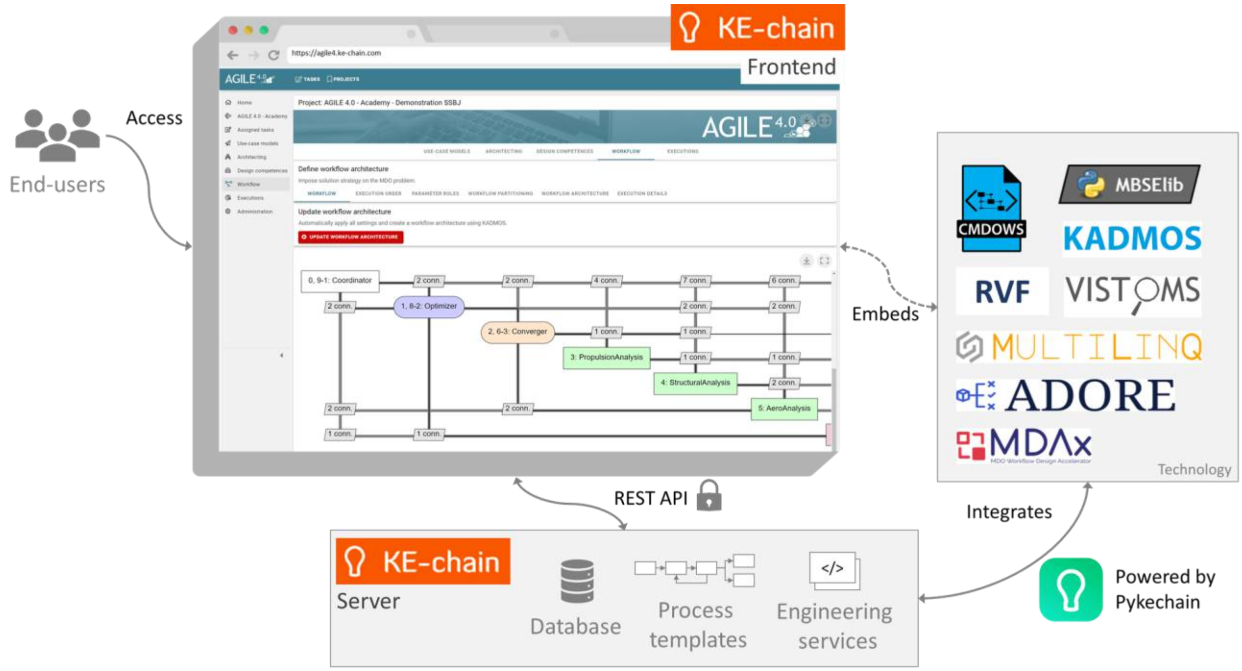


Fig. 3 Schematic overview of the KE-chain front end which embeds and provides access to OCE technology as engineering services.

B. Collaborative Systems Engineering

Several technologies have been developed in AGILE 4.0 to support, improve, and accelerate the development activities performed by multiple experts during the different steps of the collaborative Systems Engineering Product Development process introduced in Section II.

The collaborative definition and modelling of systems, stakeholders, needs, scenarios, and requirements is supported by the GUI of KE-chain. Different people having different roles in the development process (e.g. customers, requirement engineers) can collaboratively author and inspect the data regarding the first two steps of the AGILE 4.0 Systems Engineering process depicted in Fig. 1 using KE-chain. In addition, model validation scripts are integrated in the KE-chain back end in order to automatically identify possible errors, missing details and inconsistencies in the information produced by different members of the team.

The intermediate part of the AGILE 4.0 Systems Engineering Product Development process focuses on the generation of multiple alternatives for system architectures that fulfil all the needs demanded by the stakeholders. System architectures are created by multiple people through the modelling of an architecture design space based on the ADSG. The GUI for editing the ADSG is implemented in a tool called ADORE [29]. Its graphical user interface is implemented using web technologies to ensure that the back end (the Python code implementing the ADSG) and the front end (the browser-based interactive GUI code for manipulating the ADSG) do not have to reside on the same server. This enables the use of the tool and method without any need for local installation. Beyond mere user convenience, this also serves to protect IP rights and enables accelerated release cycles.

ADORE has been deployed at a central webserver. Connection to the OCE implemented in KE-chain is established using the aforementioned Pykechain Python library. Independently from the main code of ADORE, an adapter interface is created to implement authentication and project-data management via the KE-chain platform. This enables users to authenticate themselves using the same credentials they use for accessing the OCE, and to store and load ADORE projects stored as design studies in the OCE. This concept is depicted in Fig. 4. The adapter code enables adding upstream systems engineering elements (e.g. functional requirements, boundary functions) to ADORE projects. Linking upstream elements to elements in the architecture design space model represents the main forward traceability link between upstream system engineering phases and the architecting phase.

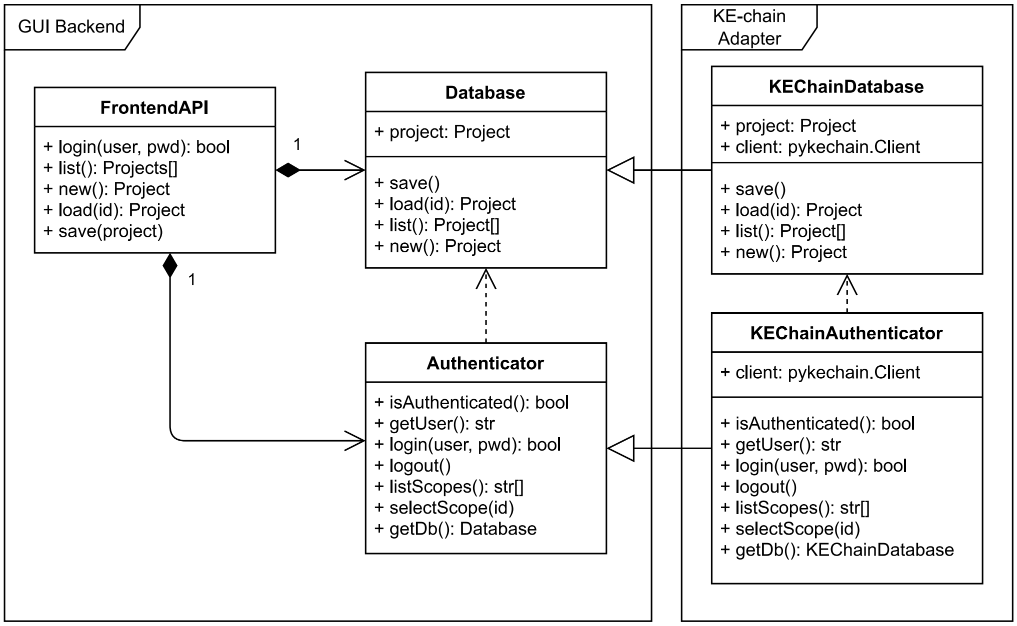


Fig. 4 Notional implementation of KE-chain adapter code for enabling authentication and project-file management through KE-chain for a web-based GUI hosted on a central server: the GUI API uses an Authenticator for managing logins and a Database for managing project files; the KE-chain adapter then specializes these two classes to communicate with the KE-chain server using pykechain.

In the last part of the AGILE 4.0 collaborative Systems Engineering approach, the generated system architectures are designed and optimized. Connecting architecture models created using ADORE to MDAO workflows that enable quantitative analysis of generated architectures is done by associating architectural elements (e.g. components and Quantities of Interest) to nodes in the central data schema (see next subsection). Establishing these links is done using the MultiLinQ tool [29]. Similar to ADORE, MultiLinQ is also implemented as a Python back end with a web-based front end. Its connection to the KE-chain collaborative environment is implemented in a similar fashion as well. This enables the user to seamlessly switch between the architecture design space model and associated architectural elements within the collaborative MDAO workflow. To do this, the KE-chain adapter for MultiLinQ implements functionality to retrieve ADORE models and to import MDAO tool input/output definitions from either the project's tool repository or from previously established collaborative workflows.

The MDAO workflow modeler MDAx [30] has been integrated in the OCE as an alternative solution to KADMOS, an open-source Python package that supports graph-based formulation of large collaborative MDAO studies [31]. MDAx offers a web-based GUI for creating and exploring collaborative MDAO workflows, which can then be exported to a variety of formats including CMDOWS (Common MDO Workflow Schema, an open-source, XML-based neutral language for workflow schemas [32]) and RCE workflows for execution. Similar to ADORE, its Python back end is connected to KE-chain using the KE-chain adapter principle. In addition to storing workflows and related CMDOWS files in the OCE, it also exposes available tool definitions from the tool repository in the OCE in the MDAx user interface: users can directly use the OCE tool repository to start creating workflows. Additionally, a restricted version of MDAx used for creating and editing tool definitions is deployed and connected to the OCE. This restricted version is called MDAx I/O and can still be used together with KADMOS, as it only relates to the tool repository and not to collaborative workflows.

The Requirements Verification Framework (RVF) tool has been implemented in KE-Chain to further support the collaborative systems engineering process [24]. The RVF connects model-based requirements to the different engineering tools in the MDAO workflows and enables the automatic verification of requirements based on MDAO results. Due to the model-based nature of the RVF and its integration with KE-chain, a clear trace from requirements to the MDAO workflow can be generated. This allows collaborating partners to inspect this trace and to better understand the development process. For example, stakeholders can inspect how their needs and requirements have been taken into account in the design process and how requirement compliance is achieved. Furthermore, domain experts can determine which requirements their engineering tools need to verify. The RVF is implemented as a Python-package. KE-chain serves as the GUI for the RVF and interacts directly with the Python code.

