CEAS Aeronautical Journal
https://doi.org/10.1007/s13272-025-00926-y

ORIGINAL PAPER

The Aircraft Design Laboratory: bridging Germany's universities for collaborative aircraft design and education

Gerrit Pauls[1] **· Volker Gollnick**[1]

Received: 31 May 2025 / Revised: 22 October 2025 / Accepted: 5 November 2025

Abstract

Germany's major aerospace institutes have joined forces to develop a holistic aircraft design environment called UNICADO (UNIversity Conceptual Aircraft Design and Optimization). This environment was created to provide an independent platform that allows universities in Germany and around the world to combine and apply their own design expertise. Although the software itself integrates the knowledge of various partners, users remain limited by their individual levels of experience, which can lead to blind spots in the design process. To address this issue, UNICADO can be used collaboratively when implemented in a so-called Aircraft Design Laboratory (ADL). The ADL is a design facility specifically tailored to the needs of collaborative aircraft design, while also offering a low-threshold entry point for young researchers and aerospace students. This paper presents the methodology behind the ADL concept and its first implementations. Using this approach, an initial educational design network connecting several ADL sites, along with a new didactic graphical user interface, has been established across multiple institutes.

Keywords UNICADO - UNIversity Conceptual Aircraft Design and Optimization · Multi-disciplinary design · Education · Aircraft design

Abbreviations

AAA	Advanced Aircraft Analysis
ADL	Aircraft Design Laboratory
AGILE	Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts
ATLAs	Advanced Technology Long-range Aircraft Concepts
BWI	Behörde für Wirtschaft und Innovation (Hamburg Office of Economics and Innovations)
CPACS	Common Parametric Aircraft Configuration Schema
DLR	Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace Center)
GUI	Graphical user interface
HDMI	High-definition multimedia interface
HTML	Hypertext markup language
IDL	Integrated design laboratory
ILT	Institut für Lufttransportsysteme (TU Hamburg's Institute of Air Transportation Systems)
MDO	Multidisciplinary design and optimization
MICADO	Multidisciplinary integrated conceptual aircraft design and optimization
RCE	Remote component environment
RDSwin	Raymer's design system (on windows)
RWTH Aachen	Rheinisch-Westfälische Technische Hochschule Aachen
SRS	Software Requirements Specification
SSH	Secure shell
TiGL Viewer	DLR's visualization tool for CPACS
TLAR	Top-level aircraft requirement(s)
TU	Technische Universität (University of Technology)
UNICADO	UNIversity Conceptual Aircraft Design and Optimization

✉ Gerrit Pauls
gerrit.pauls@tuhh.de

Volker Gollnick
volker.gollnick@tuhh.de

[1] Institute of Air Transportation Systems, Hamburg University of Technology, Harburger Schloßstr. 22a, 21073 Hamburg, Germany

Published online: 26 November 2025

VAMPzero — Virtual aircraft multidisciplinary analysis and design processes
XML — Extensible markup language

1 Introduction

In projects like VAMPzero, AGILE and ATLAs, the German Aerospace Center (DLR) has shown that conceptual aircraft design is inherently a complex, multidisciplinary task that greatly benefits from integrated, modular software environments [1–3]. Up to this date, monolithic applications such as DarCorp's Advanced Aircraft Analysis (AAA), Dan Raymer's RDSwin or Pacelab (PACE GmbH) dominate the conceptual design [4–6], but they require a great amount of knowledge from the users regarding applied methods and the application itself. Due to the fact that the implementation of those methods is enclosed because they are proprietary software, gain in the fundamental design knowledge is limited. A situation that is not feasible at university level in particular, especially as the level of system understanding required for multidisciplinary design needs to be high [7]. To address this issue, universities initiated their own open-source MDOs like the most commonly known SUAVE project (Stanford University) or OpenConcept (University of Michigan). These MDOs have the capabilities to calculate both conventional and unconventional aircraft designs while providing full access to the codebase. Even though said MDOs are tailored for academic usage, they are dominated by US-based institution. Therefore, the RWTH Aachen initiated a German counterpart called MICADO from which the conceptual design framework UNICADO (UNIversity Conceptual Aircraft Design and Optimization) evolved as part of the UNICADO-I project. In this project, MICADO's design tools were delegated to the German universities TU Berlin, TU Brunswick, TU Hamburg, TU Munich, RTWH Aachen, and Uni Stuttgart reflecting each partner's expertise.

Beside improving design methods and code efficiency, its tools were integrated into the DLR's Remote Component Environment (RCE). RCE is a graphical framework in which other tools and modules can interact with each other to form design chains in so-called workflows [8]. In these, RCE can handle data flows using customized Python scripts or its own software components. There, the former MICADO tools were embedded into a Gauss-Seidel architecture, in which aircraft design tasks are solved in a sequential manner (see Fig. 1)[12]. Additionally, the platform supports remote connections over Secure Shell (SSH) or local networks, which allows participants to share tools as black boxes without exposing internal details. To visualize calculated designs, DLR's TiGL Viewer can be integrated into such a workflow. To do so, a Common Parametric Aircraft Configuration Schema (CPACS) must be generated. Also developed by the DLR, it acts as a distributed, modular aircraft design system aimed at improving the integration of knowledge and models across various institutes and the industry (e.g., Airbus) [1, 9]. Like UNICADO, CPACS uses a node-based XML structure to represent not only the various components of an aircraft but also elements such as missions, airport infrastructure, and network systems. Using an interface to transfer the calculated data into the CPACS format, the UNICADO design loop features these visualization capabilities (see Fig. 1). Beyond that, CPACS has become a widely accepted data exchange file [10], which can be used to communicate with external tools. Therefore, UNICADO's CPACS interface enables external validation and assessment of calculated designs and can potentially inject given designs from outside partners. Through the work carried out by the UNICADO-I consortium, UNICADO

