33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4-9 SEPTEMBER, 2022

ICAS 2022 SWEDEN

SYSTEMS ARCHITECTING ASSISTANT (SARA) – ENABLING A SEAMLESS PROCESS CHAIN FROM REQUIREMENTS TO OVERALL SYSTEMS DESIGN

Nils Kuelper[1], Jasmin Broehan[1], Thimo Bielsky[1] & Frank Thielecke[1]

[1]Hamburg University of Technology, Institute of Aircraft Systems Engineering, Nesspriel 5, 21129 Hamburg, Germany

Abstract

The path towards climate neutral aviation necessitates the development of novel aircraft concepts with disruptive technological solutions on board. The integration of such innovative technology concepts, such as hydrogen-based propulsion, has major effects on the aircraft's overall systems architecture. These integration-related effects are investigated in the scope of technology concept studies for which suitable systems architecture variants need to be created and evaluated beforehand. The knowledge-based systems architecting methodology *SArA* is proposed to ensure a seamless process chain during aircraft conceptual design starting with the definition of top-level aircraft requirements and ending with the execution of technology concept studies on overall systems design level. Different approaches for analyzing a given design problem within *SArA* are discussed: a top-level requirements-based approach, a system-level-based approach considering the individual systems, and a complementary functions-based approach. Moreover, an exploration of the architectural design space is integrated, and architecture variants are created and evaluated by employing a knowledge-based approach. *SArA* is then applied by exploring the design space of the electric power supply system for a hydrogen-powered concept aircraft, considering four different architecture variants. To this end, central and distributed layouts as well as different voltage specifications are discussed. The four architectures variants are evaluated based on criteria like complexity and risk.

Keywords: aircraft, hydrogen, architecture, systems architecting

1. Introduction

The known effects of aviation on climate change drive ongoing research activities in order to support ecologically sustainable aviation [1]. Environmental requirements, such as, emission reduction targets for CO_2 and NO_x defined by the European Union within *FlightPath 2050* [23], are pushing state-of-the-art aircraft on-board systems architectures towards their optimization limits. The predicted evolutionary improvement of currently used technologies will not reduce emissions to the required extent as mentioned in *FlightPath 2050* [1, 23]. The integration of disruptive technologies into systems architectures poses a possible approach for increasing aircraft overall efficiencies and thus, reducing aircraft fuel consumption [1]. However, merely integrating novel technologies into existing systems architectures will most likely not be sufficient to reach the emission reduction targets. New concepts for power train and on-board system power supply architectures need to be investigated. Such concepts include, for example, the electrification of the power train to various extents: hybrid or fully-electric. Hydrogen-based fuel cell systems pose a promising concept for regional and short-range aircraft to fulfill novel environmental requirements, given that hydrogen as the main energy source is climate neutrally produced [1, 2].

New concepts for power train and on-board system power supply architectures are investigated during the aircraft conceptual design phase. This phase is characterized by a vast design space consisting of many elements and several conceivable technological and combinatorial solutions [27, 33].

However, little information about the design problem are available to the engineer [33]. Decisions during conceptual design influence systems performance more than decisions during detailed systems design (DSD) [41]. On the one hand, handling complexity poses a challenge during conceptual design due to a large amount of system elements and interrelations between systems as well as interrelations between components of a system [4]. Especially for disruptive technologies, such as fuel cells or for electrified on-board system power supply architectures, the underlying complexity is not yet fully understood. New system interrelations arise and effects on on-board systems need to be investigated. For example, the effects on redundancies of safety critical aircraft on-board systems like the electric power supply of components from the hydrogen supply system such as pumps need to be analyzed. Furthermore, the ongoing digitalization of aircraft systems leads to a significant increase in system functionalities and interdependencies, resulting in an even higher system complexity. On the other hand, conceptual design is influenced by uncertainty, as initial assumptions are not yet validated and little experience and few reference systems for disruptive technologies are available [4]. Moreover, certification specifications for disruptive technologies like fuel cells or handling of hydrogen are not yet fully defined by the authorities [28].

Systems architecting is an integral part during aircraft conceptual design which compromises several design steps, including overall aircraft design (OAD) and overall systems design (OSD). Currently, the definition of systems architectures is a rigid process of considering either only one or hundreds of solutions [9, 8]. Systems architecting is not included as an independent step within the design process and thus leaving a gap between OAD and OSD. Furthermore, compliance with safety and reliability requirements needs to be validated already during systems architecting. To obtain promising results from technology concept studies, the design space needs to be explored in a reasonable way, suitable architecture variants need to be efficiently selected, and early validation of conceptual safety requirements needs to be conducted before OSD. In general, OSD is a time-consuming process, hence pre-selected architecture variants can reduce development time. To ensure a seamless process from OAD to OSD, a formalized methodology for assisting the engineer in creating and evaluating systems architecture variants is proposed in this paper: the *Systems Architecting Assistant* (*SArA*).