C. Tool Integration in Cross-Organization Collaborative Workflows

The AGILE 4.0 OCE provides and integrates several technologies that support the execution of cross-organization collaborative MDAO workflows generated as described in sections IV-A and IV-B, considerably easing the practical collaboration challenges identified in section III-C.

To support connecting the variety of domain-specific tools at data level, the AGILE 4.0 OCE utilizes a common format for data compatibility among engineering tools, the Common Parametric Aircraft Configuration Schema (CPACS) [33]. CPACS is an open-source, XML-based common language for the exchange of product data. It allows storage of parametric definitions of aircraft geometries, as well as analysis results of the individual design disciplines, in a hierarchical structure that facilitates the consistent exchange of product data at multiple levels of fidelity. CPACS is collaboratively developed by, and used within, the aircraft design community. A legacy engineering tool in AGILE 4.0 context is equipped with a wrapper that enables the tool to act on product data and produce analysis results in CPACS format. This "CPACS-ification" of engineering tools facilitates specification (or "declaration") of tools in terms of their semantics at MDAO level. This enables the OCE technologies KADMOS and MDAx to select and automatically integrate partner-provided tools into collaborative workflows supporting the specified MDAO targets, as described in section IV-B above.

The AGILE 4.0 OCE provides several technologies to orchestrate the various tools distributed across the organizations involved and according to the collaborative workflow, while enabling the owners of the tools and resources to remain in control and letting the data flow in a secure way among the tools across disparate locations. Tool orchestration at a single location is basically facilitated through the Remote Component Environment (RCE) [34]. RCE is an open-source environment for the integration of stand-alone tools into automated executable workflows. In the context of the AGILE project, RCE has been equipped with a facility to import workflow definitions in CMDOWS format, allowing automatic creation of local workflows, comprising the local "CPACS-ified" tools.

Local RCE workflows in AGILE 4.0 may be equipped with Brics building blocks that facilitate the composition and execution of workflows across organizational borders in a service-oriented architecture style, while complying with the applicable security constraints and dealing with the security measures of the collaborating partners [35]. Brics provides generic protocols and middleware that can easily be used standalone as well as integrated in any workflow management tool, such as RCE. The middleware orchestrates execution and the data exchange across workflows and services that constitute the collaborative workflow, thereby complying with the ICT security measures of the organizations involved and enabling the respective specialists to remain in full control of their own tools, data and other resources being accessed. To facilitate the secure exchange of data among organizations, the OCE provides a central SharePoint server in a neutral domain that is trusted by the partners.

Brics facilitates the use of "stub tools" in local workflows for calling remote tools and workflows; cf. Fig. 5. A stub tool is a small tool that provides the local workflow with the same interface of the remote tool or workflow in terms of accepting the same inputs and producing the same outputs. Internally, however, the stub tool calls a remote tool or workflow, thereby taking care of the triggering, the synchronization, and the data transfers involved. Brics also facilitates the creation of services and local workflows that may play the role of remote tool or workflow for a stub tool. As such, Brics provides the building blocks for accomplishing a service-oriented setup for distributed workflows. A local workflow executing a tool stub temporarily becomes the *client* (requesting a service). The remote tool or workflow then becomes its *server* (providing a service). In the context of the AGILE and AGILE 4.0 projects, RCE provides Brics plug-ins that enable stub tools to be included in workflows and enable local workflows to be defined as servers.

The basic role of Brics in the service-oriented set-up is depicted in Fig. 5. In this figure, the RCE Brics plugin that represent the stub tool at the client site is labelled with 'C'. Label 'S' at the server site identifies the RCE Brics plugin that handles calls from the stub tool and calls the rest of the server workflow. When the stub tool is activated, it employs the Brics protocol to accomplish the remote tool execution as if the remote tool was part of the local workflows. The numbered arrows in Fig. 5 represent the steps taken to accomplish the remote execution. First, the input files for the remote service are uploaded to the central data server in a neutral domain (1). Next, the remote specialist gets notified (2), who in response starts the remote service (3). The service retrieves the input files from the data server (4), runs the engineering service (5), and uploads the output files to the data server (6). Finally, the output files are downloaded to the client's side (7), and the client workflow continues execution.

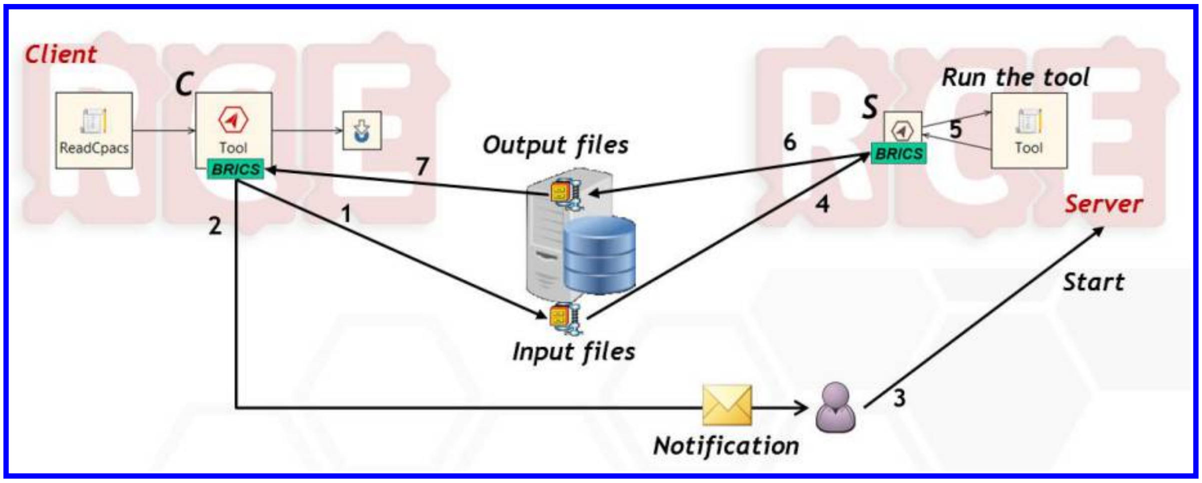


Fig. 5 Schematic representation of the Brics protocol in the context of a client and a server workflow in RCE. The numbered arrows indicate the steps taken to accomplish the remote execution, as described in the text above.

Fig. 5 depicts the simplest possible application of Brics in establishing a distributed workflow: calling a single remote tool, which has been embedded in a dedicated RCE workflow, from within a linear local workflow, with a single input file. Brics however facilitates the construction of more complex situations and has provisions to avoid repeated notification of remote specialists in case of repetitive requests for the same remote service, such as in distributed parameter studies and optimization studies. Brics also provides means for creating automated services in trusted settings, thereby eliminating the need for explicit triggering of remote specialists.

To enable successful execution of the collaborative workflow despite unresponsive partners, failing networks, and overloaded services, the AGILE 4.0 OCE provides the Brics Adapter Service (BAS), a generic server that can relay remote calls. The BAS enables collaborative workflows to be more dynamic in the sense that the identification of a remote service, which usually is hard-coded through the RCE Brics plug-ins in an RCE workflow, can be determined at runtime, based on the input data. From an RCE workflow's perspective, the BAS can be addressed as a remote service via the available RCE Brics plug-ins, but acts as an intermediary agent (a "sub-server") for one or more other remote services. The latter services are referred to as actual remote service(s). The BAS handles requests automatically, without mail notifications nor human intervention. The BAS propagates the remote service request to an actual remote service determined from the input, thereby forwarding the provided input. The BAS can do the propagation either following the Brics protocol or some other protocol. The results of the actual remote service execution are propagated back to the caller of the BAS, as if the BAS has executed the remote service.

As an example, a specific instance of BAS has been used as one of the key links in the chain of executing WhatsOpt surrogate models from within RCE workflows, as described in section IV-D below. WhatsOpt does not fully support neither the CPACS interface nor the default AGILE execution method provided by RCE. As such, WhatsOpt models cannot be integrated into RCE workflows out-of-the-box, without modifying either RCE or the WhatsOpt service. With the BAS in the loop, however, calls to WhatsOpt surrogate models initiated from within RCE workflows and addressing 'bas' as remote service, are propagated to the WhatsOpt service at ONERA, and results are passed back.

D. Surrogate Modelling

The surrogate model technologies developed to tackle collaborative challenges identified in section III.D comprise a surrogate model generator, a surrogate model repository and surrogate modelling competences. The *Surrogate Model Generator* (SMG) is a broker that facilitates the creation of surrogate models, the coordination and registration of this process and related data, and the connection of the actors involved in collaborative MDAO studies. The *Surrogate Model Repository* (SMR) eases the sharing and usage of surrogate models within collaborative MDAO studies [6]. The surrogate modelling competences allow collaborating engineers to efficiently replace design competences or subparts of the MDAO workflows.

A key enhancement provided by AGILE 4.0 embodies the ability to smoothly connect SMR to different providers of surrogate modelling competences in a transparent way from the end-user perspective. As an example, ONERA's surrogate modelling competences relies on the SMT Toolbox [36] and are made accessible "as a service" to SMR through WhatsOpt [37], a web application supporting MDAO collaborative activities.

