DOI: 10.13009/EUCASS2023-613
Aerospace Europe Conference 2023 – 10TH EUCASS – 9TH CEAS

Knowledge-based Integration of Aircraft System Simulation within Aircraft Conceptual System Design

Authors

Kristian Amadori [1], Max Baan [2], Robert Braun [3], Christopher Jouannet [1], Ingo Staack [4]

[1] Saab AB, Linköping, Sweden
[2] ParaPy, Delft, The Netherlands
[3] Linköping University, Linköping, Sweden
[4] Technische Universität Braunschweig, Germany

Abstract

This paper describes the work done on automating aircraft simulation in the conceptual aircraft design stage. The conceived knowledge-based engineering application is capable of modelling aircraft system architectures, placing them at the appropriate locations of the aircraft and generating simulation models for steady state and transient, time-dependent systems. Applying the application on a UAV use case provided by Saab showed that lead time reductions of more than 95% can be achieved and that a hundred-fold of additional architectures can be evaluated.

1. Introduction

At the Aircraft Conceptual Design (ACD) stage, not all details of the aircraft and its systems are available, so simplifications, abstractions and assumptions must be applied to ensure that all relevant engineering domains are captured. The level of detail depends on the specific aircraft design and the associated analysis tasks and focus. Models representing the various disciplines required are developed with increasing fidelity and complexity during the conceptual design phase to capture the relevant behaviours and characteristics to minimise programme risk and maximise knowledge gain.

These domain-specific models include onboard systems such as avionics, actuators, cooling systems, landing gear, cabin equipment, etc. Due to their impact on the overall aircraft capabilities, these systems are a) not negligible, b) may not be captured with correct trends by (semi-)empirical models, and c) play a prominent role in modern, highly efficient system layouts [1]. The main onboard system characteristics that can have an impact at the aircraft level are typically size, weight (fixed and variable) and inertia, power demand, cooling demand, drag contribution. These are characteristics that have a direct impact on mission capability and need to be addressed systematically [2], but also other objectives such as operational constraints, component location, redundancy and safety requirements, loads and maintenance.

At present, many aerospace OEMs approach the initial ACD phase primarily with semi-empirical data, technical data sheets of available (commercial off-the-shelf) components or subsystems, or with a limited number of estimates from steady-state simulation models. To improve confidence in early design, reduce risk and provide less uncertainty in expected performance metrics, it is desirable to include more steady-state simulations and to simplify the use and coupling of higher fidelity system analyses, such as transient simulations to explore time-dependent phenomena, while still ensuring rapid exploration of different architectures.

State of practice

A typical state-of-the-art component based ACD workflow, including the onboard system architectures, is illustrated in Figure 1. The process starts with gathering and reviewing requirements with the various stakeholders. As this is done in the ACD phase, the timeframe is short (a few days) and the group involved is small (a few people), so the resources available are limited. Several steps of the process are manual, requiring translation between different formats and the delivery of different documents. Considerable time is spent gathering the correct inputs and translating/formatting data from one step to another between the different data formats. This makes the process prone to errors due to manual editing, lack of standardisation and the duplication of information in different formats and locations (data consistency).

DOI: 10.13009/EUCASS2023-613

Currently, the step of linking onboard systems architecture to transient simulations and incorporating them into the ACD phase is seldom performed because the amount of manual work required is in contrast with the constraints of early design phases. In addition, the models required are usually of high-fidelity for two reasons: firstly, because they resemble (in a mathematical sense) the underlying physical effects and secondly, they are mostly used for verification and validation, which takes place much later in the design process when significantly more data is available and required. Typically, such system models require a full definition of the system(s) and therefore do not meet with the needs of ACD, where the goal is instead to explore different architectures to understand the possibilities offered by different approaches (aka. aircraft design/configuration vs. system design dependencies and performance impact). In summary, there is a clear need for Design Space Exploration (DSE) within ACD, which requires workflow automation and adequate model fidelity to enable large-scale DSE.