Section 2 of this paper gives a description of different types of systems architectures, states the current process chain explaining the mentioned gap in further detail, and provides a literature review of state-of-the-art systems architecting approaches. In Section 3, the *SArA* methodology is presented in depth. Section 4 provides a description of the application of SArA for evaluating different architecture layouts of the electric power supply system (EPSS) for a hydrogen-powered regional concept aircraft. Section 5 concludes with a brief summary and an outlook for upcoming research activities.

2. State-of-the-Art Systems Architecting

The objective of systems architecting is to investigate the design space during aircraft conceptual design and to provide valid architecture variants to OSD which can then be further analyzed by performing technology concept studies. During technology concept studies of OSD, systems topologies are created and system components are sized to estimate system masses and power demands to compare different technology concepts. In general, technology concept studies are performed on a higher level of detail than systems architecting. To understand and perform systems architecting during conceptual design, it is first necessary to define the terms system, architecture, and architecting. Moreover, providing the foundation of the *SArA* methodology, current systems architecting approaches are discussed. Lastly, the adapted, seamless process chain during conceptual design is presented.

2.1 Terms and Definitions in Systems Architecting

In literature, definitions for ‘system’ are provided by, e.g., the International Council of Systems Engineering (INCOSE) [29] and Kececioglu [35]. Building up on these definitions, a *system* is defined in this paper as *an organized collection of components, such as a physical equipment, an atomic software task, or a container of subsystems. These components are interconnected to fulfill one or more high-level system functions*.

For the definition of the term ‘systems architecture’, the most important literature references are shown first. Armstrong [7], Bornholdt [9], and Maier [40] similarly define an architecture as a structure of elements or entities whose attributes are needed to fulfill a certain task or function and the interrelation and communication between those entities. INCOSE [29] defines architecture in a more general matter. It describes a process or a structure including components, interfaces, constraints, and system behavior. Furthermore, similar definitions are provided by the Department of Defense (DoD) [18], de Tenorio [19], Liscouet-Hanke [39], and Ulrich et al. [52]. The definition for **system architecture** as used in this paper describes *a system architecture as all elements or components and their interdependencies required to fulfill an objective. An architecture may contain functions, logical, and physical components and may describe components within one system (system architecture) but may also include components and interactions of multiple systems (systems architecture)*.

As stated, an architecture may contain functional, logical or physical elements to describe different levels of abstraction. De Tenorio [19], Kleiner et al. [37], and MathWorks [51] define the high abstraction level as functional architecture or functional design focusing on functions and their interrelations. A definition based on the ARCADIA methodology [53] is provided by the Eclipse Foundation [21]. Hereafter, the high-level abstraction is referred to as **functional architecture** and describes *a system as a decomposition of system and subsystem functions, which are needed to accomplish an objective, and their interdependencies* as outlined in Figure 1. For example, a system function may describe the necessity to *generate high lift*.

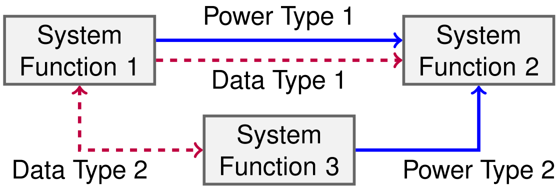


Figure 1 – Generic functional architecture.

For the intermediate level of architecture abstraction, several denominations and definitions are used. The Eclipse Foundation [21], Kleiner et al. [37], and MATHWORKS [51], similarly define this level as logical architecture or logical design. In this paper, the intermediate level of abstraction is called **logical architecture** and is characterized by *a system description based on high-level components and their data and power exchange. It demonstrates how intended functions are realized using logical component models. A detailed physical technology selection on component level is not yet conducted, but pre-selection of certain technology bricks may be included*. For example, *fowler flaps* may be identified as a valid technology to *generate high lift*. A more detailed, physical implementation is not performed. This definition is outlined in Figure 2 demonstrating the required logical components to fulfill *system function 1* (cf. Figure 1).

The third systems architecture abstraction level consists of medium to high fidelity models including executable behavior models and systems simulations. The Eclipse Foundation [21], Kleiner et al. [37], and MATHWORKS [51] define this level as physical architecture or design focusing on physical hardware needed to describe the system and its explicit behavior. In this paper, the low-level architecture abstraction is called **physical architecture** and describes *a system in terms of medium to high fidelity behavior models and systems simulations of physical hardware components based on an extensive technology selection. Detailed information about element interrelations and communications, such as a technology selection of the communication protocol of data exchange or, for example, a cable for power transfer, are considered. Components and connections in combination fulfill system functions. These functions may be validated by systems simulations*. This definition for *system function 1* (cf. Figure 1) is schematically shown in Figure 3. An example for components