In order to achieve this new capability, SMR has been extended to facilitate the registration of WhatsOpt surrogate models so they can be executed directly via the SMR user interface as well as indirectly from within RCE workflows utilizing SMR model. The registration of the WhatsOpt surrogate model at SMR is possible from WhatsOpt, making a surrogate model created using WhatsOpt to be available via the SMR as well. The metadata of the surrogate model is provided using a JavaScript Object Notation (JSON) format file including the inputs and outputs information as well as the associated WhatsOpt identifier. A WhatsOpt surrogate model registered with SMR can be selected via SMR (and hence KE-chain) as a competence for use in a study and includes wrappers to facilitate CPACS compliancy. The information is stored in a CMDOWS file and can be imported in KE-chain, thus considered as any design competence in the workflow creation.

When the resulting workflow is imported in the RCE execution platform, the process to execute the WhatsOpt surrogate model requires a remote call by the RCE workflow via the available/regular RCE Brics plug-in for accomplishing remote calls. The remote call is achieved through the BAS, which is described in section IV-C. A specific BAS instance provides WhatsOpt surrogate model calculations “as a service” and runs as a separate service next to SMR. It will recognize the identified WhatsOpt model and propagates execution to SMR. Based on the metadata registered with the WhatsOpt surrogate model, SMR translates the CPACS input file into JSON format used by the WhatsOpt server, calls the WhatsOpt API to delegate the actual execution of the model to the WhatsOpt server, translates the received JSON-formatted output into a CPACS output file, which is subsequently relayed by the BAS instance to the RCE workflow as result from the remote call. As shown on Fig. 6, the WhatsOpt RSM can be accessed through different components on the OCE. This smooth connection of SMR to different surrogate modelling competences provider facilitates the registration of surrogate models from different sources to be used in the workflows.

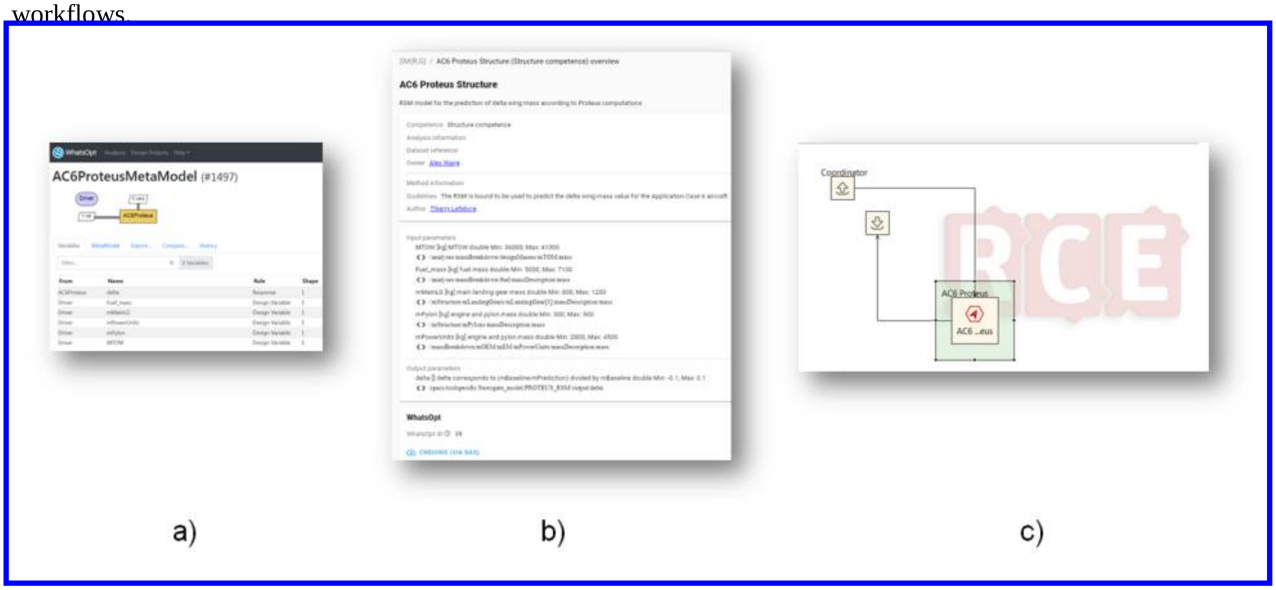


Fig. 6 Representation of the same WhatsOpt surrogate model in three different OCE components: a) WhatsOpt , b) SMR and C) RCE.

V. Application of Collaboration Solutions in AGILE 4.0

The AGILE 4.0 project addresses seven application cases (ACs) as depicted in Fig. 7. The application cases are validated by the industrial partners of the project consortium, and focus on different stages of the aircraft life-cycle, including production, certification and upgrade, thereby using the AGILE 4.0 OCE. A brief synopsis of the seven application cases follows.

A 90-passenger regional aircraft is taken as reference for two application cases focusing on the *Production* phase of the development lifecycle. One case (AC1) addresses the optimization of trailing edge flaps while taking manufacturing aspects into account. The other case (AC2) models and analyzes the production of horizontal tail planes by different industrial supply chains.

Three application cases focus on *Certification*. The main objective of these cases is to address certification aspects during the design process. Conventional and innovative architectures of propulsion and on-board systems are investigated. In particular, one case (AC3) aims at identifying and evaluating multiple system architectures driven by safety constraints. Another case (AC4) focuses on the continuous airworthiness, which mainly defines the maintenance process and aircraft maintainability. Both these cases are based on a 19-passenger regional aircraft. The third case (AC5) addresses part of the certification process of systems and airframe (e.g. electromagnetic compatibility) of Unmanned Aerial Vehicles (UAVs).

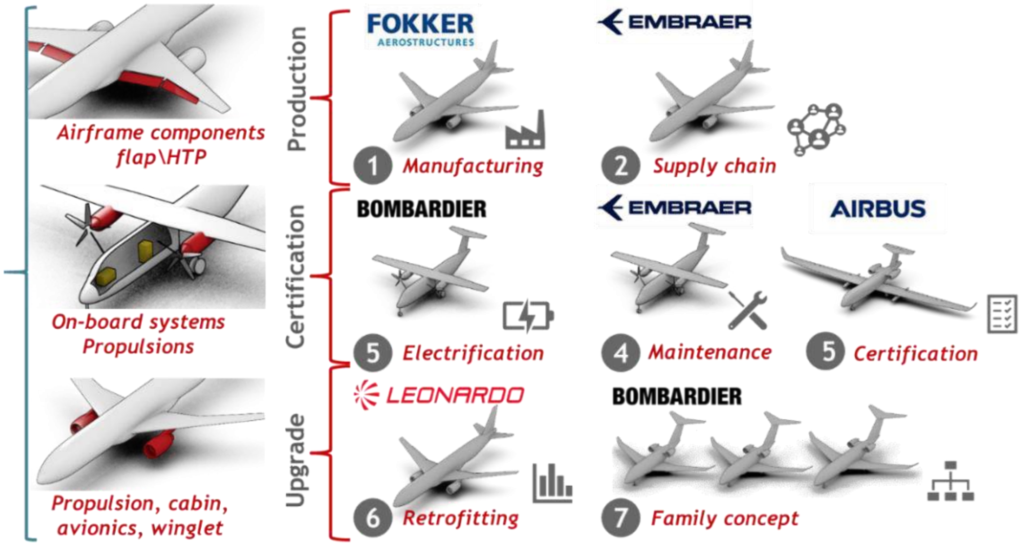


Fig. 7 AGILE 4.0 Application Cases developed through the collaboration solutions addressed in the paper.

The last two application cases address *Upgrade* aspects. The first case (AC6) deals with retrofitting an existing 90-passenger regional jet aircraft, by redesigning and reintegrating novel and improved versions of engines and on-board system architectures. The second case (AC7) tackles the development of a business-jet family composed by three 8-passenger aircraft with different cabin length and design range.

The subsequent subsections describe three application cases in more detail, including the collaboration challenges addressed, the applied AGILE 4.0 collaboration solutions, the experiences gained on how AGILE 4.0 solves the specific collaboration challenges, and any remaining collaboration challenges and issues.

A. Example Application Case: Collaborative Flap Design Optimization for Minimal Weight and Manufacturing Cost

The application case "collaborative flap design optimization for minimal weight and manufacturing cost" (AC1) focuses on the design and optimization of the inboard flap of a regional aircraft. Two flap configurations, as shown in Fig. 8, are compared: the advanced kinematics flap and the dropped hinge flap. The tradeoff focuses on the flap weight versus manufacturing cost. The advanced kinematics flap is characterized by a decoupled rotation and translation. Therefore, the expectation is that this flap is more efficient and therefore smaller than the dropped hinge flap. However, the advanced kinematics flap is more complex and therefore, the manufacturing costs are expected to be higher than for the dropped hinge flap.

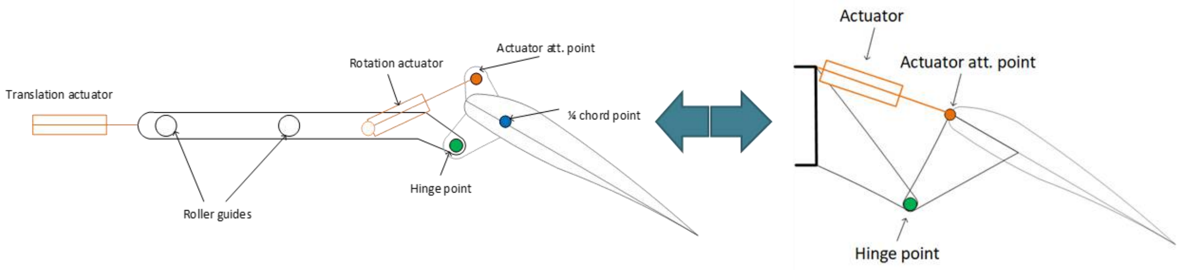


Fig. 8 Two different flap configurations are considered in AC1: the advanced kinematics flap (left) with a decoupled rotation and translation, and the dropped hinge flap (right) [38].