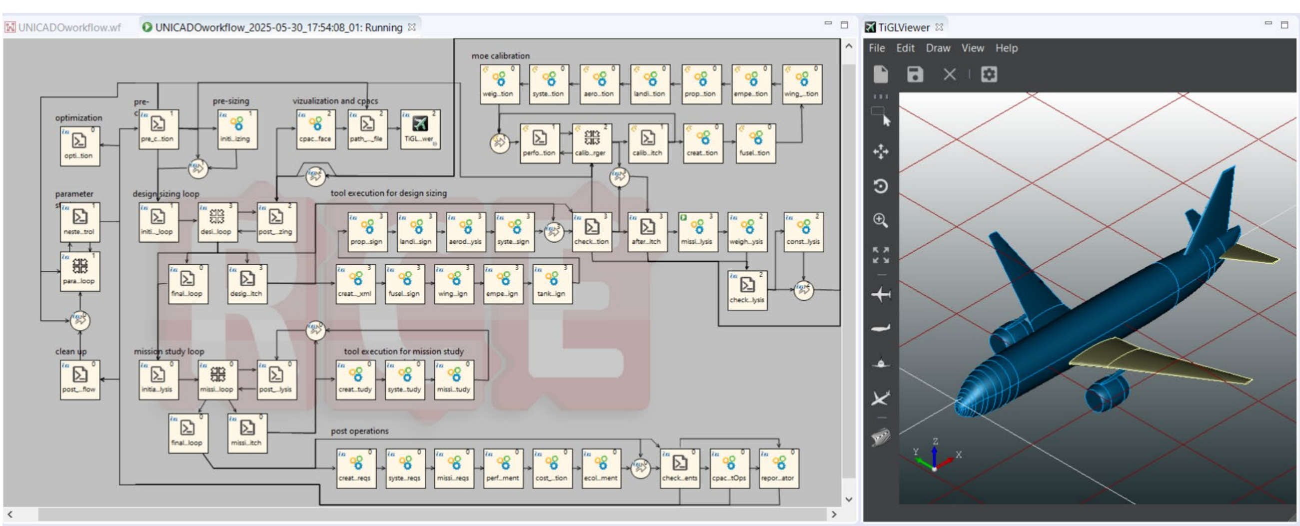


Fig. 1 UNICADO's active RCE workflow (left) updating an integrated TiGL Viewer aircraft visualization for every sizing loop (right)

became a modular open-source platform which fosters collaboration and synergy among German universities in conceptual aircraft design [11, 12]. Building on this foundation, the underlying project of this paper called UNICADO-II (UNICADO-I's successor) seeks to establish a national standard for aviation design tools to support research, education, and industrial collaboration.

To actualize this goal, the UNICADO-II consortium not only wants to offer an intuitive software, but also a suitable (physical) setup in which attendees and online guests can interact well. This shall be achieved by putting the given software into a costume graphical user-interface (GUI) which will also be integrated into a concurrent engineering facility – the Aircraft Design Laboratory (ADL). The idea behind the ADL originated from the Integrated Design Laboratory (IDL) which was built by the former DLR Air Transportation Systems Lab in cooperation with TU Hamburg's Air Transportation Systems (ILT) [13, 14]. The related project iTalent was founded by the Hamburg Office of Economics and Innovations (BWI). The IDL consists of a 6 m×2.5 m video wall, broadcasting interfaces, and a dual-network infrastructure integrated into the work desks around the whole facility (see Fig. 2). The broadcasting interfaces include a webcam (Fig. 3b), two echo-canceling microphones (Fig. 3a) and four stereo boxes in each corner of the room.

Even though the experience gained in over 10 years of existence is crucial [15], some hardware was outdated and interfaces for a permanent collaboration between the UNICADO partners were missing. Therefore, this paper lines out how the IDL and UNICADO's software were adapted to form the ADL. Furthermore, using the initial ADL setup as a blueprint, a nationwide and international concurrent engineering network, enabling simultaneous, collaborative work on aircraft design tasks across institutes is introduced.

2 Methodology

This study focuses on defining the requirements for a scalable ADL framework and tailoring the UNICADO software and the IDL to integrate effectively within it. The initial step involved compiling a Software Requirements Specification (SRS) sheet in collaboration with the UNICADO partners. The SRS, developed in adherence to ISO/IEC/IEEE 29148:2018 standards [16], defines and categorizes requirements into specification types as summarized in Table 1. To handle the multitude of these functional requirements, they are bundled into the non-functional requirements user-friendliness, safety, and interfacing (see subsections 2.3 – 2.2; please mind, that the ADL SRS is a subdocument of the overall UNICADO SRS which results in a nonuniform numbering). In addition, test cases were defined with which the fulfillment of each requirement is checked. A short overview of all functional requirements and corresponding test cases can be found in the Tables 2, 3 & 4. While most hardware-related requirements are tested via user-feedback within the IDL/ADL, software adaptions like GUI development and server setup are overseen by the issue system of

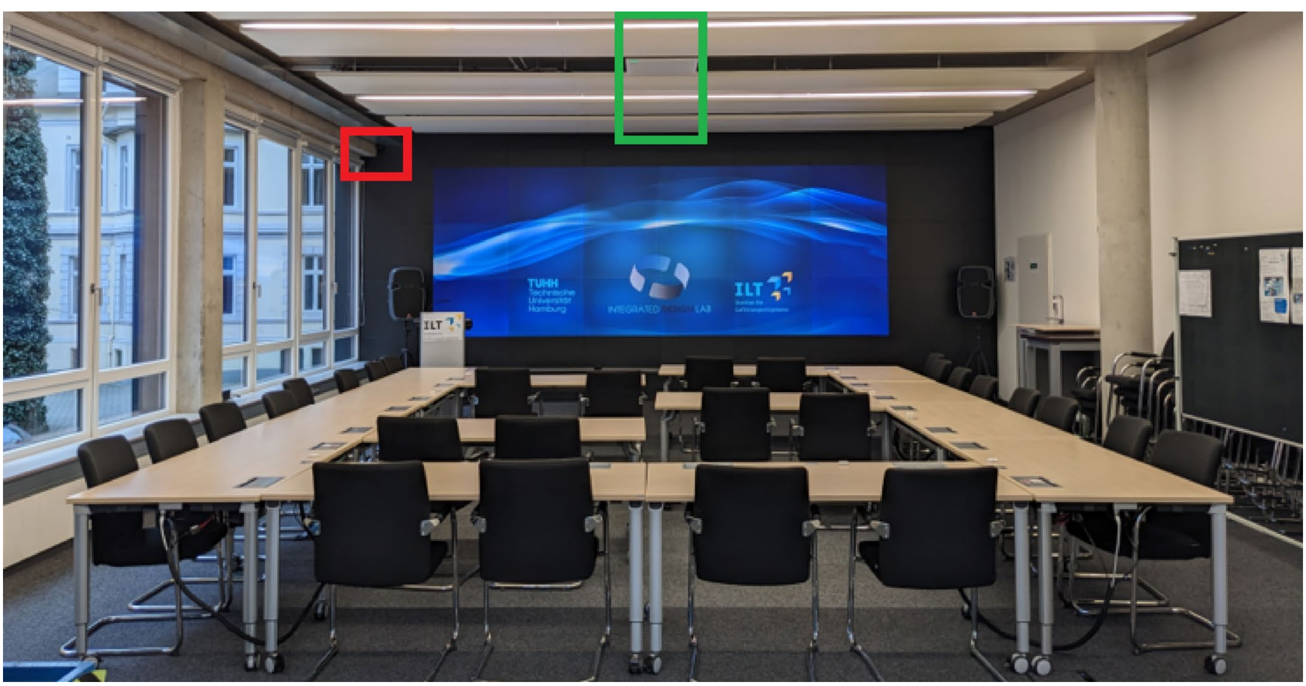


Fig. 2 Rear view of the Integrated Design Laboratory (red mark: position of a webcam; green mark: position of echo-canceling ceiling microphones)

(a) Ceiling microphones in the center of the room.