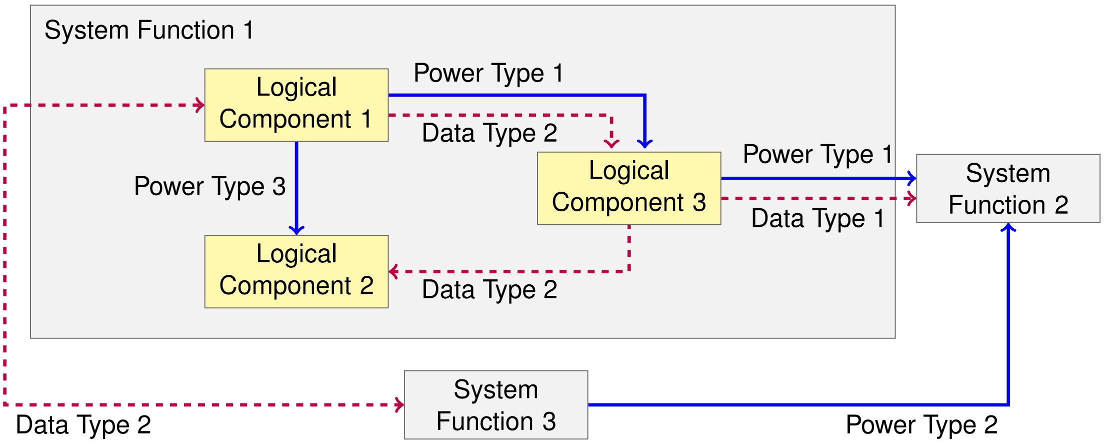


Figure 2 – Generic logical architecture.

of a physical architecture are *fowler flaps* considering detailed information about the functional and technological concept, such as the motion path, as well as a behavior simulation. The physical architecture goes beyond the in this paper presented systems architecting methodology and is stated here only for completeness.

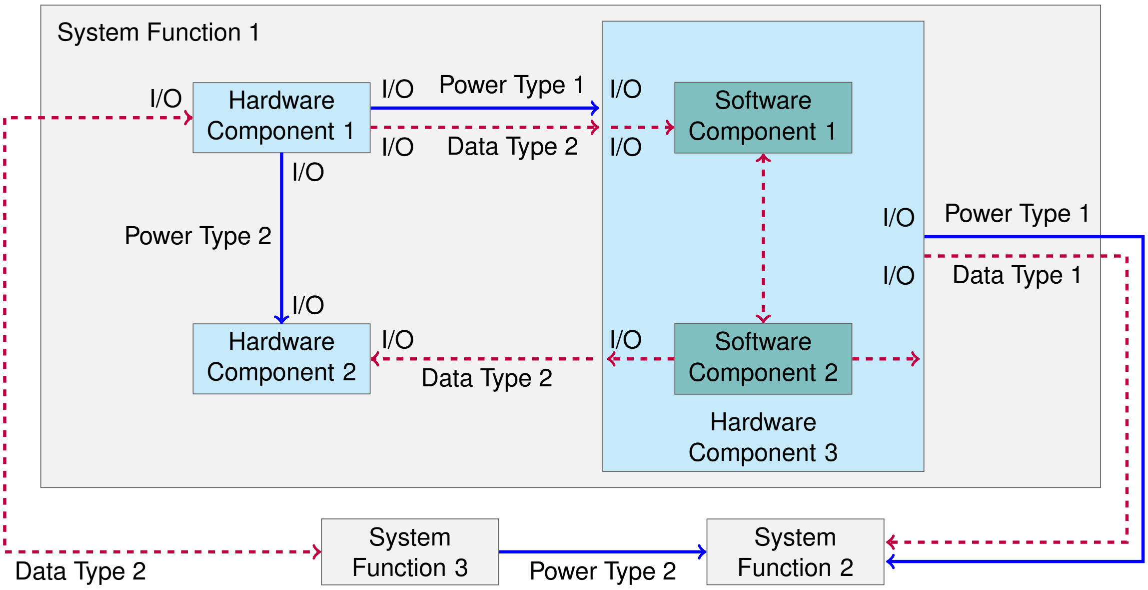


Figure 3 – Generic physical architecture.

Systems architecting in general includes the process of efficiently creating and selecting promising systems architecture variants to fulfill necessary system functions based on a vast design space of alternatives [34]. Armstrong [7], INCOSE [29], Liscouet-Hanke [39], and Maier et al. [41] extend this general definition similarly. In this paper, **systems architecting** for conceptual design is defined as *the non-automated and creative process, during which the engineer defines architectures on two levels of abstraction: functional and logical. The process consists of analyzing the given design problem based on requirements, design space exploration, generation of architecture variants, and evaluating those variants to provide only promising variants to OSD*. Within SArA, systems architectures are considered as promising variants if it can be estimated that certain performance requirements in terms of, for instance, mass, power demand, direct operating cost, and component degradation are satisfied during technology concept studies.