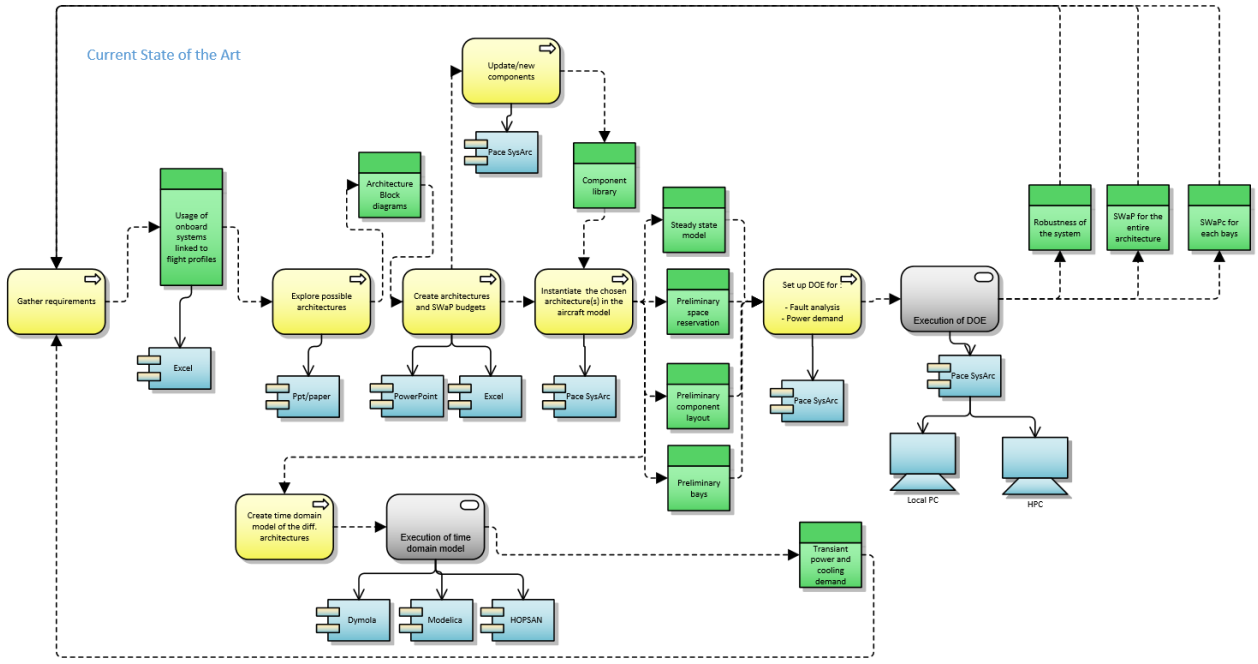

Figure 1: overview of the state-of-the-art process of a system-centric aircraft conceptual design task.

2. Frame of Reference

This paper focus on aircraft system simulation, in particular steady-state systems modelling and transient, time-dependent systems modelling.

2.1 Aircraft system simulation

Component-based system simulation is commonly used to develop and analyse system architectures. For this purpose, a set of components (often from a component library) is assembled through various types of connections and relationships, forming a system architecture to achieve the desired functionality. In the context of ACD, onboard systems are often modelled in one dimension only, but can also be two or three dimensional (3D) systems, for example to study multi-body mechanics, such as landing gear mechanics. System simulation typically involves solving a set of differential equations with respect to time by integrating state variables using forward or backward differentiation methods. While each modelling approach has its inevitable advantages and disadvantages, there is a clear need for higher fidelity time-dependent models, especially for highly integrated systems analysis, to identify the dynamic interdependencies between systems [3].

2.2 Steady-state systems modelling

One can describe the system architectures as a logic combination of components without the knowledge of detailed physical properties. Figure 2 shows such a model of an aircraft actuation system in which arrows indicate the logical relationships between the components. No detailed explicit physical connections or energy flows are included in that model, it is a pure logical relationship metamodel. However, due to the empirical origin of that metamodel, based on existing system architectures (see e.g. [4]) and domain expert knowledge, the physics are implicitly included in the system logical relationship (e.g., in kind of energy paths, comparable with structural load paths). Following the

DOI: 10.13009/EUCASS2023-613

classical setup of (power/energy-based) onboard systems break-down in Power Generation Systems (PGS), Power Distribution Systems (PDS), and Power Consumption Systems (PCS), the PGS is indicated as red dashed boxes, and a part of the PDS and PCS is indicated as blue dotted boxes in the figure.