(b) Webcam behind the speaker's table facing towards the plenum.

Fig. 3 Echo-canceling ceiling microphones (**a**) and a USB webcam (**b**) to broadcast the inside of the IDL/ADL

Table 1 Overview of specification types needed for the ADL adaption

ID	Description
USE	Usability: comprehensibility, learnability, operability, attractiveness
FUN	Framework functionalities
INT	Interface and data exchange requirements
REL	Reliability: maturity, fault tolerance, recoverability
MAIN	Maintainability: analyzability, modifiability, stability, testability

Table 2 Safety requirements and corresponding test cases for the ADL

ID	Description	Test Case
FUN-0031	The ADL must NOT be a security vulnerability for the operator	Hackathon (perform a stress test)
REL-0002	The system must be designed in such a way that the user can never make permanent changes to it	Hackathon (perform a stress test)

UNICADO's Gitlab platform git.rwth-aachen.de/unicado. (see Fig. 4).

2.1 Safety requirements

Because many external participants interact with the ADL, security concerns must also be taken into account when

Table 3 Interfacing requirements and corresponding test cases for the ADL

ID	Description	Test Case
FUN-0035	The ADL must support collaborative work of groups of varying sizes	Test ADL with groups of different sizes
INT-0006	External students must be able to access UNICADO in the ADL via their own laptops	Let external students try to use UNICADO in an ADL via own laptop
INT-0007	External partners must be able to access UNICADO in the ADL via their own laptops	Let external partners try to use UNICADO in an ADL via own laptop
INT-0008	The ADL must provide access to a shared database	Users can pull a database file from Gitlab or the UNICADO's web page
MAIN-0020	User feedback shall be integrated into software enhancements	External users can add issues to a open-source Gitlab repository
PERF-0001	The infrastructure setup must meet defined cost and scalability limits	Calculate costs of planned infrastructure and compare to defined cost limit

Table 4 User-friendliness requirements and corresponding test cases for the ADL

ID	Description	Test Case
USE-0005	The ADL must receive appropriate user manuals supplemented by web support (videos, forums, etc.)	Check if interactive support exists and if it helps users; let unexperienced users assess the support
USE-0006	The users should be guided through all processes as simply as possible and still achieve valid results	Perform user stress test
USE-0007	TLARS and other design parameters shall be scalable via a slider	Verify if a slider allows adjusting parameters
USE-0008	Forms with design parameters should be intuitive for users	Let inexperienced users perform a parameter study, check validity of results, and collect feedback on user-friendliness
USE-0009	During the design loops, the designed aircraft must be visualized so that the process it is going through becomes apparent. If necessary, a comparison with other design results should be possible	Run design process and check if design steps are visualized
USE-0010	User should receive visual feedback on the changes introduced	Change TLARs and parameters and verify if changes are visualized
USE-0011	Installation of the software shall not take longer than 1 h	Measure installation and setup time
FUN-0032	In the ADL it must be possible for teams to work in parallel on the same project or side by side on projects	Let several users work in parallel on the same or different projects
FUN-0033	The ADL must have a central server on which preset modes of the UNICADO GUI for different user groups allow easy entry into the design process	Verify functionality of a central server
FUN-0034	Central server for publication of the results in databases or other exchange folders must be available	Verify functionality of a central server
FUN-0036	There shall be a test lab available for students	Test the test lab with students
FUN-0037	There shall be an upgrade button in the GUI to update the software, which and the content must be transparent	Open the GUI and search for the Upgrade button
MAIN-0019	As soon as a publishable version is available, updates should be made automatically (if necessary) via the web	Try to update ADL via web
PERF-0005	The results of an ADL design must be identical to a design made with the conventional UNICADO software	Compare ADL GUI results with conventional UNICADO software using identical parameters
FUN-0044	Data results on the ADL server need to be visualized and comparable regarding e.g. TLARS	Check import and visualization of data sets

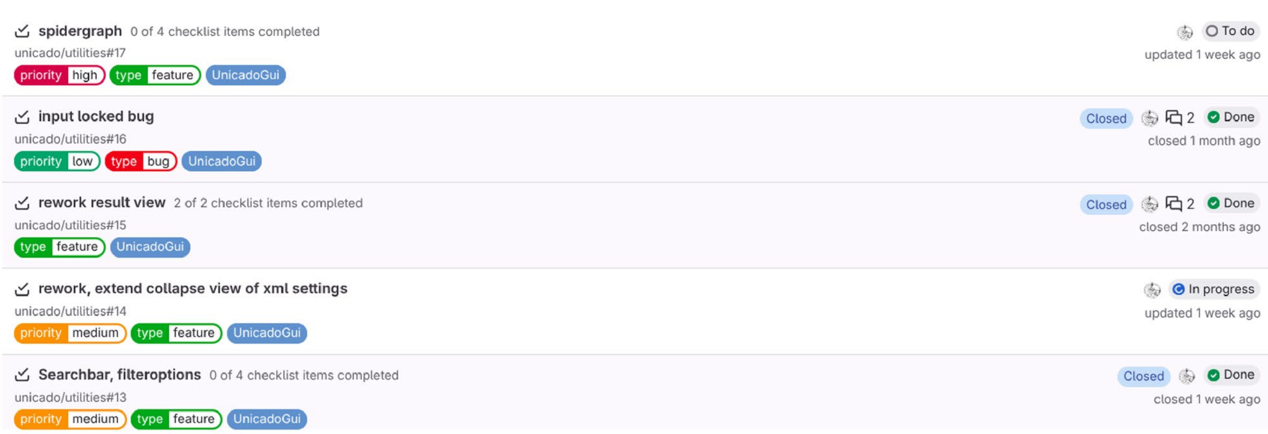


Fig. 4 Example of the UNICADO's Gitlab issue list for GUI adaptions

implementing this setup. These are described in the safety requirements, which are addressed in subsection 3.3.

2.2 Interfacing requirements

Interfacing requirements describe features the users need to interact with each other or the ADL's periphery. Adaptions regarding these requirements include scaling (PERF-0001 & FUN-0035; see Sect. 5) and the ergonomics of the facility (e.g., INT-0006; see Sect. 3).

2.3 User-friendliness requirements

These requirements define the overall functionality which shall increase synergy effects and user-friendliness. Those adaptions mainly affect the GUI's behavior (e.g., USE-0009 which is handled by the CPCAS interface or USE-0007 and FUN-0044 which demand certain graphical elements; see Sect. 4).