Besides defining terms associated with systems architecting, the authors like to point out **systems topology** as a term which is not part of systems architecting. Topology can be seen as an extension of architectures by considering location-oriented, geometrical parameters, and spacial arrangements of system components and its connections [9, 39].

2.2 Review of Existing Systems Architecting Approaches

To create and evaluate aircraft systems architectures during conceptual design, several systems architecting approaches have already been proposed which are used as a basis for *SArA*. One approach has been introduced by Crawley et al. [17] stating the importance of systems architectures for understanding, designing, and managing complex systems. Four levels of architecture abstraction are proposed: functional, physical, technical, and dynamically-operational. Richards et al. [46] state the necessity to perform a design space exploration to prevent the identification of only locally optimal solutions. Therefore, an understanding of the underlying relationships is essential. Methods for conceptual design based on available knowledge, such as physical laws and geometric models, are proposed by Pfennig [44]. Since problem definition is essential for problem-solving, he states that an engineer needs to be supported during this manual step, whereas solving the problem can be automated. Garriga et al. [27] provide a modeling framework for systems architecting and integrate feasibility as a metric to analyze fulfillment of system requirements, but also the competitiveness of architectures in terms of mass and power demands. Parameter and physics-based methods are proposed by Liscouet-Hanke [39] to create architectures and perform technology concept studies while considering system functions based on logic trees.

Bornholdt [9] elaborates on a function-based architecting approach for defining systems architecture variants and explains methods of a functional decomposition. Safety and reliability analyses are automatically performed to reduce the design space and to numerically identify an optimal architecture solution in terms of dispatch reliability and system mass. Another function-based approach is proposed by Armstrong [6] to assist the engineer in flexibly exploring the design space. Architecture variants are automatically generated considering constraints and cross system effects. The approach is extended by Armstrong [7] by focusing on safety and reliability aspects of systems architectures to consider off-nominal behavior. Load-shedding, system safety, and functional hazard assessments are considered to identify architecture specific critical requirements. Judt et al. [32, 33] proposed a function-based, automated systems architecting approach considering architectural degrees of freedom which are analyzed on aircraft, architecture, and component level. The generation of systems architectures is based on a numerical optimization. However, the engineer is needed to manually select system technologies. The knowledge with respect to technological solutions is organized in a database. De Tenorio [19] focuses on model-based methods for collaborative, knowledge-based systems architecting using the systems modeling language (SysML). Agile problem decomposition methods are proposed which are selected depending on the context: A top-down approach for conventional and a bottom-up approach for innovative architectures are integrated. An optimization algorithm sizes the systems on architecture level.

An approach for a numerical, multi-criteria optimization of avionics architectures based on binary programming (BP) is proposed by Annighöfer [5]. Raksch [45] proposes multi-criteria optimization algorithms for redundancy allocation to develop fault-tolerant systems architectures. Another multi-objective optimization approach for systems architecting considering safety, reliability, costs, and weight is proposed by Johansson et al. [31]. However, reliability analyzes need to be performed manually by the engineer. Using a multi-objective, mixed-integer optimization for architecture selection, Bussemaker et al. [13] focus on a method for systematically modeling and analyzing the design space using graph theory. The approach is extended by Bussemaker et al. [14] proposing a formalized architecture design space description as enabler for optimization. Multiple approaches as means for architecture generation, such as full enumeration of the design space, design of experiments, and optimization, are discussed. Finn et al. [24] propose a generic architecture optimization algorithm for cyber-physical systems considering a design space exploration based on a component library.

To determine the effects of systems architectures on aircraft level during conceptual design, an approach based on minimal architecture size and parameter robustness against uncertainty on system level is provided by Jackson [30]. Chakraborty [15] developed an integrated, parallel, and modular approach for sizing and adapting both systems and aircraft. System elements are automatically connected based on heuristic-based methods considering redundancy requirements and concepts of existing aircraft. Architecture variants are saved as a matrix of alternatives (MoA). Engineers are required to manually identify impractical variants with a high complexity or technology uncertainty. Fioriti et al. [25] propose an approach for automated architecture generation based on a full design space enumeration and a selection of architecture variants based on costs, mass, and fuel consumption. A knowledge-based pre-processing step is required to exclude less promising architecture variants from further consideration.