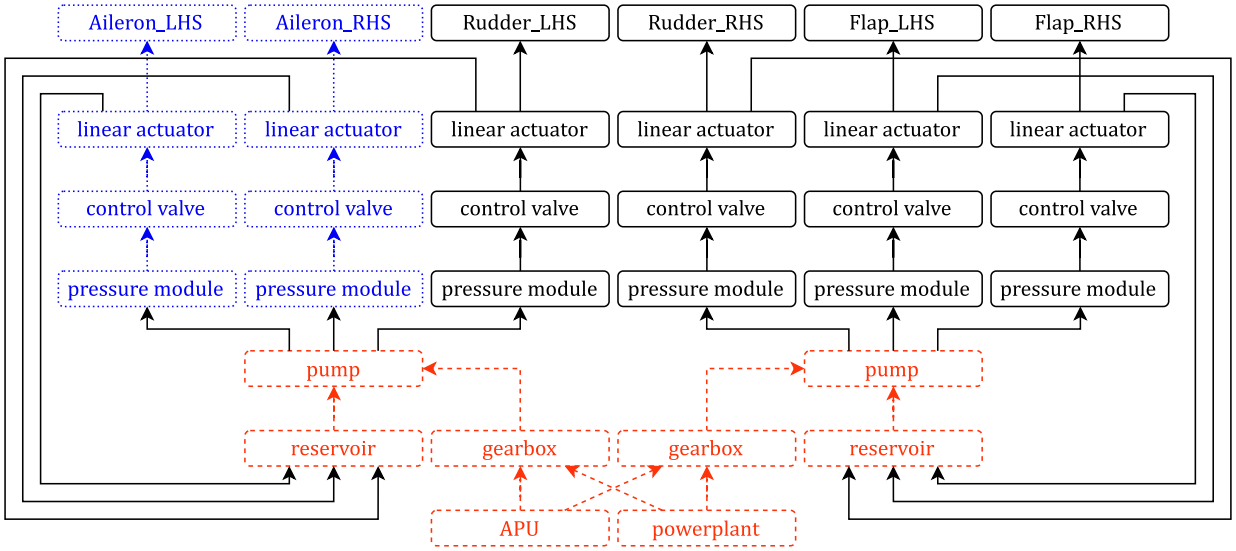

Figure 2: A logical relationship model of a hydraulic system for control surface actuation.

The component or module (an alternative name for subsystem) based relationship model shown in Figure 2 goes beyond the purely top-level functional PGS/PDS/PCS approach by adding components or modules for the sake of control, such as the control valve for each hydraulic actuator. Table 1 shows the high-level coexistence of the main function (realised by the physical power/force components) and its associated and required control part, denoted here as the cyber part[1] of the hydraulic actuation system. In general, it can be concluded that almost any type of high-efficient energy/power system excludes the use of control by throttling due to the high losses. Instead, highly efficient systems represent a combination of the physical implementation to fulfil the required function and the related control part which often includes some kind of electronics, here referred to as the cyber part.

Table 1: Systematic break-down of the actuation system as a cyber-physical system.

System	Physical Realisation	Cyber or Control Part
Power generation system	Create hydraulic power (pressure & flow)	Control the supplied power. E.g., in a constant pressure system by observing and regulating the supply pressure
Power distribution system	Support hydraulic power to all consumers	Eventually provide separation means to enhance system reliability (e.g., isolate leaking consumers)
	Enable inter-system power transfer	Control the power (no fluid!) transfer between the systems
Power consumption system	Deliver mechanic force, moment or velocity	Control the delivery of mechanic power or force, or enable a (target) position control

2.3 Transient, time-dependent systems modelling

To simulate a model in a time-dependent system simulation tool such as Hopsan [5], a complete description of all physical components and their interconnections is required. This requires significantly more information and knowledge than is described by the model classically used in the ACD stage. The additional information required includes both detailed inter-component connection details, largely extending the strict functional-causal relationships identified in Figure 2 and Table 1, and the need and logic for some auxiliary components that are required on the target

[1] The term "cyber" for the control system is used here because the sense and control loop is now often realised by mechatronic systems with the control algorithm implemented in "cyber" software.

DOI: 10.13009/EUCASS2023-613

system and consequently also in the simulation model (e.g., pressure relief valve). In addition, some simulation tool specific adaptations may also be required, depending for example on which component models are available, how they are allowed to be connected and the number and type of connection ports available on each component. This task may also involve additional components (e.g., a time-step delay component to solve numerical loops) that are not part of the target system.

2.4 Information modelling and implementation strategies

To increase and intensify the compatibility between aircraft design and systems modelling/engineering, the available information must be documented in an accessible way for the tools used in the design process. A common way today is to formulate the aircraft-related data in XML/CPACS [6] or JSON, but more dynamic methods that allow direct integration of a high-level API for querying, such as relational databases [7] or ontologies [8], are also potential candidates. A recent publication by Gradel et.al. links an XML database to the well-established systems engineering modelling languages UML and SysML to provide intuitive graphical representation and simplified manual data manipulation [9].

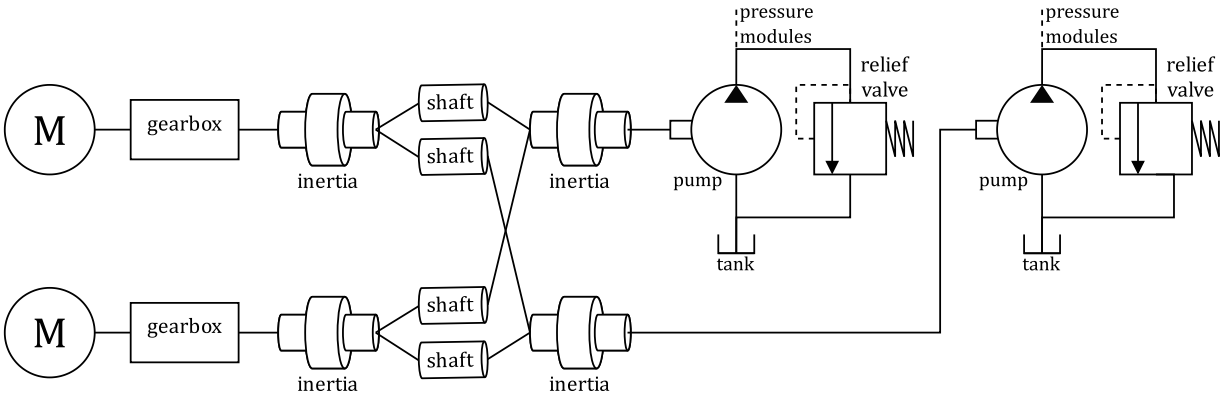

Figure 3: Hopsan model of a power generation system in UAV, corresponding to the red part of Figure 2.