2.4 ADL integration process

An incremental development methodology aligned with hardware capabilities was adopted to fulfill the aforementioned requirements (see Fig. 5). After defining the initial requirements and integrating the unmodified versions of the IDL and UNICADO software, each development increment undergoes user-friendliness, interfacing, and security evaluations before progressing or revising changes for the next iteration, ensuring continuous improvement. Using this procedure, potential solutions to specific SRS requirements are first identified and then translated into corresponding software and/or hardware adaptations, which are subsequently implemented within the system.

Following implementation, the solution is verified using the predefined test cases to ensure that it correctly fulfills the specified requirements and does not introduce regressions or incompatibilities. Based on the results of this verification, the process either advances with the integrated solution or re-enters the earlier analysis phase to identify a more suitable alternative. Finally, the overall system is validated to confirm that the implemented changes meet user and operational needs. In particular, the video hardware presented significant compatibility challenges, often necessitating multiple iterations of this loop.

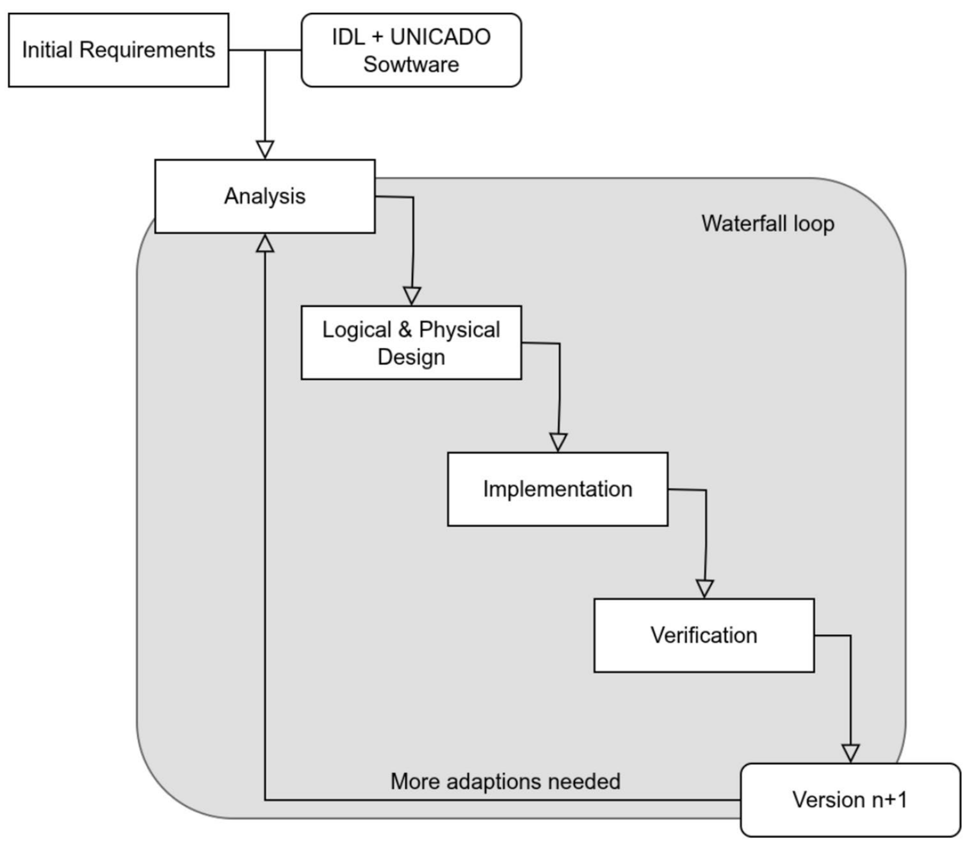


Fig. 5 Waterfall development model incorporating iterative feedback loops

3 ADL architecture

3.1 Initial ADL setup

Based on the given methodology from section 2, work is carried out on the IDL. In this section, key specification of the new ADL are shown to give a brief overview of its initial version which shall act as a blueprint for other ADL clones across the UNICADO-II consortium.

3.2 Multimedia Integration

Prior to the work of UNICADO-II, sharing content on the video wall required active streaming by the presenter, which complicated hybrid meetings, especially for participants without internet access. Although multi-input streaming (e.g., split-screen) was theoretically possible, the complexity hindered practical use, especially among less experienced users. Furthermore, audio and webcam management was dependent on participants' personal laptops, often resulting in disruptions. To streamline multimedia interaction, the ADL gateway server was configured as the central media hub, aggregating all video and audio signals (see Fig. 6). The entire video wall content is cloned and captured by a dedicated video card on this server. The gateway manages conferencing tools such as Zoom and Microsoft Teams, pre-configured for seamless operation alongside audio from the two echo-cancelling ceiling microphones. Audio outputs are distributed via the four stereo speakers in each corner of the room. A user-friendly touchpad integrated into the speaker's table enables initiating hybrid meetings with a single tap, either through the web conference tool or calendar app (see Fig. 7). Multiple interfaces were implemented to allow users to project content onto the video wall:

- ADL gateway server as presenter PC via ClickShare (supports split-screen)
- Windows PCs/laptops via ClickShare (supports split-screen)
- Linux PCs/laptops via Google Cast (supports split-screen)
- Mac, iPhone, or iPad via Apple AirPlay (supports split-screen)
- Direct HDMI connection at the speaker's table (Short-cutting the Clickshare's signal via HDMI switch with hot-plug)

Since this setup was implemented, lectures including split-screen sharing of different content and multiple hybrid meetings of UNICADO and other internal and external projects took place there. After integrating minor tweaks based on user feedback, the concept has proven itself to be feasible and has been used since.

3.3 Networking

A dual-network architecture was already existing, with one being the institute's secured internal network, which restricts guest access for security reasons, and a separate local network isolated from external internet access (see Table 2.1, FUN-0031). The latter was only used to broadcast video signals from pre-mounted PCs via a video over IP transmitter [15]. Since most users prefer bringing their own machines nowadays and the prior integrated ClickShare system enables seamless streaming, the local