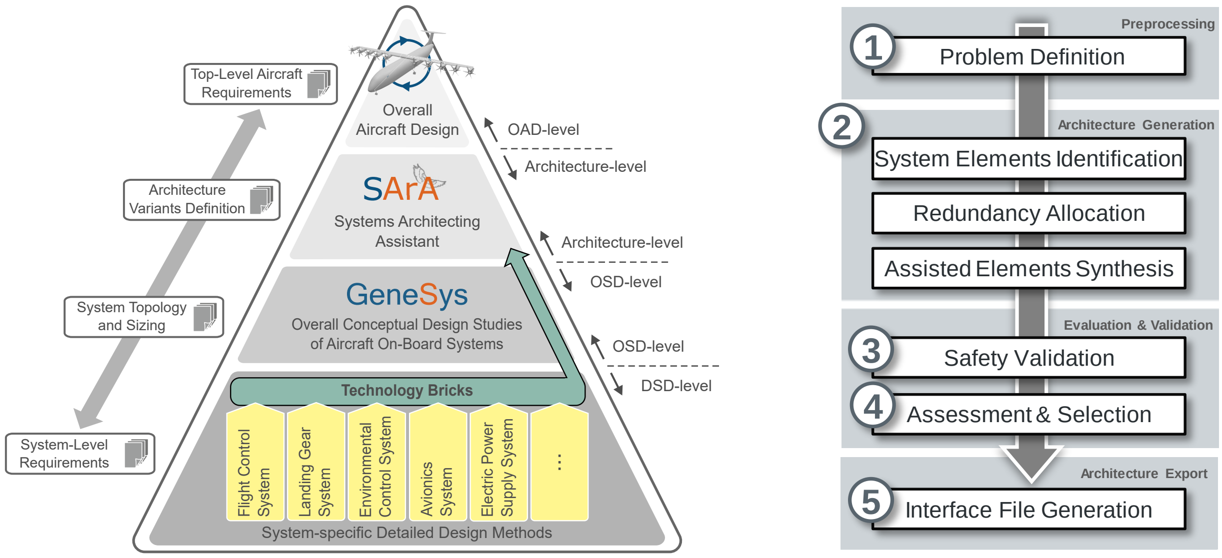
The presented existing systems architecting approaches are the foundation of the methods within *SArA* to assist the engineer during systems architecting. However, of the presented approaches only practical methods are considered for *SArA*. Rigid methods include the use of a functional decomposition, as described, for example, by Bornholdt [9], even though it includes several drawbacks and is often considered to be less intuitive [19]. However, it may be helpful for the evaluation of novel technologies as little experience is available, and system functions need to be identified. Furthermore, methods like full enumeration or optimization of the architecture design space are proposed, accepting drawbacks, such as an over-examination of the design space and thus, considering many non-beneficial systems architecture variants which would directly be rejected by an experienced systems engineer [34]. Consequently, high implementation efforts, computational power, and development time are generally necessary [9, 34]. A more efficient approach focuses on a knowledge-based, manual systems architecture definition procedure performed by experienced systems engineers [34]. Based on a small amount of available information and few validated assumptions, architecture validation and evaluation are two key challenges during systems architecting. Presented approaches, e.g., by Judt et al. [33] do not consider only systems architecting process steps but move towards OSD-level by integrating system sizing approaches and medium fidelity simulations. However, a clear separation between conceptual design levels, as shown in Figure 4a, is desirable to enable process steps which can be performed independently.

2.3 Adapted Process Chain

To ensure a seamless process chain and to enable technology concept studies on OSD-level, the systems architecting methodology *SArA* is integrated within the conceptual design process between OAD-level and OSD-level, as shown in Figure 4a.

Conceptual design starts at defining geometric characteristics and top-level aircraft requirements (TLARs) on OAD-level. The set of typical TLARs like flight altitude, payload, and range should be extended by low-emission requirements to support the sustainability goals [22]. Based on TLARs requirements on OSD-level, such as secondary power demands, are identified [22, 27]. For TLARs management the Common Parametric Aircraft Configuration Schema (CPACS) defined by the German Aerospace Center (DLR) [3] with respect to standardized, interdisciplinary aircraft design is employed in the scope of SArA. CPACS is also used as an interface for other design disciplines to provide parametric geometry information and TLARs.

To validate assumptions and to identify the impact of promising systems architecture variants on aircraft performance, such as mass, energy demands, and pollutant emissions, technology concept studies are performed as part of OSD. A parametric and integrated approach to support system engineers in conducting such studies for systems design is provided by the *GeneSys* framework which was developed at the Institute of Aircraft Systems Engineering (FST) at Hamburg University of Technology (TUHH) [8, 9, 34]. The framework consists of several modules, such as systems topology generation or systems sizing. In general, OSD is a time-consuming process, hence well-defined and pre-selected variants can save iteration runs and development time during aircraft conceptual design.



(a) Placement of *SArA* in the conceptual aircraft design phase.

(b) Process of *SArA*.

Figure 4 – Placement and systems architecting process of *SArA*.

3. Systems Architecting Assistant – SArA

To ensure the seamless process chain during conceptual design by closing the gap between OAD and OSD, the systems architecting assistant methodology *SArA* including a formalized process for systems architecting is proposed. In general, aircraft systems architectures can be designed based on an *evolutionary approach*, i.e., by considering already existing architectures as a baseline and performing slight changes to them with the goal to improve overall systems efficiency. Since starting at aircraft or architecture level, at which TLARs are considered, the *evolutionary approach* can be seen as a top-down approach [7, 19]. Whereas a bottom-up approach or *revolutionary design* starts at component level based on an already performed selection of technologies. This approach is well suited to integrate disruptive technologies into systems architecture variants which typically causes a fundamental redesign of many interfacing systems [7, 19]. Within systems architecting, it is necessary to include both the top-down and the bottom-up approach to perform evolutionary changes to architectures with respect to the integration of novel technologies. Therefore, both approaches are supported by *SArA*. To integrate methods for handling system complexity and uncertainty of aircraft on-board systems, which can be safety critical, such as the EPSS, a model-based approach is used.