Four AGILE 4.0 partners are collaborating in this AC, as shown in Fig. 9. Each partner provides different analysis tools necessary to enable the specified trade-off. GKN Fokker provides two analysis tools: the Multidisciplinary

Modeler (MDM) [39] and the Cost Analysis Tool for Manufacturing of Aircraft Components (CATMAC) [40]. MDM is a Knowledge Based Engineering (KBE) tool that generates the geometric model of the flap. The MDM has a module called CAD2FEM that generates the FEM model of the flap. CATMAC calculates the manufacturing and assembly costs of the flap. TU Delft provides two analysis tools: Proteus [41], which uses the FEM model from the MDM to size the composite skin of the flap, and the Landing Performance Tool, which calculates the landing distance of the aircraft. NLR provides the AMLoad [42] tool, which calculates the loads on the flap for different load cases. CFS Engineering from Lausanne, Switzerland, provides one analysis tool, the Navier-Stokes Multi Block (NSMB) CFD solver [43], which calculates the aerodynamic performance of the flap.

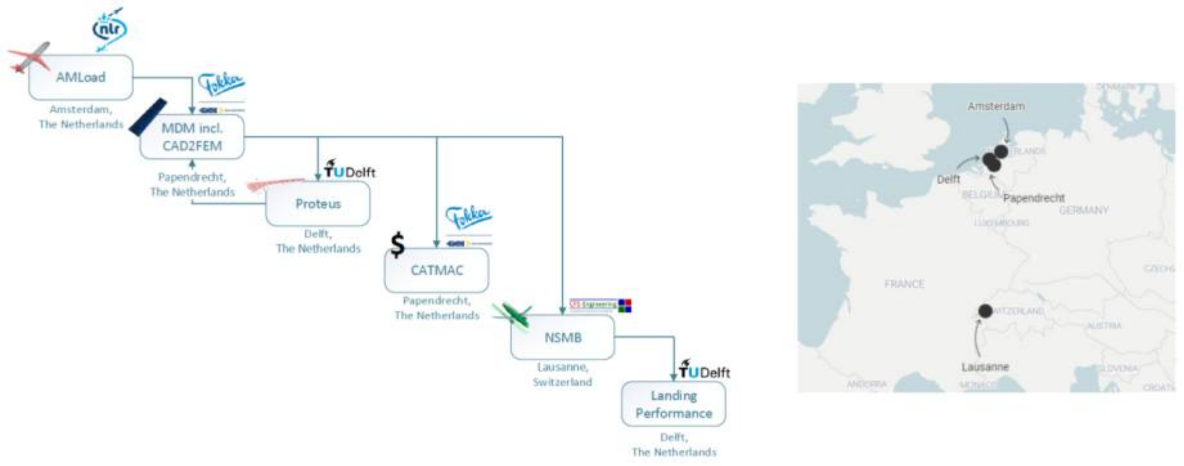


Fig. 9 Overview of the analysis tools used within AC1, including their geographical location.

Several collaboration challenges have been identified within AC1. The biggest challenges are related to the tool integration as described in Section III-C. As explained in Section IV-C, all tools within the AGILE 4.0 workflows need to be "CPACS-ified", meaning that they use CPACS files as input and output. Within the workflow of AC1, the tool Proteus needs a detailed FEM model, which is generated by the tool MDM. However, it is not possible to add this FEM model to a CPACS file. Therefore, multiple files need to be transferred between the MDM and Proteus (both the CPACS file as well as the FEM model files). This challenge has been overcome by extending the Brics functionalities in RCE. Several functionalities have been developed that enable the transfer of multiple files between different servers, allowing high-fidelity tool data - for which CPACS was not sufficient – to be communicated between Proteus and the MDM.

A second challenge experienced within AC1 is related to the consistency of the CPACS input and output files of the different engineering tools. To enable the automatic MDAO workflow formulation, all tools need to have the correct input and output elements as defined in CPACS, such that the correct data is transferred between the different engineering tools. The structure (meaning the type and number of elements) of these files cannot change during MDAO workflow execution. This means that each time the CPACS files changes (e.g. a flap rib is added or removed) all files need to be updated and the workflow needs to be reformulated. This challenge, already identified during the previous AGILE project [44], has partially been solved within AGILE 4.0 using the MDAx I/O editor as explained in Section IV-B. The GUI of the MDAx I/O editor makes it easier to update the CPACS files and keep all CPACS I/O files consistent. This saves a lot of time compared to the manual manipulation of CPACS files that was required before.

Finally, a third collaboration challenge experienced within AC1 related to the NSMB tool. This tool requires some manual steps to run and has a relative long execution time compared to the other AC1 engineering tools. Therefore, it is difficult to integrate the tool in a collaborative workflow as already indicated in Section III-D. This challenge can be solved by generating a surrogate model of the tool. The surrogate model is currently under development using the AGILE 4.0 tools as described in Section IV-D. Integrating the surrogate model in the MDAO workflow will significantly reduce the execution time of the workflow and will make the workflow fully automated and therefore easier to run.

Overall, the collaboration solutions as described in the previous section have made the collaboration within AC1 easier. Due to KE-chain, all partners could access and update the relevant data, while the data are stored in one place. Furthermore, KE-chain, ADORE and the RVF together enabled the inspection of the relations between the different systems engineering elements. Starting from the stakeholders, needs and requirements, through the system architecting and MDAO workflow formulation, to the automatic verification of the requirements. This increased the common understanding of the application case between the partners involved.

Even though several collaboration challenges have been solved, there are still some challenges left within AC1. One of the challenges involves the resilient execution of the workflow. In the case that one of the partners servers fails or gets disconnected, for example when it is accidently restarted, the engineering tool on that server cannot simply be restarted as it has lost track of the number of iterations. In this case, the entire workflow needs to be restarted which can be very inconvenient. This inconvenience is being solved in AGILE 4.0 by improving the Brics functionalities, enabling a remote service to continue a previously interrupted series of iterations.

Another collaboration challenge that arises within AC1 is the installation of the extra tools, such as Brics and RCE, at the partner's servers to enable the execution of the cross-organizational collaborative workflows. This may pose challenges especially for industrial partners who have very strict rules on software installation and security, and usually requires convincing IT departments.

B. Example Application Case: Collaborative Airframe Upgrade Design

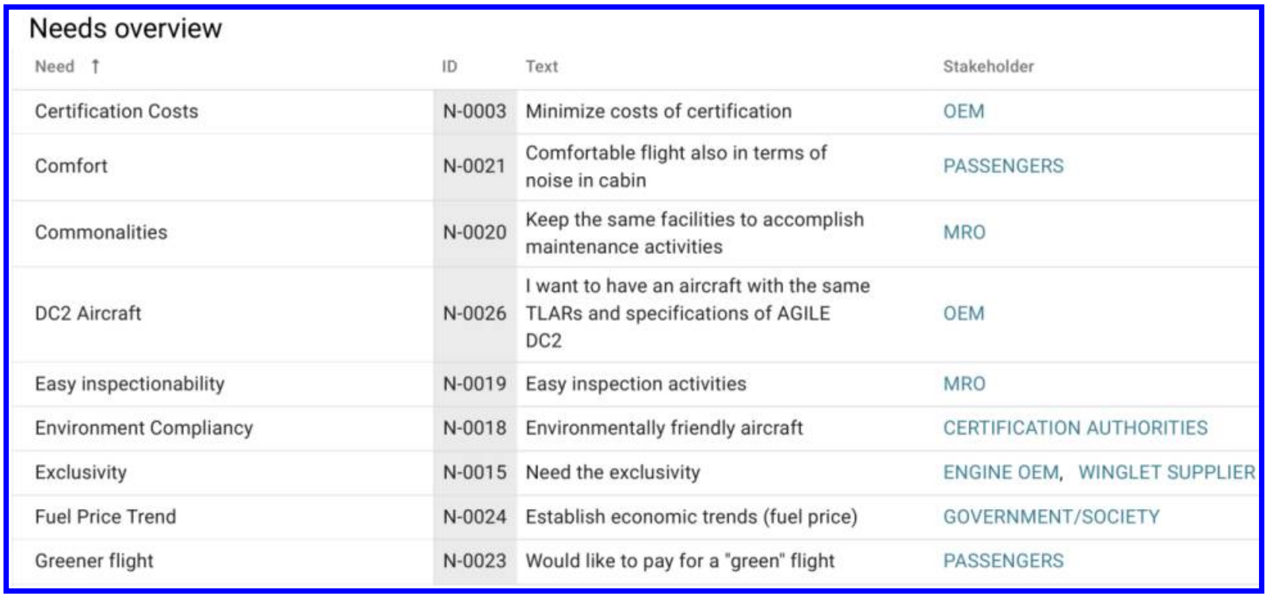
The application case "collaborative airframe upgrade design" (AC6) is focused on a regional-jet 90-passenger aircraft with a design range of 1890 nm. The analysis concerns two retrofitting solutions which can be applied to the reference system. The retrofitting packages considered are: (i) Engine upgrade: High Bypass Ratio (BPR) geared turbofan engines will replace the conventional ones, resulting in fuel consumption, maintenance, noise and air emission improvement. The engines to be installed are designed within the AGILE 4.0 project, their architecture is like Pratt & Whitney PW1000G-series engines and is characterized by a BPR between 9 and 15. (ii) On-Board System (OBS) architecture electrification: three different progressive systems electrifications are considered, obtained through hydraulic and pneumatic system removal. More electric aircraft (MEA) and all electric aircraft (AEA) configuration can bring to weight, fuel consumption and maintenance upgrades. Fig. 10 represents the whole system and highlights the two components targeted by the retrofitting activities.