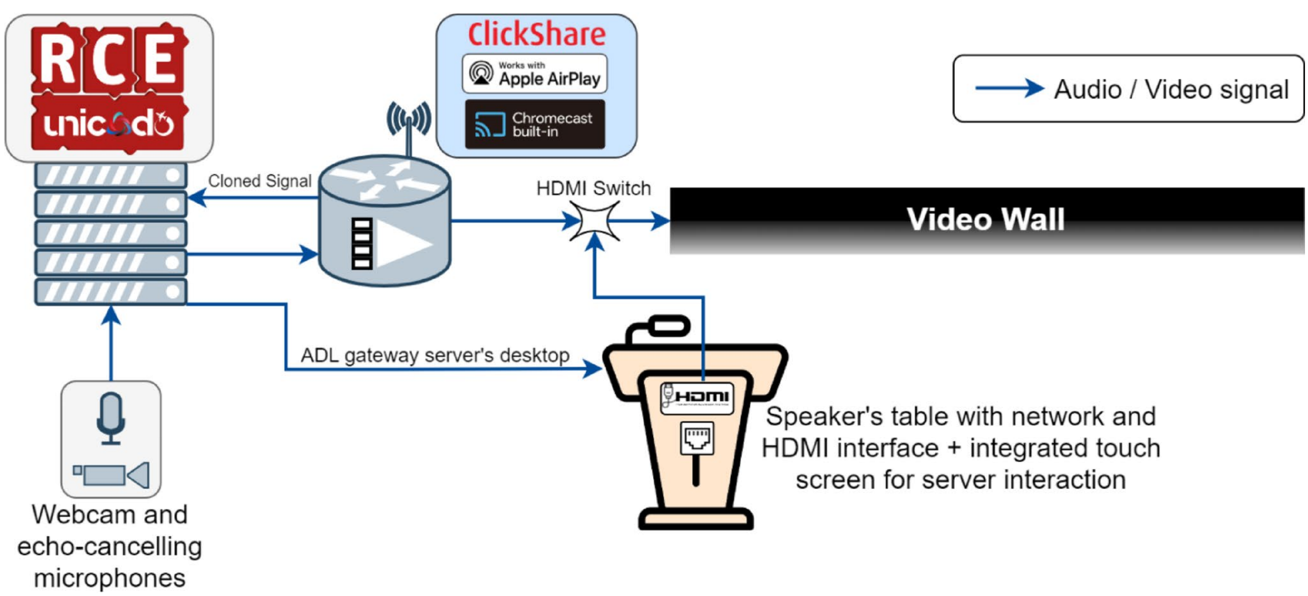


Fig. 6 Multimedia Setup of the ADL

Fig. 7 Host view from the speaker's table. ClickShare is opened automatically for screen sharing (app in the middle); a Zoom meeting is active in the background

network was repurposed. Now it allows guest devices to connect with a centralized ADL gateway server which is essential for establishing RCE-based design chains through which tools and data can be shared. It maintains a static IP address and operates concurrently on both networks, bridging communication between internal servers and external partners through a relay server. The relay server functions as a secure hub, extending RCE connections across multiple networks via SSH tunnels. For this connectivity, external servers were registered on the ILT's whitelist and added to the relay server's trusted hosts. Eventually, users across various ADL sites can connect seamlessly to the UNICADO network, enabling a nation-wide and international collaborative environment where tools, data, and computing requests flow freely and securely across partners (see Fig. 8). Moreover, a high-performance server located at the ILT was added to the network, making it possible to outsource computationally intensive studies.

4 Adaptations of the UNICADO software

UNICADO, being a comprehensive aircraft design platform, requires numerous input parameters across its modules to accurately describe diverse design aspects. In the time writing, the configuration files for the modules within the sizing loop comprise 1,839 editable parameters (see Table 5). Excluding general control settings shared across modules, 1,659 parameters remain. Furthermore, the central aircraft exchange XML file contains 111 parameters within its requirements block, and the RCE workflow includes an additional 54 configuration entries. This vast number of parameters can overwhelm new users which contradicts a low-threshold entry into UNICADO. In addition, each configuration file needs to be manipulated one by one using a file editor. Due to that, guiding through the design process can be arduous (Fig. 9).

In most cases, the number of parameters which a user needs to modify for a specific design task is fundamentally smaller. For example, the choice between *tube and wing* and *blended wing body* configuration (see Fig. 10) already cuts the used parameters of the *Fuselage Design* module in half. This is because many parameters are interdependent and activated only when certain conditions are set. As it is time-consuming identifying these dependencies manually, a web-based GUI for UNICADO is under development that dynamically presents only the essential parameters adapted from the user's design choices. For the given example of fuselage design, the placement of cargo and passengers within a blended wing body configuration presents additional complexity. In this case, the payload must be accommodated within so-called *payload tubes*, which require specification through additional input parameters (e.g., *floor_angle_between_passenger_tubes* to define their arrangement within the aircraft). For a conventual *tube and wing* design, these considerations

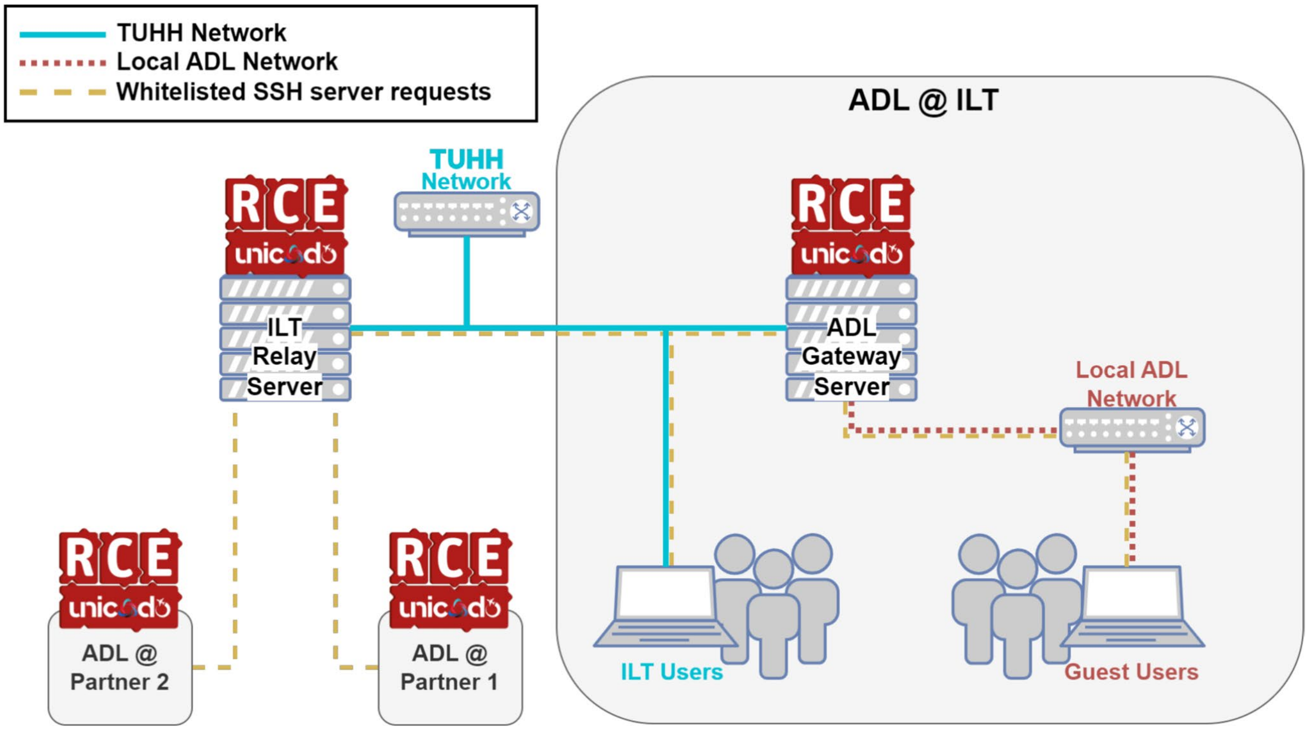


Fig. 8 Network scheme for the UNICADO network

Table 5 Amount of editable configuration parameters for each module within the sizing loop