3.1 Problem Definition

To identify valid and promising systems architectures for conventional and novel aircraft concepts, it is necessary to understand and handle requirements, thus analyzing a given systems architecting problem [44].

Evolutionary Design

As the first process step within the SArA methodology, the architecting problem is defined as shown in Figure 4b. To this end, the *evolutionary design* is based on the TLARs. The exploration of the systems architecture design space starts by examining systems architectures of existing aircraft and their applicability to the defined problem formulation, i.e., fulfilling TLARs and derived system requirements.

If the problem definition implies the alteration of only one particular system, such as the propulsion system which, for instance, may be changed from a conventional turbofan engine to a fuel cell system, the effects on other systems are identified. In this example, due to the omission of bleed air, a conventional environmental control system poses an infeasible solution. Thus, a different technological solution on system level is identified. A systems-based approach, where the categorization of individual systems follow the ATA chapter definitions, is beneficial for the search of already existing technology concepts on system level. In general, if an existing systems architecture already poses a valid solution for an identified conceptual design problem, it may act as a baseline for other newly created architecture variants during the evaluation step of the *SArA* process. For any baseline architecture identified, the step of architecture generation within *SArA* is skipped. Furthermore, the validation step of *SArA* of the baseline architecture is skipped as well, since safety and reliability have already been validated once the architecture was certified as part of an aircraft entry-into-service process. However, since component reliability may improve over time, new validation may prove beneficial and is recommended.

Revolutionary Design

In general, a function-based approach is considered to be part of the *revolutionary design*. For disruptive technologies, it is likely that neither on aircraft level based on TLARs nor on system level a valid, existing solution can be identified, since not all requirements are yet known. In this case, a functional decomposition (function-based approach) is performed to identify the functions and requirements a certain novel technology needs to fulfill. It is also a solution-bias free method by not considering technologies but only functions [14]. However, this approach is only considered as a complementary problem decomposition method for *SArA* because it is less intuitive [19], rigid, and based on a subjective interpretation of the underlying system functions. Moreover, higher amount of work is needed [9].

3.2 Generation of Architecture Variants

As a second step of *SArA*, an exploration of the design space is performed. This exploration assists the engineer with the generation of systems architecture variants. This reduces the risk of both being limited to local solutions in the solution space and being limited to expert-biased design solutions [46]. Within *SArA*, the design space exploration and the generation of systems architecture variants is a knowledge-based and heuristic approach. Parts of existing and known systems architectures are used as starting point during architecture generation. These partial architectures are complemented by system components which are based on given technology bricks. Technology bricks describe partial or full technology concepts. They are developed in detail during DSD and are incorporated during systems architecting as abstracted logical technology bricks (cf. Figure 4a). Differing technological solutions can be displayed using a matrix of alternatives (MoA) [15], as presented exemplary in Table 1.

Table 1 – Exemplary inputs for a matrix of alternatives for technological solutions for a set of given logical system elements.

Logical system elements	**Technological Alternative**
Secondary power type	hydraulic, electric, pneumatic, mechanical
Nominal power supply source	auxiliary power unit (APU), fuel cell system (FuCS), battery
Emergency power supply source	APU, FuCS, battery, ram air turbine (RAT)
Energy source	kerosene, synthetic fuel, hydrogen

To fulfill given requirements, a single technological alternative may be selected from the MoA. In addition, a combination of multiple technology concepts may be a valid solution. For example, power for aircraft on-board systems can be provided either as a combination of electric, hydraulic, and pneumatic power or fully-electrically. Information, which technology or component fulfills necessary functions and requirements, can be stored within a database [24, 33]. However, this is not considered in this paper and is subject to further investigations. Besides conventional and established technology concepts, also disruptive technologies are considered within the MoA. Thus, a function-based approach as part of the *revolutionary design* is necessary to understand which technology bricks can also be implemented as alternative variants.