Regardless of the chosen information representation approach, a meta-level model containing more physically based system architecture design information is needed to create a "real" component-based, physically feasible system layout. Figure 3 shows such a complete physical PGS model (shown as red dashed boxes and lines in Figure 2) as it could be realised in Hopsan. As can be seen, additional inertia and shaft components are required to connect both power sources to both hydraulic pumps. A pressure relief valve is also added to keep the pressure at a constant level. Alternatively, this could have been achieved by using a pressure compensated pump or a centrifugal pump, but some means of pressure control is required. This information is not included in the original diagram.

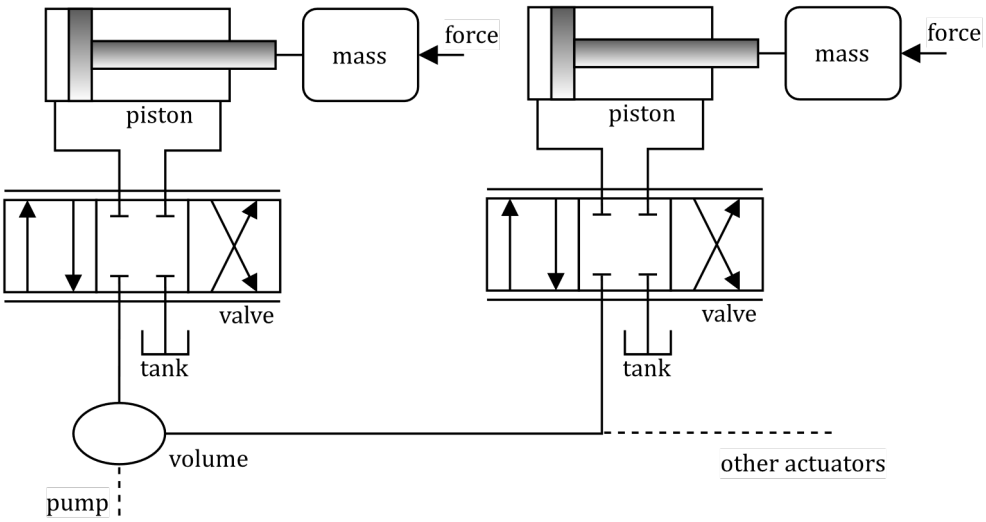

Figure 4: Hopsan model of part of the PDS and the PCS system of the UAV, corresponding to blue dotted part of Figure 2. The volume serves as a metamodel of the pressure module, the mass and force represent the linear dynamics of the control surface.

DOI: 10.13009/EUCASS2023-613

Following the transformation process from the system architecture sketch to the selection of the system setup and finally to the creation of the simulation model, Figure 4 shows a part of the Power Distribution and Consumption System (PDS and PCS), represented by the blue dotted boxes in Figure 2, as a possible realisation in the Hopsan software. For simplicity, only two of the six actuators are shown. The pressure modules are replaced by a single volume (one for each pump), which allows multiple (power distribution) connections. The mass and force components represent the equivalent inertia and external forces on each control surface. Note that unlike the PCS functional diagram, the linear actuator - representing the power consumer - is not directly connected to the tank. Instead, it has two connections to a valve which provides the control functionality (position, force or speed) by controlling the connection of the cylinder chambers to the pump and the tank.

3. UAV System Simulation

The Design Exploration Framework based on AI for froNt-loaded Engineering project (short: DEFAINE) [10] addresses two main needs of Saab regarding onboard systems modelling and assessment within aircraft conceptual design: exploration of different architecture alternatives for onboard systems and automation of time-domain modelling of onboard systems. This will enable:

- Automatic generation of analysis models of onboard system architectures
- Searching for feasible solutions and performing sensitivity analyses to account for the uncertainties in the concept design
- Trade studies between different architectural concepts
- Faster identification of onboard system architecture and design candidates leading to more efficient front loading.

Using the ParaPy KBE software development kit [11], a knowledge-based engineering (KBE) web application has been created to automate the generation of system architectures and system simulation models. This application, called ASYSTOR (Aircraft SYStems configuraTOR), automates aircraft geometry generation, system selection and sizing, system placement and system simulation.

3.1 Use Case

The use case is an Unmanned Aerial Vehicle (UAV) designed following the design process shown in Figure 1, including onboard system as depicted in Figure 5. The focus is on the Primary and Secondary Flight Control System (PFCS, SFCS). The PFCS is an excellent candidate for information reuse through knowledge-based engineering because of its well-defined design principle and strict reliability constraints. It resembles a so-called Class A system, which is essential for the safe operation of an aircraft.