Fig. 10 AGILE 4.0 AC 6 Aircraft with Engines and OBS highlighted.

Designing an aircraft upgrade is not a straightforward activity. The replacement of a single component involves different phenomena and actors. For instance, it is fundamentally essential to analyze the impact on both performance and cost of such an operation. Performance depends on aerodynamic, structural, and propulsive characteristics of the new solution. Costs are affected by engine, system or aircraft manufacturers, certification authorities, governments, passengers, and airliners. As a result, a wide range of disciplines and stakeholders must be consulted to obtain a coherent and feasible solution. This leads to several challenges facing the designers. The most significant ones are: (i) The huge number of disciplines involved can raise difficulties related to management and schematization of the MDAO workflow required for the analysis; (ii) Each discipline must be executed sequentially with a different tool; The connection of input and output of all the competences could be challenging; (iii) The large number of tools involved during the analysis requires the support of different specialists – each of them will provide their own competences which may be limited by intellectual property constraints; (iv) The huge number of stakeholders and scenarios considered requires an appropriate methodology to account for all actors' needs during the design, and in addition, once the final solution has been defined, a methodology to verify if the requirements are satisfied is needed; (v) High fidelity computations are essential to reach accurate and realistic solutions for each discipline considered, however their exploitation will result in a significant increase in computational time; (vi) Different retrofitting solutions can be achieved, and the best solution will not necessarily be unique, and tradeoff analyses must be performed to understand which solution is more suitable to the scenario and stakeholders needs considered.

These challenges make a collaboration-enabling technology essential to perform such a complex design activity. The remainder of this subsection explains how the OCE has contributed to facing the challenges, showing the benefits of the collaboration solutions illustrated in section IV.

*KE-chain, KADMOS, ADORE***.** KE-chain offers the possibility to schematize and manage all the Systems Engineering and MDAO activities. Thanks to this service, it has been possible to elicit a clear overview of all the stakeholders involved in the scenario and the needs arising from their inclusion in the design process. Fig. 11 shows an excerpt of this schematization.



Needs overview

Need ↑	ID	Text	Stakeholder
Certification Costs	N-0003	Minimize costs of certification	OEM
Comfort	N-0021	Comfortable flight also in terms of noise in cabin	PASSENGERS
Commonalities	N-0020	Keep the same facilities to accomplish maintenance activities	MRO
DC2 Aircraft	N-0026	I want to have an aircraft with the same TLARs and specifications of AGILE DC2	OEM
Easy inspectionability	N-0019	Easy inspection activities	MRO
Environment Compliancy	N-0018	Environmentally friendly aircraft	CERTIFICATION AUTHORITIES
Exclusivity	N-0015	Need the exclusivity	ENGINE OEM, WINGLET SUPPLIER
Fuel Price Trend	N-0024	Establish economic trends (fuel price)	GOVERNMENT/SOCIETY
Greener flight	N-0023	Would like to pay for a "green" flight	PASSENGERS

Fig. 11 Excerpt of AGILE 4.0 AC6 stakeholders and needs.

From an analogous schema, each need can then be converted to a problem role, such as an objective, a design variable, or a constraint using the RVF. In addition, each problem role can be linked to a specific discipline. It is easy to imagine how the management of the connection between all these items would require an adequate system engineering platform. Considering the different retrofitting solutions, which can be carried out separately or in combination with one other, a schematization of the achievable platform solutions helps the designer to understand which components and disciplines are required to analyze each solution. For instance, an electrification of the aircraft OBS will lead to the removal of pneumatic and hydraulic systems and to the installation of a bleedless engine. That means that some components and some sources of fuel consumption need not be included in the overall analysis. This kind of optimization is provided by ADORE, which makes it possible to generate several systems architectures, thereby defining the main differences among them. Subsequently, KADMOS is used for defining all the available competences in the form of tools needed to complete the analysis. Once the required inputs and outputs of each discipline are indicated, an XDSM of the MDAO workflow is automatically generated using the problem roles previously defined. The AC6 schema illustrating all the disciplines involved in the process is showed in Fig. 12.

*RCE, Brics***.** A CMDOWS file has been generated and imported in RCE, allowing automatic generation of an executable collaborative converged design-of-experiments (DOE) workflow. Thanks to Brics technology, each specialist executes his or her own tool remotely, only providing the outcome of the individual computations. The CPACS format allows to store the preliminary and the evaluated pieces of information concerning the analyzed platform in a single output file.

*SMR***.** High-fidelity results are reached through the execution of this workflow. The definition of two surrogate models allowed a reduction from hours to seconds for the execution time of these tools. A response surface model from CFD computations is obtained to estimate the aircraft drag in cruise condition, also accounting for the new installed engine geometry. The second surrogate model is exploited to size the wing structure according to the aircraft masses and loads. Both these surrogate models allow better performance and cost estimations related to the retrofitted aircraft, which are the main objectives of the analysis.

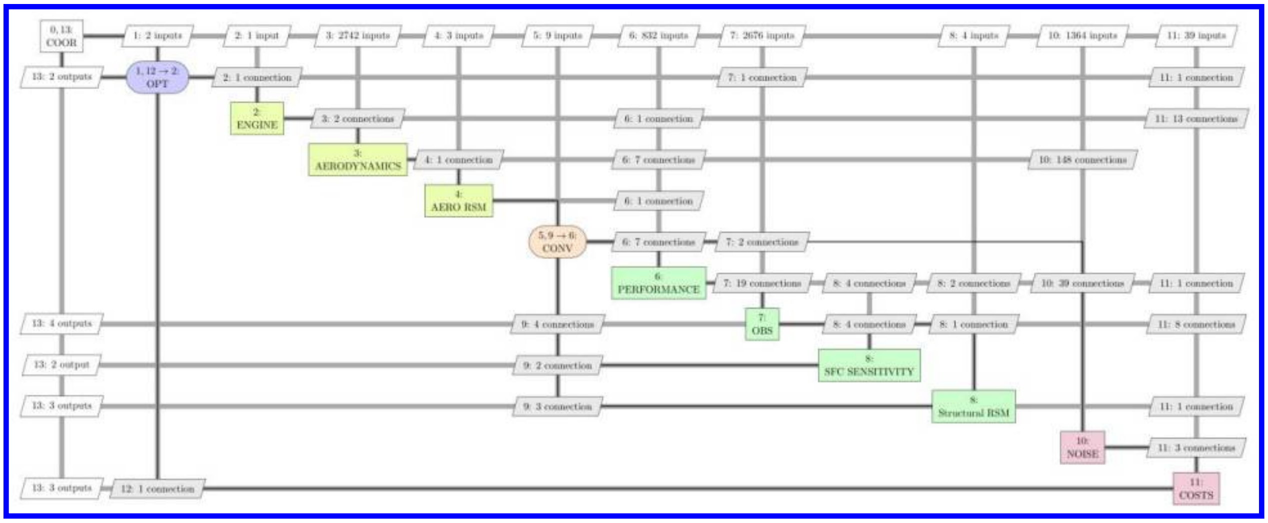


Fig. 12 AGILE 4.0 AC6 MDAO XDSM of the MDAO workflow.

C. Example Application Case: Collaborative Family Concept Design

The application case 'collaborative family concept design' (AC7) deals with the design of a business jet family consisting of three 8-passenger business jets with various design ranges and cabin lengths [45]. The three business jets are designed using an Overall Aircraft Design (OAD) tool in combination with several higher-fidelity analysis tools. Then, family-level non-recurring and operating costs are calculated, forming the basis of the tradeoff: Increasing the number of shared components may reduce non-recurring costs (i.e. save money for the manufacturer), but however may also increase operating costs due to the design of a less efficient aircraft. The optimizer has the choice to share wings, engines, empennage (vertical and horizontal tailplanes), landing gear, and/or on-board system. In addition, for each aircraft, three wing design variables are available: wing sweep, wing thickness-to-chord ratio (t/c), and the rear-spar location as percentage of chord. These design variables together with categorical design variables for selecting which components to share between aircraft make up a hierarchical design problem [26] – for some values of the shared variables for instance, some of the wing design variables might be inactive.

From a collaborative engineering viewpoint, this application case is challenging as it involves coupling many disciplinary tools, multiple aircraft being designed at the same time, and hierarchical design variables. The collaborative MDAO workflow consists of two levels: a family-level and an aircraft-level workflow. The aircraft-level workflow sizes one aircraft at a time, optionally with shared components, whereas the family-level workflow connects the aircraft-level workflows and calculates family-level cost metrics. The family-level workflow starts by initializing the three business jets using the empirical OAD tool. Next, the aircraft design MDAO is executed until convergence, after which the aircraft-level costs are calculated. Finally, family-level metrics are calculated. The aircraft-level workflow starts by initializing the respective aircraft from its previous design and with components shared according to the sharing design variables. Finally, the aircraft-level MDAO is executed where first several high-fidelity tools are executed in parallel, the results of which are then merged together before resizing the aircraft using the OAD tool to ensure that the aircraft design remains consistent. The family-level and aircraft-level workflow models are shown in Fig. 13 and Fig. 14, respectively.