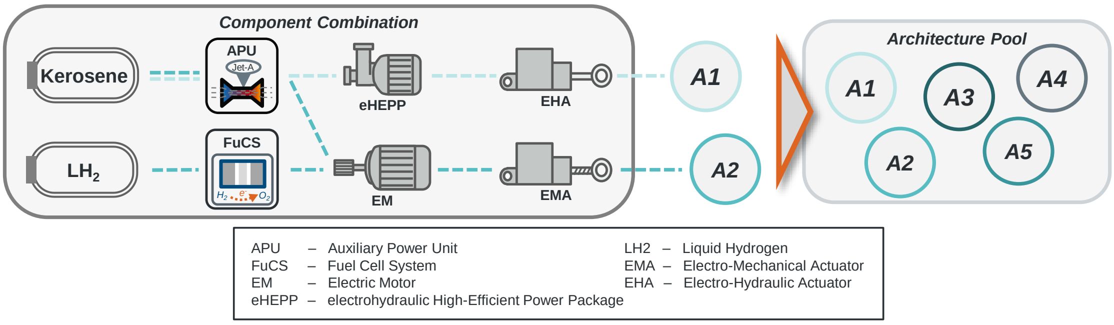


Figure 5 – Generation of pool of systems architecture variants based on system components within *SArA*.

As shown in Figure 5, systems engineers create architecture variants (e.g. *A2*) based on model-based system elements and pre-selected technologies. While architecture generation is performed, component redundancies based on experience and existing architectures are considered. However, further research in this field is necessary to provide a feasible and adequate, possibly automated method for performing redundancy checks. The manually generated architectures are combined into a pool of possible architecture variants. To further understand the impacts of certain technology selections, extreme architecture variants, for instance, strongly conservative, purely disruptive, and highly redundant, are investigated. In addition, also knowledge-based architecture variants are created. The goal of this process step of *SArA* is to create systems architecture variants in order to set the basis for validation and evaluation. The amount of generated variants strongly depends on the given problem and the trade study capacity in terms of computational resources and time.

SArA focuses on a knowledge-based approach to benefit from systems engineers' knowledge and experience and to ensure high efficiency and effectiveness because the solution space does not have to be spanned over impractical variants. Even though experts decisions can be biased, subjective, or conservative [13], alternatives to the knowledge-based approach have several drawbacks as well: Full enumeration of the design space is only feasible if the amount of valid architectures is around a few hundred solutions or the generation of a single variant requires only a few seconds [14]. Otherwise, thousands of possible solutions need to be generated and later assessed requiring high efforts, computational power, and time [9]. Furthermore, many non-beneficial systems architectures are considered. Other approaches, which require optimization algorithms, may be more feasible than a full enumeration of the design space, but still require high development and implementation efforts for the optimization algorithm. Furthermore, the traceability of generated design solutions to given requirements might pose a significant challenge. This results in high computational power and calculation times [5]. Moreover, the resulting architecture variants after optimization are often similar to existing systems architectures or those found by experts within a significantly shorter period of time [14, 34].

3.3 Validation of Architecture Variants

Each previously generated architecture variant within the architecture pool needs to be validated to demonstrate compliance with safety requirements. Before architecture validation, it is not guaranteed if architecture variants fulfill all constraints [14]. To obtain only valid architecture variants for further investigation, non-compliant variants need to be excluded from further consideration, as illustrated in Figure 6. Within *SArA*, currently a functional hazard assessment (FHA) and a preliminary system safety analysis (PSSA) need to be performed manually according to SAE ARP 4761 [50] for architecture validation. However, to elaborate a feasible, adequate, and automated method for performing validation within *SArA*, further research in this field is necessary. Elaborated redundancy and automated reliability approaches have been proposed by Raksch [45] and by Bornholdt [9].

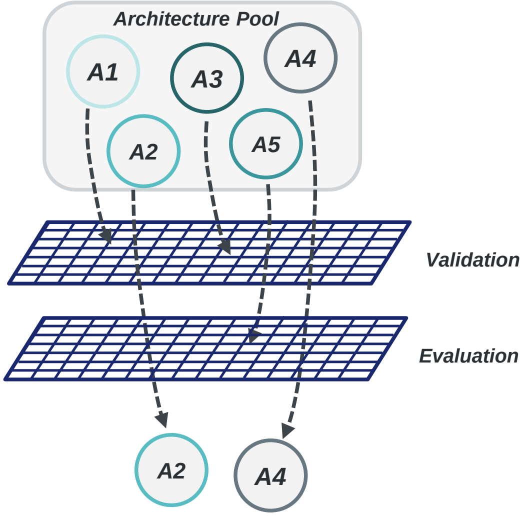


Figure 6 – Systems architecture variants validation and evaluation within *SArA*.

3.4 Evaluating Architecture Variants

After having a set of validated architectures that fulfill all system requirements and boundary conditions, an evaluation is conducted to rank the remaining architecture variants and to specify which variants should be investigated first. This can also help to further reduce the architecture pool, for example, in case architectures meet all requirements but would be too costly to implement. Usually, simulation is performed to evaluate a system's behavior and deduce its performance. However, during conceptual design, high fidelity architectures and simulation models are not yet available, especially when technologies with a high amount of uncertainty are considered. This lack of model fidelity of logical architectures increases the difficulty to measure regular key performance indicators (KPIs) such as mass, or costs. Therefore, a metric is needed that serves as a substitute measurement for parameters that are difficult to quantify. One possible approach is the usage of architecture proxy metrics [12]. According to Bussemaker et al. [12], architecture complexity and component technology readiness levels (TRLs) can be used as proxy metrics to measure costs and risk, respectively.