Module	Parameters
Aerodynamic Analysis	134
Constraint Analysis	48
Create Mission XML	53
Empennage Design	184
Fuselage Design	621
Landing Gear Design	54
Mission Analysis	51
Propulsion Design	31
Systems Design	490
(Fuel) Tank Design	38
Weight and Balance Analysis	22
Wing Design	113

become obsolete. To enable this simplified parameter handling, a standardized, keyword-based XML structure was defined for the aircraft exchange and module configuration files. In this format, parameter values reside in terminal *endnodes*, which do not contain further children. The GUI interprets these keywords to render appropriate interactive elements (e.g., sliders, dropdown menus) that streamline user input (see Figs. 9 and 10). For numerical parameters, an endnode is structured as follows:

To display a switch or a drop-down menu in the GUI, the keywords *Selector* and *Switch* were introduced. With these, users can decide what design methods shall be applied (e.g., configure a fuselage for *tube and wing* or a *blended wing body*) or whether a specific behavior should be active (e.g., automatic altitude changes in the mission

Fig. 9 Example of a numerical parameter represented via an editable text field and a slider (left end = endnode's lower boundaries; right end = endnode's upper boundary)

```
<node description="Description of the node">
        <value>VALUE</value>
        <unit>UNIT</unit>
        <lower_boundary>LOW</lower_boundary>
        <upper_boundary>HIGH</upper_boundary>
</node>
```

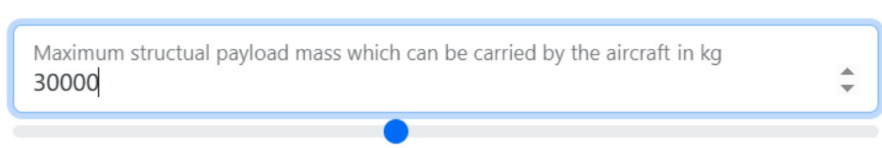
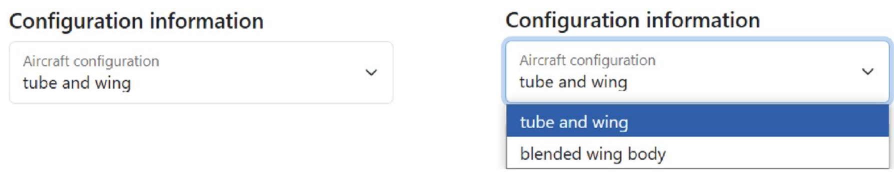


Fig. 10 Example of an *endnode* displayed as a dropdown menu using the *selector* keyword (left: closed; right: expanded)

```
<node description="Description. Switch: True (details) / False (
    details)">
        <value>True/False</value>
</node>
```

```
<node description="Description. Selector: OPTION_1 (details) /
    OPTION_2 (details) / ... / OPTION_N (details)">
        <value>OPTION_1 − N</value>
</node>
```



analysis). Boundaries and units are not needed for these endnodes:

After these style changes were adopted by all modules of the UNICADO tool chain, an initial GUI version was deployed which is continuously improved based on user feedback and the requirements from the SRS. Due to the fact, that the GUI is able the lead the user through the aircraft XML's requirements block and the modules' configuration files in an intuitive manner (Fig. 11), test users reviewing the GUI's Gitlab issue board already tend to favor its usage. With the introduced XML style being a coding standard for all future work, more adaptions of the UNICADO software to interact with the GUI are not contemplated. Hence, further GUI development can be carried out in parallel to the UNICADO design software.

5 Definition of scalable ADL clones

With the initial ADL setup, a scaled-down version was designed to fit the budgets and infrastructure of other partner institutes, ensuring that all sites maintain compatibility and connectivity. The minimum architecture elements for these scaled ADL clones are:

- Networking capability to include guest users:
  - Either a dual-network architecture or a single network accessible to both guests and hosts.
- At least two full HD video display elements (e.g., projector, powerwall, TV).
- At least one webcam and microphone for audiovisual broadcasting.
- A central gateway server acting as:
  - The interface to the UNICADO network.
  - The hosting platform for shared tools.
  - The broadcasting instance (optional).
- Multiple, easily accessible video input interfaces for users.

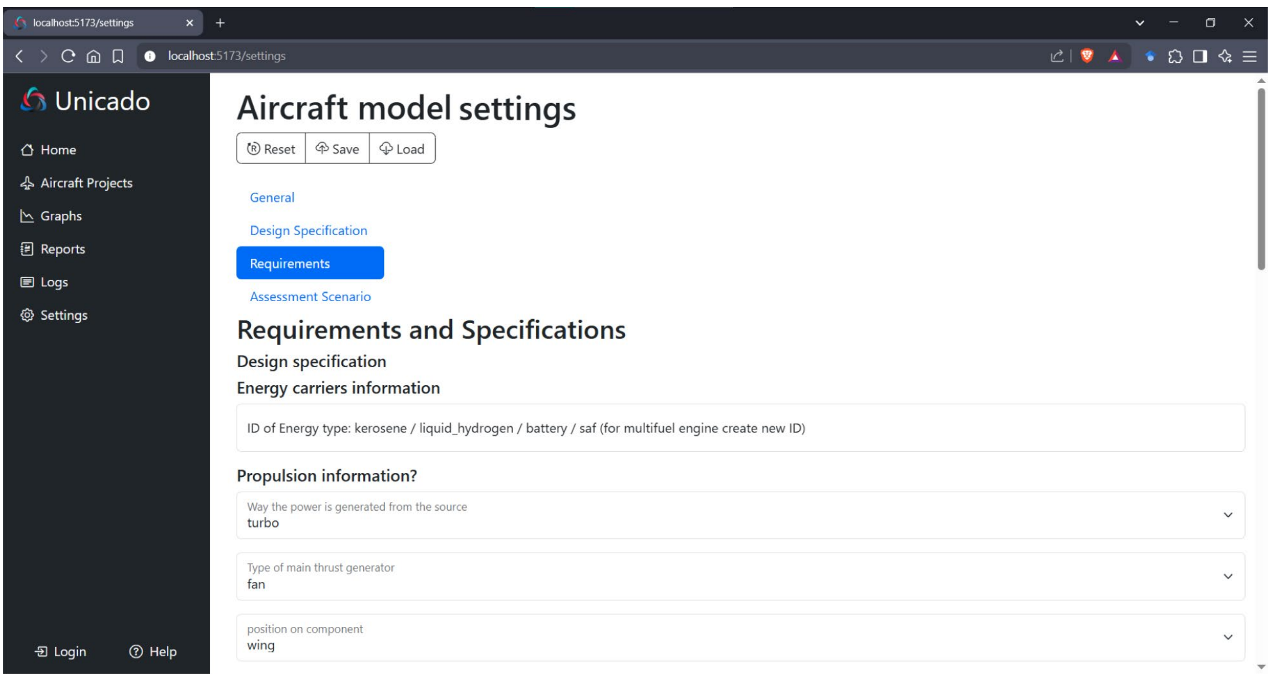
Following this groundwork, partner institutes at RWTH Aachen (Fig. 12) and TU Berlin have begun deploying their own ADL sites based on these specifications and the lessons learned by the ILT.