The coupled high-fidelity disciplines include on-board systems design using the ASTRID tool for aircraft on-board systems sizing and tradeoff analysis in initial design, a tool for tailplane sizing and maximum lift estimation, the Proteus tool for wing-mass estimation, and the AMC tool for mission simulation. The tools are provided by different partners. Wing-mass estimation involves optimizing the composite wing structure, taking aeroelastic stability, strength, and buckling constraints into account. One such wing optimization takes about an hour to compute. Therefore, it was not feasible to directly include this tool in the workflow. A surrogate modelling approach is therefore used instead. A DOE was generated varying several wing parameters, the fuselage length and the aircraft MTOW. Together with results from the OAD tool, a multi-fidelity Kriging model was trained and included in the workflow. The workflow uses Brics for remote execution of the ASTRID and the tailplane sizing and maximum lift estimation tools. The remote cost estimation tool is executed using Brics as well. To prepare the AC7 workflow for optimization, however, the execution time of one family design had to be reduced. To keep the execution time under one hour, it was decided to replace the ASTRID and tailplane sizing and maximum lift estimation tools with surrogate models. For the latter, an approach similar to the one applied to wing mass estimation was applied. For the ASTRID tool, however, IP rights constraints were applicable, and therefore it was not possible to create a surrogate model locally. Instead, the SMR service is used to call a surrogate model residing at the partner's computer.

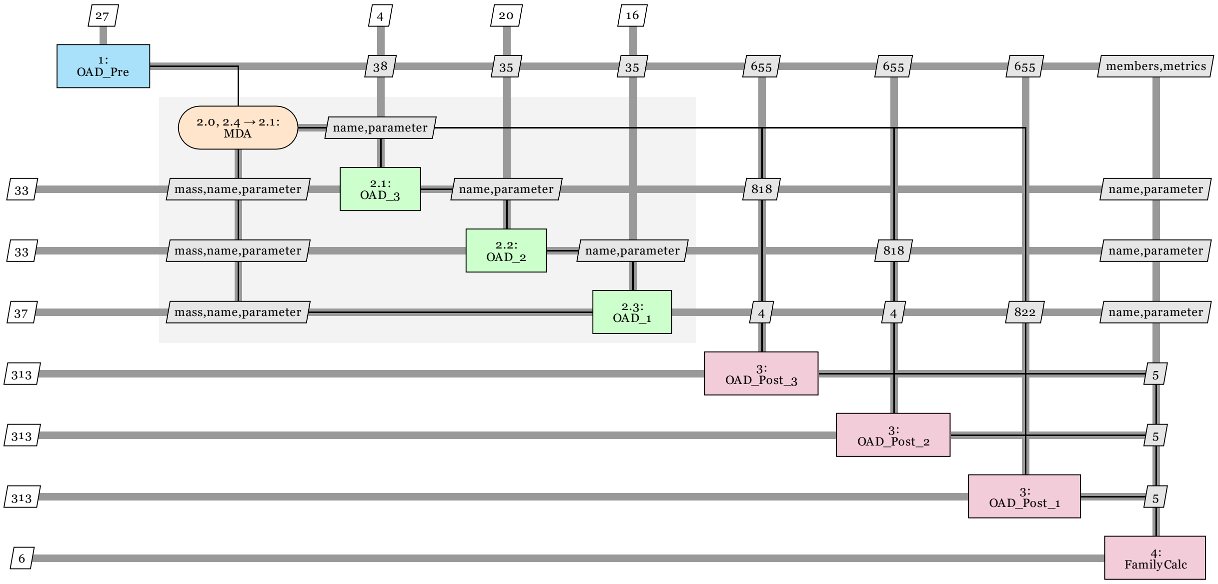


Fig. 13 AC7 XDSM visualizations of the family-level workflow.

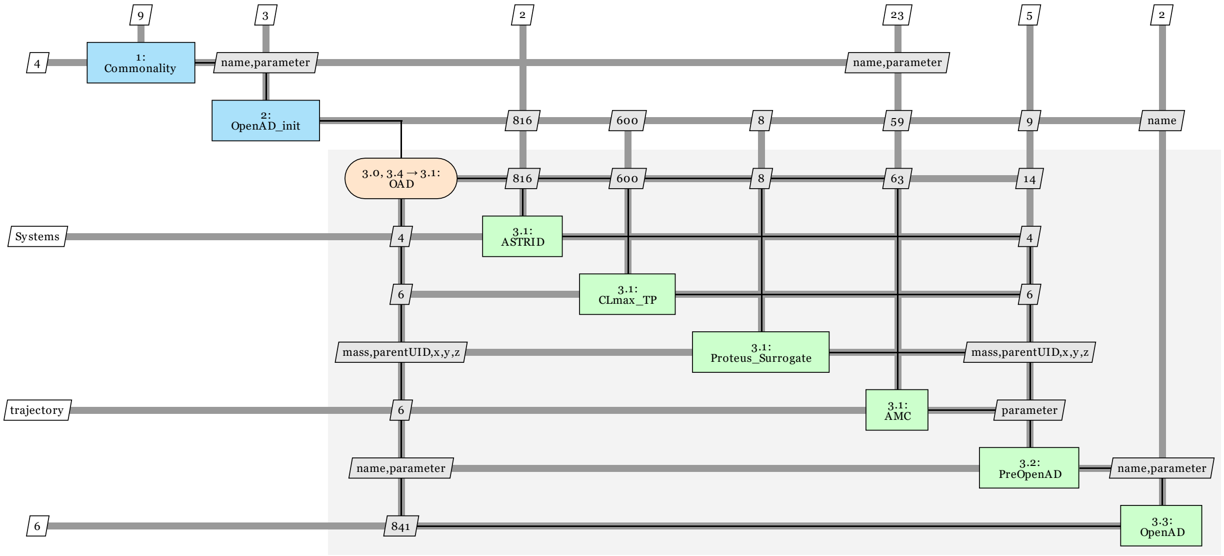


Fig. 14 AC7 XDSM visualizations of the aircraft-level workflow.

Both the family-level and the aircraft-level workflows are modelled using MDAx and implemented in RCE. The RCE workflows are created directly from MDAx export, but require some modifications before they are actually executable: Disciplines had to be replaced either with the appropriate script(s) and tool(s) or by a Brics call; scripts were added to convert between family-level and aircraft-level CPACS files; and converger, output writer, and Brics settings were iteratively checked and corrected. Treating the family-level and aircraft-level CPACS files differently enabled using one RCE workflow to design each of the three aircraft. This approach, however, also introduces several additional challenges: the CPACS file needs to be correctly filtered before being sent to the aircraft-level workflow, and defining tool I/O and creating MDAx and RCE workflows from it is not straightforward.

As far as hierarchical design variables are concerned: This is a common situation in architecture optimization problems [46] and is therefore solved by using ADORE to execute the optimization problem. The architecture design space model is constructed such that wing design variables are linked to each wing component. The decision to use a common wing or not then automatically leads to the deactivation of the wing design variables if a common wing is chosen. Once the DOE or optimization algorithm starts to generate design vectors to be analyzed, ADORE takes care of converting the design vectors to a physical architecture model that can be evaluated. As architectures are evaluated using the MDAO workflow in RCE, a bridge needs to be established. This is done using MultiLinQ which maps an architecture to a CPACS file and back, using rules defined using its GUI. Communication between MultiLinQ and the workflow in RCE is established using Brics: A Brics Python adapter sends the file to the workflow and waits for it to send back a response. Fig. 15 shows this in a sequence diagram.

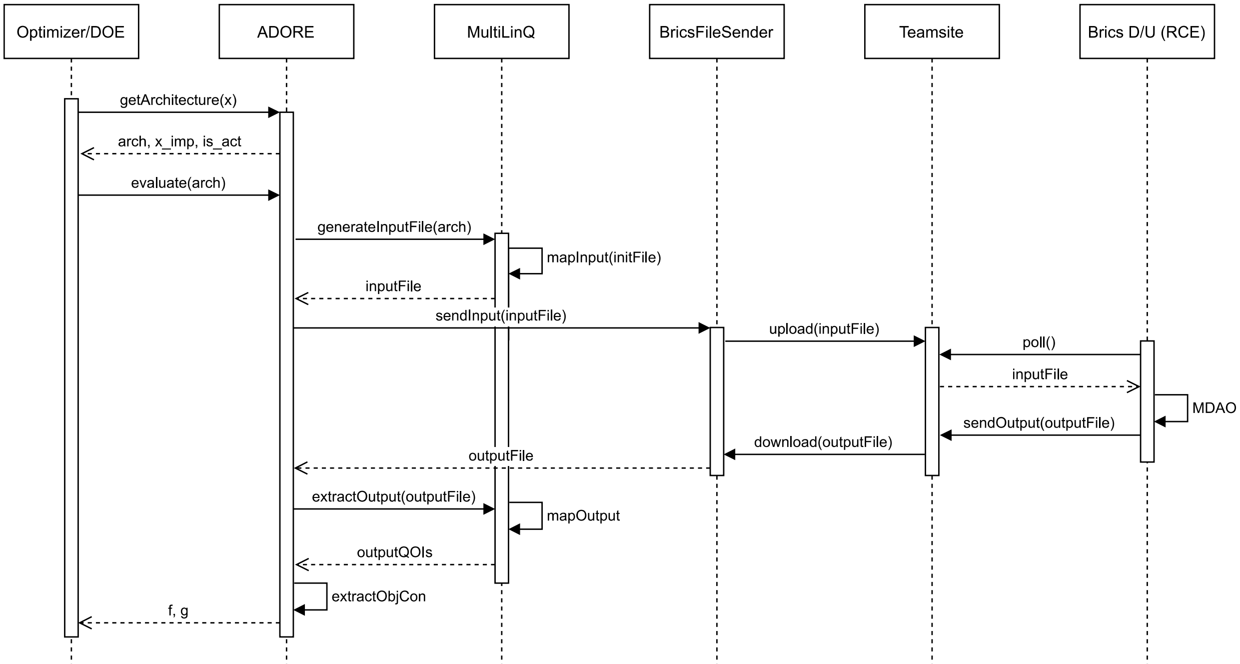


Fig. 15 Sequence diagram showing how one design vector is analyzed: ADORE is used to generate an architecture, MultiLinQ to map the architecture to and from the CPACS file, Brics to send the file to the MDAO workflow in RCE.