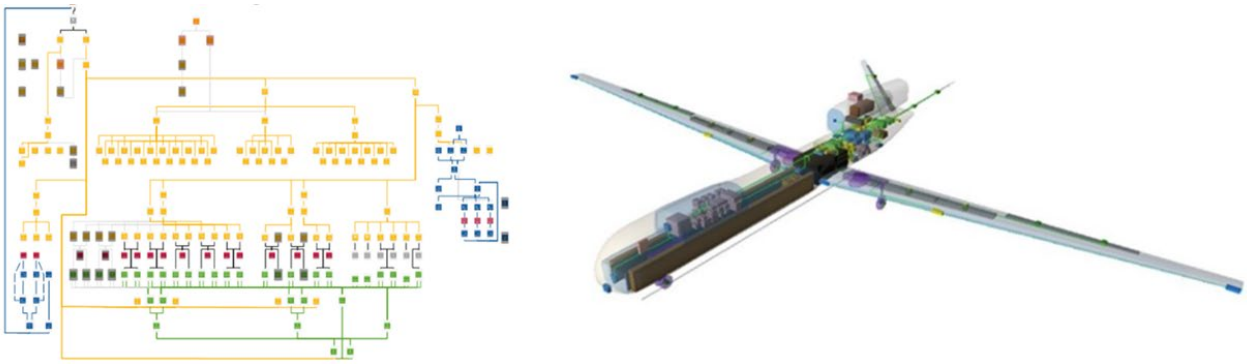

Figure 5: Schematic of the flight control system architecture (left) and the component allocation (right).

Consequently, the system safety dominates the system architecture designs before system efficiency, weight and other system properties. Previous work on this topic has already been done in [12]. By analysing existing (hydraulic) PFCSs (e.g., using the system properties published in [4]), the safety and reliability imposed design rules have been obtained, like e.g., three parallel (hydraulic) PGSs and PDSs for redundancy reasons of a CS-25 category certified aircraft.

DOI: 10.13009/EUCASS2023-613

3.2 Automating the system simulation

The ASYSTOR application is built in an object-oriented way using the central product model approach, integrating four different engineering perspectives: geometry, system architecture, steady-state analysis and transient analysis. All information exchange between the different models takes place through this central product model. Using ParaPy's graphical user interface package, a web application has been developed to expose the design parameters and visualisations of the different engineering perspectives to the engineer.

Geometry

The ASYSTOR geometry perspective (see Figure 6) allows the engineer to vary the configuration and dimensions of the aircraft. By making use of the geometry libraries of ParaPy, the capability to automatically generate geometry within seconds is included in the application The engineer can vary planform parameters such as wingspan, but also topology parameters such as the type of empennage (V-tail or T-tail). Both types of parameters affect the placement of system components. For example, a V-tail has two movables, while a T-tail has three, resulting in more components. Also, a larger span will result in a different position of the movable, which in turn will result in a different position of the actuator system.

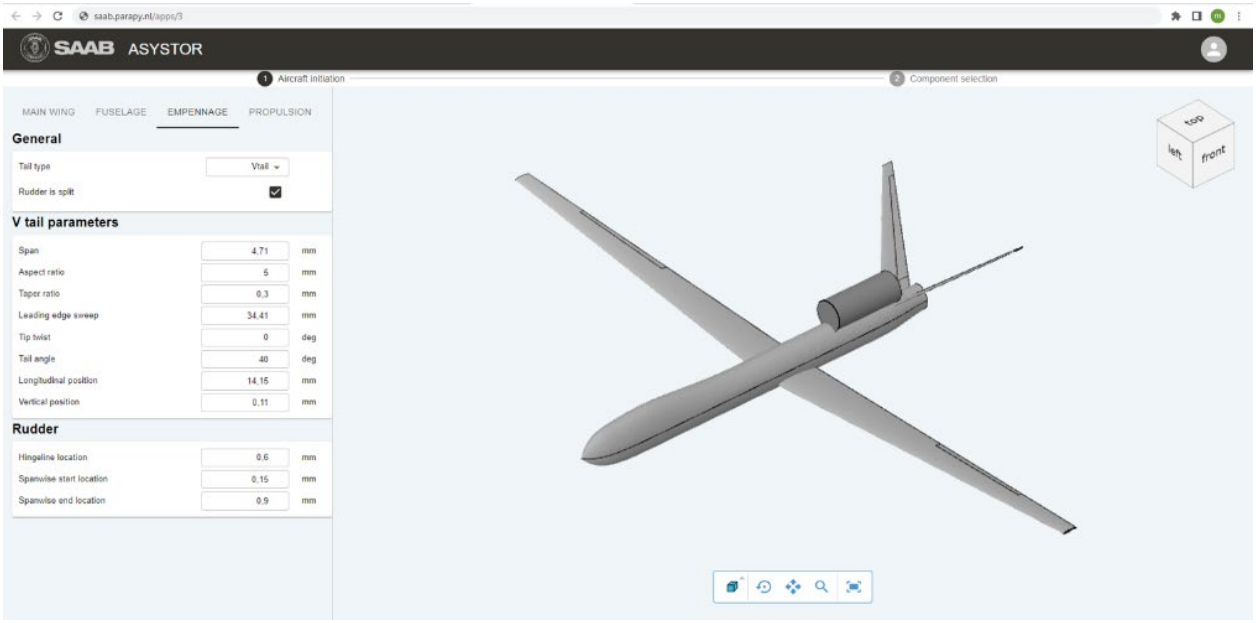

Figure 6: The geometry editor with the input fields (left) and the aircraft geometry representation (right).