5.1 Lessons learned

Multimedia technology is frequently plagued by compatibility issues that prove remarkably resistant to resolution. E.g., echo-canceling ceiling microphones perform well in theory, but they increase the system's complexity dramatically and they do not deliver the excellent sound quality one would expect. Even though, solving those problems were time-consuming on the developer's end, utilization of the IDL increased dramatically. While lectures have not been held there due to complexity concerns, this was changed after the first iterations of the hardware adaption loop. Beyond that, (hybrid) meetings initially needed to be overseen by well-instructed members of the ILT. This dependency vanished over time giving more room to focus on software-related issues. For this reason, newer generations of all-in-one solutions for webcam, audio output and input (e.g., see Fig. 12) are preferred for further ADL adaptions. Those usually offer a plug and play functionality which can easily be implemented into any kind of ADL. Even though these devices should not exceed a reasonable financial limit, the ADL will benefit from investments in upscale products since low-priced ones hinder the seamless interaction between online plenum and on-site guests immensely.

```xml
<energy_carriers description="Energy carriers information">
    <energy_carrier ID="0" description="One specific energy carrier">
        <type description="Energy type (for multifuel engine create new ID). Selector: kerosene / liquid_hydrogen">
            <value>kerosene</value>
        </type>
    </energy_carrier>
</energy_carriers>
<propulsion description="Propulsion information">
    <propulsor ID="0" description="Information for specific propulsor">
        <powertrain description="Way the power is generated from the source. Selector: turbo">
            <value>turbo</value>
        </powertrain>
        <type description="Type of main thrust generator. Selector: fan">
            <value>fan</value>
        </type>
        <position description="Propulsor position (arrangement order acc to ID order)">
            <parent_component description="Position on component. Selector: wing / fuselage / empennage">
                <value>wing</value>
            </parent_component>
```

(a) Propulsion requirements in the aircraft XML's requirements block



(b) Editable propulsion requirements displayed in the GUI

Fig. 11 Effect of using the keyword-based endnode style (a) on the GUI (a)

6 Conclusion and future work

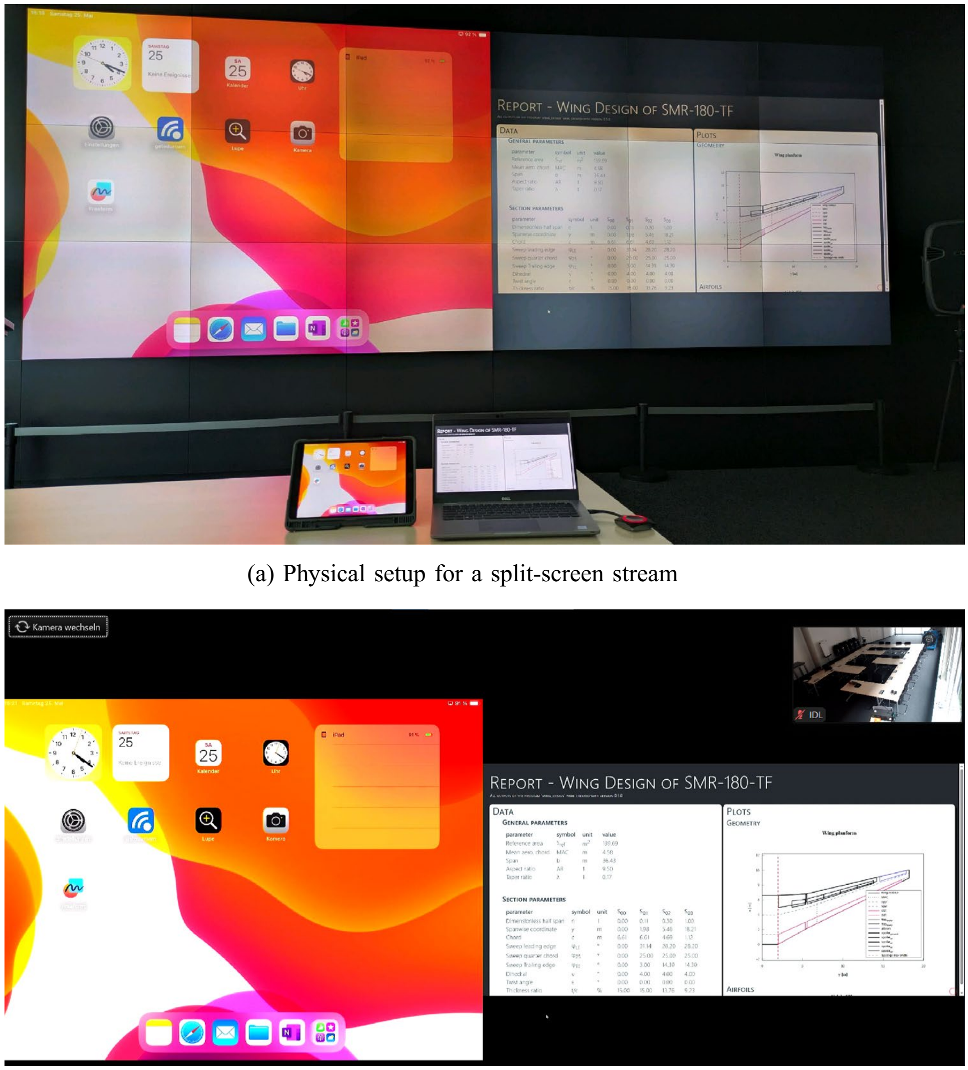
In this work, significant adaptations were made on both the software and hardware fronts to realize the ADL concept integrated with the UNICADO platform. On the software side, the UNICADO design environment was enhanced by developing a new, intuitive web-based GUI that dynamically guides users through essential design parameters. This was achieved by standardizing and restructuring the aircraft exchange and module configuration XML files into a keyword-based endnode format, enabling the GUI to present only relevant inputs via user-friendly elements such as sliders, drop-down menus, and switches. This approach effectively reduces the cognitive load on users by hiding the complexity of the large parameter space, thus lowering the entry barrier for novices. However, to fulfill the full SRS sheet, further adaptions of the visualization and analysis functionality must be integrated.