Experience gained from the AC7 workflow shows that Brics works well within the context of IP-rights constraints, and in general for remote or cross-environment (Python to RCE) data communication. Compared to the total execution time of the workflow, Brics adds a perfectly reasonable amount of overhead, only about 1 minute of overhead per hour of execution time. The Brics technology is flexible and can be used with more integration environments and neutral data servers than merely what is used in AGILE 4.0. Applying Brics in RCE workflows using the RCE Brics plugins is fairly simple. Using Brics for the construction of collaborative workflows demanding more complicated interactions between the participating partners is possible but requires some extra effort to call Brics from custom scripts, to configure Brics properly, and to test the setup. For example, to start the aircraft-level workflow only once, and to enable it to be called three times from the family-level workflow at the same time, is not possible using the standard RCE Brics plugins without custom scripting.

Creating surrogate models is challenging as there are many decisions to be taken that influence the quality of the model, such as choosing the type of model, hyperparameters, and selecting the DOE points. The collaborative engineering environment cannot assist with such aspects. However SMR turned out to be a useful tool for managing the remotely available surrogate model, thereby assuring protection of related IP.

The family design aspect of AC7 also made for a good test of the CPACS data standard. In most projects and application cases within AGILE 4.0, only one aircraft is designed at a time. Therefore, in the past, whenever a CPACS interface was added to a tool, it would only be able to handle a CPACS file with one aircraft in it. Therefore, in the aircraft-level workflow – where the actual empirical and physics-based aircraft design and analysis takes place – the CPACS file containing the aircraft family is filtered to only contain the one aircraft currently being designed. For this reason, several tool-specific trees were created with their own respective identifiers for filtering, something that is not (yet) supported by the CPACS standard. On the family-level, MDAx is well-suited to create RCE workflows that implement such filtering. However, to achieve that, several custom scripts were needed to automatically modify tool I/O definitions for the three aircraft. Adding such modelling features to the MDAx GUI would be a valuable addition.

Using ADORE together with MultiLinQ to drive the optimization loop works very well, especially when applying newly developed optimization algorithms implemented in Python. Using MultiLinQ to define the logic for mapping a physical architecture to a CPACS file, however, takes quite a lot of debugging and it remains to be seen whether that approach is flexible enough to cover all application cases. Additionally, it is not possible to start an ADORE optimization without having to install ADORE and MultiLinQ locally. In the AGILE 4.0 project this will be solved by letting the tool developers execute the tools and connecting through Brics. In the future, however, it should be possible to start an optimization without any such interaction, preferably controlled through a web GUI. Finally, it is not yet possible to give RCE control over the optimization loop and integrate ADORE and MultiLinQ directly in the RCE workflow.

VI. Conclusions

In this paper, we address the collaboration challenges faced by aerospace engineers across disciplines and organizations in the collaborative development of the next-generation aircraft and its systems and components in the context of the EU-funded AGILE 4.0 research project. We describe several collaboration technologies made available through the integrated framework (OCE), which enables the collaborating engineers to face these challenges. We describe the application of the technologies as well as the experiences gained in several application case studies. The collaboration technologies are already actively applied in other R&D projects that are operating in the same field of collaborative aeronautic design on complementary application cases, as for instance in the EU H2020 project IMOTHEP [47].

Several collaborative systems engineering technologies are implemented by the AGILE 4.0 project. Requirements management is implemented directly in the KE-chain platform using RVF. Architecture modelling is implemented using the ADORE tool. Logic for linking generated architectures to MDAO (i.e. CPACS) is defined using MultiLinQ. Finally, collaborative MDAO workflows are modelled using MDAx and KADMOS. Integrated access to and usage of these tools is facilitated by KE-chain, which acts as a central hub for initiating collaborative MDAO studies and the definition of MDAO workflows. Execution of collaborative MDAO workflows, which potentially span the IT networks of the collaborating organizations, is facilitated through "CPACS-ified" engineering tools, the RCE workflow management tool, and Brics technology for connecting local workflows across organizational borders. The use of surrogate modelling – an essential technology in efficient collaborative design studies and optimizations – is supported using the SMR, WhatsOpt, and BAS tools. The integrated application of collaborative systems engineering technologies is made possible through the AGILE 4.0 integrated framework OCE, and is demonstrated in the context of several application case studies in the AGILE 4.0 project.

The Collaborative Flap Design Optimization application case highlights several collaboration challenges. Thanks to the newly developed technologies from AGILE 4.0, most of these challenges were successfully overcome. New Brics functionalities enable the automatic transfer of high-fidelity models between different disciplines. The MDAx I/O generator greatly reduces the effort to keep CPACS files between disciplines consistent. The surrogate modelling tools resolve the challenges encountered with computationally expensive tools. And finally, KE-chain, ADORE, and RVF together enable the inspection of the relations between the systems engineering elements, increasing the traceability of the design study and increasing the common understanding of the application case between the partners involved.

The Airframe Upgrade Design application case successfully faces a large number of collaboration challenges, thanks to the AGILE 4.0 OCE technologies. During the preliminary design phase, OCE allows considering all stakeholders' needs involved in the process accounting for different scenarios. In addition, such a collaborative environment makes it possible to include all the effects coming from the huge number of disciplines required to obtain a coherent and realistic solution. High-fidelity results can be obtained in drastically reduced computation time thanks to the integration of surrogate models provided via the SMR in the design workflow. Tradeoff analyses to evaluate the best retrofit solution according to stakeholders' needs are hereby proven and within reach.

The Family Concept Design application case introduces several challenges to the collaborative environment. For example, Brics technology is successfully applied in the setup and application of a large and complex collaborative workflow, using custom scripts rather than the ready-to-use RCE Brics Plug-ins. Several disciplines are integrated in the workflow using surrogate models to reduce the optimization workflow runtime. This application case also demonstrates successful application of several collaborative systems engineering tools, such as MDAx for workflow modelling, ADORE for manipulating ADSGs, and MultiLinQ for linking architecture models created using ADORE to MDAO workflows.

In summary, the AGILE 4.0 project demonstrates that collaboration technologies, made available through the integrated framework OCE, successfully support dispersed teams of experts to jointly apply engineering competences and MBSE and MDAO technologies for the collaborative development of complex aeronautical products, able to effectively face the challenges of modern collaborative aircraft development, despite the traditional legal and technical hurdles impeding efficient collaboration.

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.

References

[1] Airbus, "Global Market Forcast 2021-2040," [Online]. Available: https://www.airbus.com/en/products-services/commercial-aircraft/market/global-market-forecast. [Accessed 12 April 2022].

[2] T. Pardessus, "Concurrent Engineering Development and Practices for Aircraft Design at Airbus," in *24th International Congress of the Aeronautical Sciences ICAS*, 2004.

[3] E. Moerland, P.-D. Ciampa, S. Zur, E. Baalbergen, N. Noskov, R. D'Ippolito and R. Lombardi, "Collaborative Architecture supporting the next generation of MDAO within the AGILE paradigm," *Progress in Aerospace Sciences* , vol. 119, 2020.

[4] J. Bruneel, P. D'Este and A. Salter, "Investigating the factors that diminish the barriers to university-industry collaboration, Research Polic," 2010.

[5] E. Baalbergen, W. Lammen, N. Noskov, P.-D. Ciampa and E. Moerland, "Integrated collaboration capabilities for competitive aircraft design," in *8th EASN-CEAS International Workshop on Manufacturing for Growth & Innovation*, 2018.

[6] E. Baalbergen, E. Moerland, W. Lammen and P.-D. Ciampa, "Methods Supporting the Efficient Collaborative Design of Future Aircraft," in *Aerospace Europe 6th CEAS Conference* , Bucharest, Romania, 2017.

[7] U.S. Department of State, Directorate of Defense Trade Controls, "ITAR & Export Controls," [Online]. Available: https://www.pmddtc.state.gov/ddtc_public?id=ddtc_public_portal_itar_landing. [Accessed 3 May 2022].

[8] M. Durstewitz, B. Kiefner, R. Kueke, H. Putkonen, P. Repo and T. Tuikka, "Virtual collaboration environment for aircraft design," in *Sixth International Conference on Information Visualisation*, London, UK, 2002.

[9] A. Kunz, G. Fadel, J. Taiber and M. Schichtel, "Towards Collaboration in Engineering of Tomorrow – Building highly interactive Distributed Collaboration Platforms," in *SAE Technical Papers,* 2005.