Systems architecture selection

The logical structure and configuration of a hydraulic system for actuating the UAV's control surfaces can be configured as shown on the left in Figure 7. The system can be described in ASYSTOR according to the metrics shown in Figure 2.

Based on this system metamodel, the system architecture editing perspective focuses on selecting components based on the aircraft topology and system requirements and connecting the systems. Before performing these actions, the aircraft topology must first be analysed as it will determine the number of systems required. Other drivers for the number of systems are system requirements such as redundancy, for example, leading to split movables. Once the number of components has been determined, they are sized, and the connections between them are formalised in a graph-like structure. At this stage, a functional representation of the system architecture is available, as shown on the right of Figure 7.

DOI: 10.13009/EUCASS2023-613

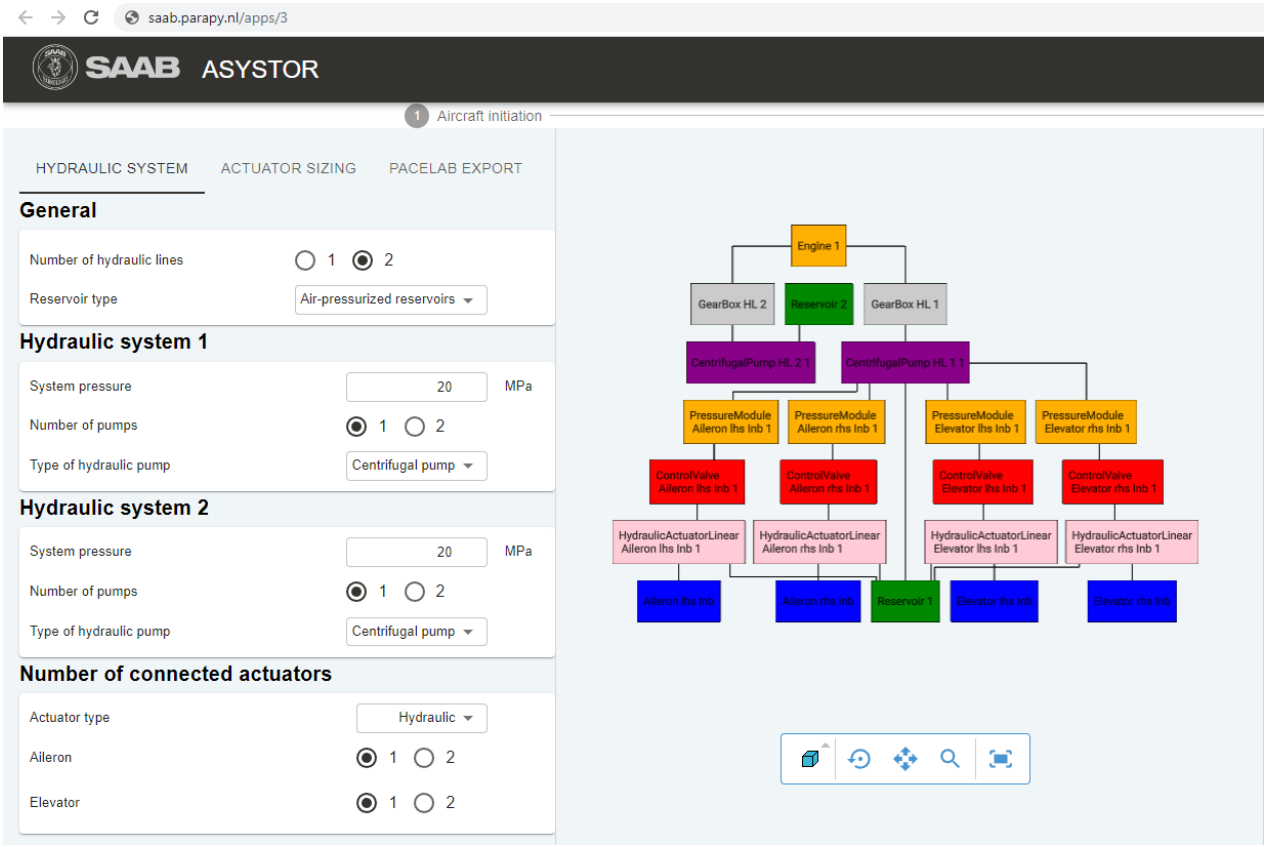

Figure 7: The system architecture editor with the input fields (left) and the architecture representation (right).