Fig. 12 ADL Aachen with integrated webcam and microphone within two smartboards (left and right monitor)

On the hardware side, the IDL was transformed into the initial ADL setup by upgrading its networking and multimedia infrastructure. A dual-network architecture was implemented, separating guest access and internal secure networks to facilitate safe and flexible remote connectivity through an ADL gateway server and relay servers, using SSH tunnels to extend RCE-based workflows across institutions. On said network, a high-performance server offers its service which already lowers computational time by 50 % and more per aircraft design in comparison to a regular work laptop. The multimedia system was overhauled by centralizing audio and video management via the gateway server equipped with dedicated video capture hardware and multiple video input methods (including ClickShare, Google Cast, and Apple AirPlay), and an intuitive touchpad interface to control hybrid meetings like in the example below (Fig. 13). Finally, a scalable, minimum hardware and software configuration was defined to allow partner institutions with varying budget and infrastructure capabilities to deploy compatible and interconnected ADL clones until late 2025, thus enabling a nationwide concurrent engineering network leveraging the UNICADO platform.

For the next steps, the GUI shall be extended to include an RCE interface to start workflows directly instead of only preparing RCE's input files. Also, the results will be visualized by including the HTML-based tool reports into its own framework. Additionally, a comprehensive database will be created to store validated aircraft designs, enabling users to explore design parameters and Top-Level Aircraft Requirements (TLARs) to understand their impact on various systems. Moreover, each ADL will serve as a hub for student engagement through design challenges and other activities, fostering the development of future researchers capable of tackling complex aircraft systems. To support this, a lecture series involving all partners is being developed preparing students for the diverse disciplines within the UNICADO design chain. Said lecture will be supplemented by practical units using the UNICADO software and will culminate in a design challenge. The practical units of the course will include training on UNICADO using GUI within each partner's ADL.

Fig. 13 Two devices sharing screens via Apple AirPlay (iPad, left) and ClickShare dongle (laptop, right)



(a) Physical setup for a split-screen stream

(b) Zoom stream via the ADL gateway server of the two split-screen devices

Author contributions G.P. is the main author of this manuscript. V.G. reviewed the manuscript and added valuable input for further improvement.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

1. Böhnke, D., Nagel, B., Gollnick, V.: An approach to multi-fidelity in conceptual aircraft design in distributed design environments. In: 2011 Aerospace Conference, pp. 1–10 (2011). https://doi.org/10.1109/AERO.2011.5747542 . IEEE
2. Lefebvre, T., Bartoli, N., Dubreuil, S., Panzeri, M., Lombardi, R., Della Vecchia, P., Stingo, L., Nicolosi, F., De Marco, A., Ciampa, P.D., et al.: Enhancing optimization capabilities using the agile collaborative mdo framework with application to wing and nacelle design. Prog. Aerosp. Sci. **119**, 100649 (2020). https://doi.org/10.1016/j.paerosci.2020.100649
3. Hartmann, J., Pfeiffer, T., Breymann, B., Silberhorn, D., Moerland, E., Weiss, M., Nagel, B.: Collaborative conceptual design of a mid-range aircraft under consideration of advanced methods for

technology assessment. In: Proceedings of the 31st International Congress of the Aeronautical Sciences, Belo Horizonte, Brazil, pp. 9–14 (2018)
4. DARcorporation: Advanced Aircraft Analysis (AAA). Accessed May 25, 2025 (2025). https://www.darcorp.com/advanced-aircraft-analysis-software/
5. Raymer, D.P.: RDSwin: Aircraft Design Software. Accessed May 25, 2025 (2025). https://aircraftdesign.com/rdswin/
6. PACE Aerospace & IT GmbH: Desmo (formerly Pacelab APD & SysArc). Accessed May 25, 2025 (2025). https://pace.txtgroup.com/products/preliminary-design/
7. Belie, G.: Non-technical barriers to multidisciplinary optimization in the aerospace industry. In: 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, p. 5439 (2002). https://doi.org/10.2514/6.2002-5439
8. Seider, D., Litz, M., Schreiber, A., Fischer, P.M., Gerndt, A.: Open source software framework for applications in aeronautics and space. In: 2012 IEEE Aerospace Conference, pp. 1–11 (2012). https://doi.org/10.1109/AERO.2012.6187340 . IEEE
9. Maierl, R., Liersch, C.M., Deinert, S., Kleinert, J., Alder, M., Moerland, E.: Application of cpacs in military aircraft design. (2024). Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV
10. Alder, M., Moerland, E., Jepsen, J., Nagel, B.: Recent advances in establishing a common language for aircraft design with cpacs (2020)
11. Schültke, F., Aigner, B., Effing, T., Strathoff, P., Stumpf, E.: Micado: overview of recent developments within the conceptual aircraft design and optimization environment. (2021). https://doi.org/10.25967/530093 . Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV
12. Zimmnau, M., Schültke, F., Stumpf, E.: Unicado: multidisciplinary analysis in conceptual aircraft design. CEAS Aeronaut. J. **14**(1), 75–89 (2023). https://doi.org/10.1007/s13272-022-00620-3
13. Bachmann, A., Lakemeier, J., Moerland, E.: An integrated laboratory for collaborative design in the air transportation system. In: Concurrent Engineering Approaches for Sustainable Product Development in a Multi-Disciplinary Environment: Proceedings of the 19th ISPE International Conference on Concurrent Engineering, pp. 1009–1020 (2012). https://doi.org/10.1007/978-1-4471-4426-7_85 . Springer
14. Dineva, E., Bachmann, A., Moerland, E., Nagel, B., Gollnick, V.: Empirical performance evaluation in collaborative aircraft design tasks. In: 20th ISPE International Conference on Concurrent Engineering, pp. 110–118 (2013).https://doi.org/10.3233/978-1-61499-302-5-110 . IOS Press
15. Dineva, E., Bachmann, A., Knodt, U., Nagel, B.: Human expertise as the critical challenge in participative multidisciplinary design optimization–an empirical approach. In: Moving Integrated Product Development to Service Clouds in the Global Economy, pp. 223–232 (2014). https://doi.org/10.3233/978-1-61499-440-4-223 . IOS Press
16. ISO, I.: Iec/ieee 29148: 2018 systems and software engineering-life cycle processes. Requirements engineering (2018)