[10] TSCP Membership Organization, "About the Transglobal Secure Collaboration Program," [Online]. Available: https://www.tscp.org/about-tscp/. [Accessed 3 May 2022].

[11] BoostAeroSpace Consortium, "Collaborate with your peers," [Online]. Available: https://boostaerospace.com/. [Accessed 3 May 2022].

[12] German Federal Ministry for Economic Affairs and Energy, "Collaboration Platform For The International Development Of Aircraft And Space Vehicles," [Online]. Available: https://www.bmwi.de/Redaktion/EN/Artikel/Digital-World/GAIA-X-Use-Cases/dasc-lab.html. [Accessed 3 May 2022].

[13] E. Kesseler and M. Guenov, Advances in Collaborative Civil Aeronautical Multidisciplinary Design Optimization, American Institute of Aeronautics and Astronautics, 2010.

[14] E. Baalbergen, W. Vankan and A. Kanakis, "A practical approach for coordination of multi-partner engineering jobs in the design of small aircraft," *CESAR Special Issue of Journal Czech Aerospace Proceedings / Letecký zpravodaj, Journal for Czech Aerospace Research,* vol. 3, 2009.

[15] P. Coleman, Innovations in collaborative modelling and simulation to deliver the Behavioural Digital Aircraft. CRESCENDO Forum Participants Handbook, Toulouse: CRESCENDO Consortium, 2012.

[16] TOICA Consortium, "TOICA Final Report - Publishable Summary," 2016. [Online]. Available: https://cordis.europa.eu/docs/results/604/604981/final1-toica-final-report-publishablesummary-v1-0.pdf. [Accessed 3 May 2022].

[17] T. v. d. Berg, B. Beijer and E. Moerland, "Application of an integrated and distributed multidisciplinary product development framework to a multi-tier aircraft design case," in *AIAA Aviation Forum*, 2019.

[18] E. Baalbergen, W. Lammen, A. d. Wit, R. Maas, S. Moghadasi, J. Kos and F. Chiacchio, "Collaborative Engineering Technologies Enabling Multi-Partner Thermal Analysis In Early Design Stages Of Aircraft," in *ECCOMAS Congress 2016, VII European Congress on Computational Methods in Applied Sciences and Engineering*, Crete Island, Greece, 2016.

[19] AGILE Consortium, "AGILE Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts," [Online]. Available: http://www.agile-project.eu. [Accessed 3 May 2022].

[20] P. D. Ciampa and B. Nagel, "Accelerating the Development of Complex Systems in Aeronautics via MBSE and MDAO: a Roadmap to Agility," in *AIAA Aviation Forum*, Washington (US-DC), 2021.

[21] AGILE 4.0 Project Consortium, "AGILE 4.0 - Towards cyber-physical collaborative aircraft development," [Online]. Available: https://www.agile4.eu/. [Accessed 3 May 2022].

[22] 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 Forum*, Washington (US-DC), 2021.

[23] L. Boggero, P. D. Ciampa and B. Nagel, "An MBSE Architectural Framework for the Agile Defintion of Complex System Architectures," in *AIAA Aviation Forum*, Chicago (US-IL), 2022.

[24] A. M. R. M. Bruggeman, B. van Manen, T. van der Laan, T. van den Berg and G. La Rocca, "An MBSE-Based Requirement Verification Framework to support the MDAO Process," in *AIAA Aviation Forum*, Chicago (US-IL), 2022.

[25] P. D. Ciampa, L. Boggero, J. Vankan, T. Lefebvre, B. Beijer, A. Bruggeman, T. van der Laan, M. Fioriti, P. della Vecchia and J. Vos, "MBSE-MDAO agility in Aircraft Development: Achievements and Open Challenges from the AGILE4.0 Project," in *AIAA Aviation Forum*, Chicago (US-IL), 2022.

[26] J. H. Bussemaker, P. D. Ciampa and B. Nagel, "System Architecture Design Space Exploration: An Approach to Modeling and Optimization," in *AIAA Aviation Forum*, 2020.

[27] KE-works, "Optimal digital recording and reporting," [Online]. Available: https://ke-chain.com/en/. [Accessed 3 May 2022].

[28] KE-works, "KE-chain Python SDK," [Online]. Available: https://pykechain.readthedocs.io/. [Accessed 3 May 2022].

[29] J. H. Bussemaker, L. Boggero and P. D. Ciampa, "From System Architecting to System Design and Optimization: A Link Between MBSE and MDAO," *INCOSE International Symposium*, June 2022.

[30] A. P. Risueño, J. Bussemaker, P. D. Ciampa and B. Nagel, "MDAx: Agile Generation of Collaborative MDAO Workflows for Complex Systems," in *AIAA Aviation Forum*, 2020.

[31] I. van Gent and G. La Rocca, "Formulation and integration of MDAO systems for collaborative design: A graph-based methodological approach," *Aerospace Science and Technology*, vol. 90, pp. 410-433, 2019.

[32] I. van Gent, G. La Rocca and M. F. M. Hoogreef, "CMDOWS: A proposed new standard to store and exchange MDO systems," *CEAS Aeronautical Journal*, vol. 9, no. 4, pp. 607-627, 2018.

[33] B. Nagel, D. Böhnke, P. Schmollgruber, A. Rizzi, G. La Rocca and J. J. Alonso, "Communication in aircraft design: Can we establish a common language?," in *28th International Congress of the Aeronautical Sciences*, Brisbane, Australia, 2012.

[34] B. Boden, J. Flink, N. Forst, R. Mischke, K. Schaffert, A. Weinert, A. Wohlan and A. Schreiber, "RCE: an integration environment for engineering and science," *SoftwareX*, 2021.

[35] E. Baalbergen, J. Kos, C. Louriou, C. Campguilhem and J. Barron, "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, October 2017.

[36] M. Bouhlel, J. Hwang, N. Bartoli, R. Lafage, J. Morlier and J. Martins, "A Python surrogate modeling framework with derivatives," *Advances in Engineering Software,* pp. vol.135, p. 102662, 2019.

[37] R. Lafage, S. Defoort and T. Lefèbvre, "WhatsOpt: a web application for multidisciplinary design analysis and optimization," in *AIAA Aviation Forum,* Dallas (US-TX), 2019.

[38] T. van der Laan, B. v. Manen, A. M. R. M. Bruggeman, Z. Wang, D. Peeters, J. Sonneveld and H. Timmermans, "Bringing Manufacturing into the MDO domain using MBSE," in *AIAA Aviation Forum*, Chicago (US-IL), 2022.

[39] T. van den Berg and A. H. van der Laan, "A multidisciplinary modeling system for structural design applied to aircraft moveables," in *AIAA Aviation Forum*, 2021.

[40] T. van der Laan, A. Johman, T. van Puffelen, S. Nolet, B. van Maanen, E. Daugulis and T. van den Berg, "An open source part cost estimation tool for MDO purposes," in *AIAA Aviation Forum*, 2021.

[41] N. Werter and R. Breuker, "A novel dynamic aeroelastic framework for aeroelastic tailoring and structural optimisation," *Composite Structures,* vol. 158, pp. 369-386, December 2016.

[42] D. Rajpal, R. De Breuker, H. Timmermans, W. F. Lammen and F. Torrigiani, "Including Aeroelastic Tailoring in the Conceptual Design Process of a Composite Strut-Braced Wing," in *31st Congress of the International Council of the Aeronautical Sciences*, Belo Horizonte, Brazil, 2018.

[43] Y. Hoarau, D. Pena, J. B. Vos, D. Charbonnier, A. Gehri, M. Braza, T. Deloze and E. Laurendeau, "Recent Developments of the Navier Stokes Multi Block (NSMB) CFD Solver," in *AIAA SciTech Forum*, 2016.

[44] I. van Gent, B. Aigner, B. Beijer and G. La Rocca, "A Critical Look at Design Automation Solutions for Collaborative MDO in the AGILE Paradigm," in *AIAA Aviation Forum*, 2018.

[45] J. H. Bussemaker, P. D. Ciampa, J. Singh, M. Fioriti, C. Cabaleiro, Z. Wang, D. J. M. Peeters, P. Hansmann, P. D. Vecchia and M. Mandorino, "Collaborative Design of a Business Jet Family Using the AGILE 4.0 MBSE Environment," in *AIAA Aviation Forum*, Chicago (US-IL), 2022.

[46] J. H. Bussemaker, N. Bartoli, T. Lefebvre, P. D. Ciampa and B. Nagel, "Effectiveness of Surrogate-Based Optimization Algorithms for System Architecture Optimization," in *AIAA Aviation Forum*, 2021.

[47] P. Novelli, "IMOTHEP: towards hybrid propulsion for commercial aviation: Project Overview," in *AIAA Aviation Forum*, 2021.

This article has been cited by:

1. Luca Boggero, Pier Davide Ciampa, Björn Nagel. An MBSE architectural framework for the agile definition of complex system architectures . [Abstract] [PDF] [PDF Plus]
2. Jasper H. Bussemaker, Pier Davide Ciampa, Jasveer Singh, Marco Fioriti, Carlos Cabaleiro De La Hoz, Zhijun Wang, Daniël Peeters, Philipp Hansmann, Pierluigi Della Vecchia, Massimo Mandorino. Collaborative Design of a Business Jet Family Using the AGILE 4.0 MBSE Environment . [Abstract] [PDF] [PDF Plus]