Systems placement

The functional representation of the system architecture is converted into a 3D representation using the geometry definition of the aircraft. Topology aspects such as split movables are taken into account when positioning the systems. The combination of the functional representation and the 3D positions of the systems are used to generate simulation models for steady-state analysis and transient analysis.

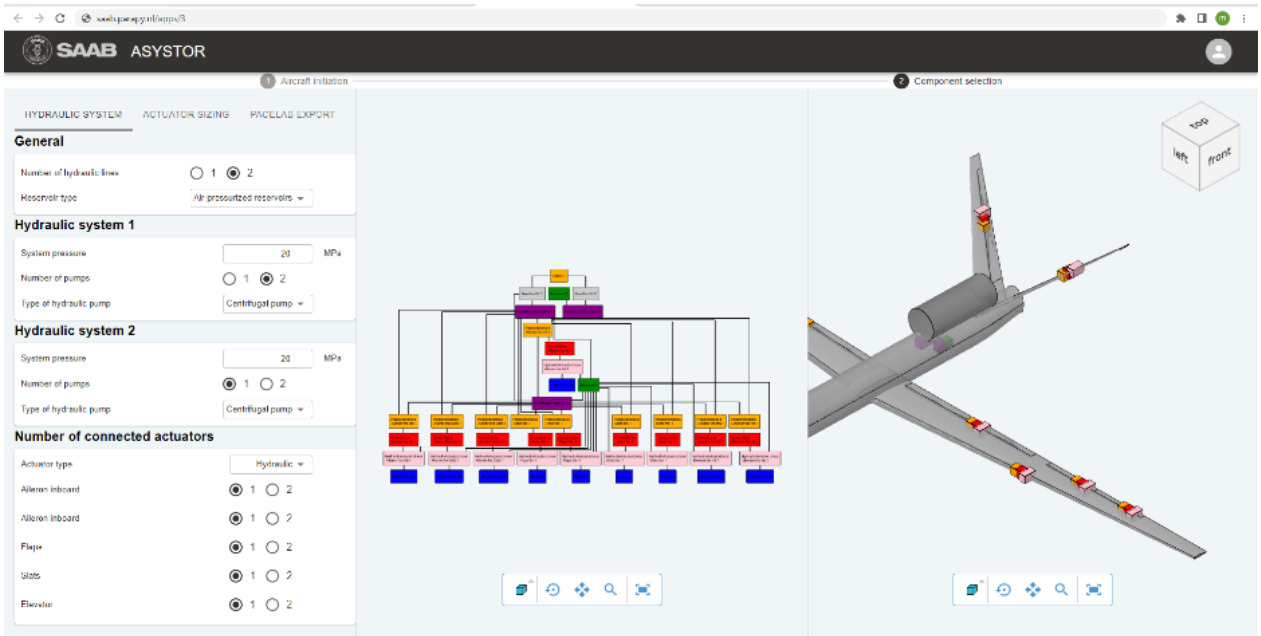

Figure 8: The systems´ components placement (right) in the aircraft.

DOI: 10.13009/EUCASS2023-613

3.3 Results and Discussion

The design process demonstrated provides various quantifiable and not-quantifiable measures addressing the design process efficiency, effort and design (result) improvements. The main improvements are:

I. Reduction of development and labour time, mainly due to the change of the design process by means of automation, thus reusing the invested (initial) modelling time (i.e., front-loading of the design process)
II. Increased number of design variants evaluated (within the same amount of time or effort)
III. Increase of the analysis credibility with the help of higher-fidelity analysis method (here: dynamic cyber-physical system simulation).

The process demonstrated basically removes most of the manual repetitive design tasks through automation. Instead, an implementation investment must be made in advance in terms of a) implementing the tailor-made KBE application for the problem at hand and b) retrieving and implementing the system architecture and system sizing rules. While the former can be done by engineers with programming experience, the latter is more of an iterative process involving system domain expert(s) until all relevant KBE rules have been identified, implemented and tuned to achieve the desired results.

Table 2 shows the KPIs addressed and achieved in the use case presented. The benchmark is based on Saab's current approach. With the implementation of the use case, the lead time was reduced by more than 95%. The reason why some time is still required (0.5 - 1 hour) is that the engineer is still in the loop. The engineer is constantly iterating the design variables in the user interface, making design decisions based on the project requirements and the results of previous analysis. With this acceleration in lead time, and the fact that engineers who lack some of the domain-specific knowledge of modelling systems can now still evaluate architectures (the knowledge rules that these inexperienced engineers lack are captured within the ASYSTOR application), a larger number of designs can be analysed by a larger group. This can even be taken a step further by automatically generating and evaluating all possible architectures (so there is no longer a human in the loop). Such a full-factorial approach leads to more than 3000 evaluations for the use case studied, which is a large increase compared to the 25 that are typically evaluated in a project.

Table 2: Key performance indices reached by the design automation on the use case.

KPI	Benchmark (2021)	Measured (2023-03)
Lead-time for design update [%]	8-40 [h]	0.5 – 1 [h]
Designs evaluated	25	3125

The realised design process focuses on the automation through information reuse. However, enabling the reuse by making the design rules and relationships available to the design composer requires an overhead in formulating the KBE rules. In the use case presented, this is done by an engineer familiar with the ParaPy software based on advice from target domain experts. It is not yet known whether this type of work can be performed in the future by a trained domain expert in cyber-physical product modelling, or whether this is a task that requires a new type of expert, dedicated to KBE and design automation. In the CAD domain, where KBE is an established state-of-the-art method, these tasks are performed by the designers themselves and no additional specialist is required during the design process.

4. Conclusions

This work has demonstrated that it is possible to explore large amounts of different architecture alternatives for onboard systems and to automate the time-domain modelling of onboard systems in the ACD phase. The ability of the proposed application to rapidly generate system architectures, geometrically position components and generate input files for simulation tools has resulted in a significant reduction in lead time, allowing more architecture alternatives to be explored compared to the current more traditional approach.

As with any kind of automation, questions arise about the initial overhead of implementation effort, flexibility and reusability (for other use cases), maintainability (may be hampered due to complexity), and overhead costs due to additional software and work steps. Looking at product development in general, a broad breakthrough of knowledge-based engineering for complex cyber-physical product modelling has not yet been reached. To achieve this, challenges related to an inefficient (from a human perspective, not from a computational perspective) way of defining rules and the lack of an adequate (graphical) representation of the implemented rules must first be overcome. This is not a tool or programming language specific shortcoming, but a logical consequence of the overwhelming complexity in a system/graph network of so many possible choices/branches and the inherent interconnection of these (architecture or sizing) rules by the underlying physical effects, the system interconnections and the influence of the control system.

DOI: 10.13009/EUCASS2023-613

Acknowledgement

The research presented in this paper has partially been performed in the framework of the DEFAINE (Design Exploration Framework based on AI for froNt-loaded Engineering) project and has received funding from ITEA 3 programme.

References

[1] S. Liscouët Hanke, "A Model-Based Methodology for Integrated Preliminary Sizing and Analysis of Aircraft Power System Architectures," Institut National des Sciences Appliquées (INSA) de Toulouse, Université De Toulouse, France, 2008.

[2] D. Scholz, "DOCsys - A Method to Evaluate Aircraft Systems," in *Workshop: DGLR Fachausschuß S2 - Luftfahrtsysteme, Deutsche Gesellschaft für Luft- und Raumfahrt*, München, Germany, 1998.

[3] H. Ounis, B. Sareni, X. Roboam and A. De Andrade, "Multi-level Integrated Optimal Design for Power Systems of More Electric Aircraft," *Mathematics and Computers in Simulation,* vol. 130, p. 223–235, 2016.

[4] AIR5005A, "Aerospace - Commercial Aircraft Hydraulic Systems," A-6A1 Commercial Aircraft Comittee, SAE International, 2010.

[5] R. Braun, P. Nordin, L. Ericson, L. V. Larsson, P. Krus and M. Pettersson, "Hopsan: an Open-Source Tool for Rapid Modelling and Simulation of Fluid and Mechatronic Systems," in *Proceedings of the BATH/ASME 2020 Symposium on Fluid Power and Motion Control*, Bath, United Kingdom, 2020.

[6] M. Alder, E. Moerland, J. Jepsen and B. Nagel, "Recent Advances in Establishing a Common Language for Aircraft Design with CPACS," in *Aerospace Europe Conference 2020*, Bordeaux, France, 2020.

[7] T. Lwin, J.-W. Lee, S. Kim, Y.-H. Byun and N. V. Nguyen, "Advanced Aircraft Design - Certification Framework with Database Development," in *Tenth International Conference on Computer Applications (ICCA)*, Myanmar, 2012.

[8] L. Knöös Franzén, "A System of Systems View in Early Product Development: An Ontology-Based Approach," Linköping Studies in Science and Technology, Dissertations, Linköping, Sweden, 2023.

[9] S. Gradel, P. Hansmann and E. Stumpf, "SystemXF: a Novel Approach for Holistic System Modeling in Aircraft Conceptual Design," *CEAS Aeronautical Journal, Springer ,* vol. 14, p. 57–73, December 2022.

[10] ITEA4, *The DEFAINE project,* online, URL: https://www.defaine.eu/ (accessed 2023-05-07), 2023.

[11] ParaPy, "ParaPy KBE Software Development Kit," ParaPy B.V., 2023. [Online]. Available: https://parapy.nl/. [Accessed 29 06 2023].

[12] I. Staack and P. Krus, "Integration of On-Board Power Systems Simulation in Conceptual Aircraft Design," in *Proceedings of the 4th CEAS European Air & Space Conference*, Linköping, Sweden, 2013.

[13] M. Axin, R. Braun, A. Dell'Amico, B. Eriksson, P. Nordin, K. Pettersson, I. Staack and P. Krus, "Next Generation Simulation Software using Transmission Line Elements," in *Fluid Power and Motion Control (FMPC), Bath, UK*, 2010.