Maier et al. [41] define complexity as the amount and types of interrelationships among elements in a system, indirectly measuring the difficulty to design, build, and use a system. Therefore, complexity is seen as a substitute measurement for design, manufacturing, and operating costs. TRLs are used to calculate the risk of change propagation. For instance, a component associated with a low TRL has a higher risk of being exchanged in later design iterations and therefore inducing change to its interfacing components. These changes can later on lead to design iterations and cause high effort and costs, especially if many other components are affected.

Usually, complexity is assessed with experience of systems engineers [36]. However, to accelerate the systems architecting process, this experience shall be formalized into a method for automated feedback while evaluating a systems architecture. For measuring complexity, the most intuitive concept is to count the amount of elements of a system N_e, the amount of types these elements consist of N_t, and the amount of interfaces between elements N_i. It should be noted, that the three categories are usually not equally weighted [36]. Introducing a new element type, such as a component with a new part number, can lead to higher design and operating costs compared to introducing another instance of a component, which has already been used within the systems architecture. Developing an optimal weighted complexity metric holds potential for further research. Especially, due to the fact that it can be considered a rather intuition- or knowledge-based metric, which makes the formulation of a strict law challenging. One approach could be the use of fuzzy logic, as it translates experience-based knowledge and human reasoning in the form of logical rules [10].

To evaluate a logical architecture's risk of major design iterations, Garg et al. [26] state that the relationship between component TRLs and risk is established by calculating the component risk r_c as

$$r_\mathrm{c} = l_\mathrm{c} \cdot i_\mathrm{c} \tag{1}$$

with likelihood of change l_c and severity of impact i_c. As a scale for measuring the likelihood that a modification of a component's design is necessary or even a different technology selection is needed, NASA's classification of TRLs can be used (cf. Table 2). One approach for measuring the severity of impact is to estimate the potential for components that undergo a design change to propagate further design changes to other components. When a formal logical architecture model is available, each component's amount of interfaces to other components can be used as an impact factor. Therefore, the severity of impact scales linearly with the amount of each component's interfaces. Thus, it is concluded that components with higher connectivity, i.e., a higher amount of interfaces, have a higher risk to propagate change [26] and lead to major design iterations. To model these component interconnections, design structure matrices (DSMs) [16] or graphical MBSE tools can be used. After calculating a risk value for each component according to Equation 1, a mean risk $\overline{r_a}$ for the overall architecture is estimated. Moreover, all component risk values are used to create visual risk graphs indicating high risk nodes in complex architectures.

In conclusion, the goal is not to find an optimal architecture, but to identify a set of architectures that corresponds to a pareto-front. This shall allow pre-sorting before possible time-consuming system architecture studies as well as technology concept studies on OSD-level. The presented metrics exhaust the potential for evaluation of logical architectures by no means. One aspect that needs further investigation in particular is a generic metric for architecture potential as the two presented metrics may penalize disruptive and innovative technologies.

Table 2 – Technological Readiness Levels adapted from [42].

TRL	Definition
9	Actual system “flight proven” through successful mission operations
8	Actual system completed and “flight qualified” through test and demonstration
7	System prototype demonstration in a target environment
6	System/subsystem model or prototype demonstration in a relevant environment
5	Component and/or breadboard validation in relevant environment
4	Component and/or breadboard validation in laboratory environment
3	Analytical and experimental critical function and/or characteristic proof-of-concept
2	Technology concept and/or application formulated
1	Basic principles observed and reported

3.5 Interface to Overall Systems Design

Although *SArA* was developed to support systems architectures composed of functional, logical, and physical components, the focus lies on logical systems architecture variants. OSD-level tools, such as the *GeneSys* framework, require logical systems architecture variants to perform technology concept studies (cf. Figure 4a). Logical, model-based architecture variants enable easy modification of components, connections, and properties representing system parameters. In addition, by supporting systems architectures based on functional components within *SArA*, the integration of functional decomposition results is enabled. Furthermore, the complementary consideration of systems architectures composed of physical components establishes a feedback loop from DSD. This enables the integration of technology bricks with higher levels of fidelity to decrease uncertainty of logical systems architectures.

To ensure a seamless process chain from *SArA* to OSD, a parametric, generic interface template is necessary to exchange relevant design and architecture information. The interface file includes a parametric systems architecture definition containing both meta data and properties of system components and connections. A schematic, exemplary representation of the interface file is shown in Figure 7. Via this dedicated text-based XML-template, valid systems architecture variants are accessed by OSD frameworks to perform sizing and steady-state simulation of aircraft on-board systems. Besides providing an interface to OSD, the XML-file can be used to save valid and promising systems architecture variants and its meta information. Thus, OSD can be performed more independently of *SArA*.

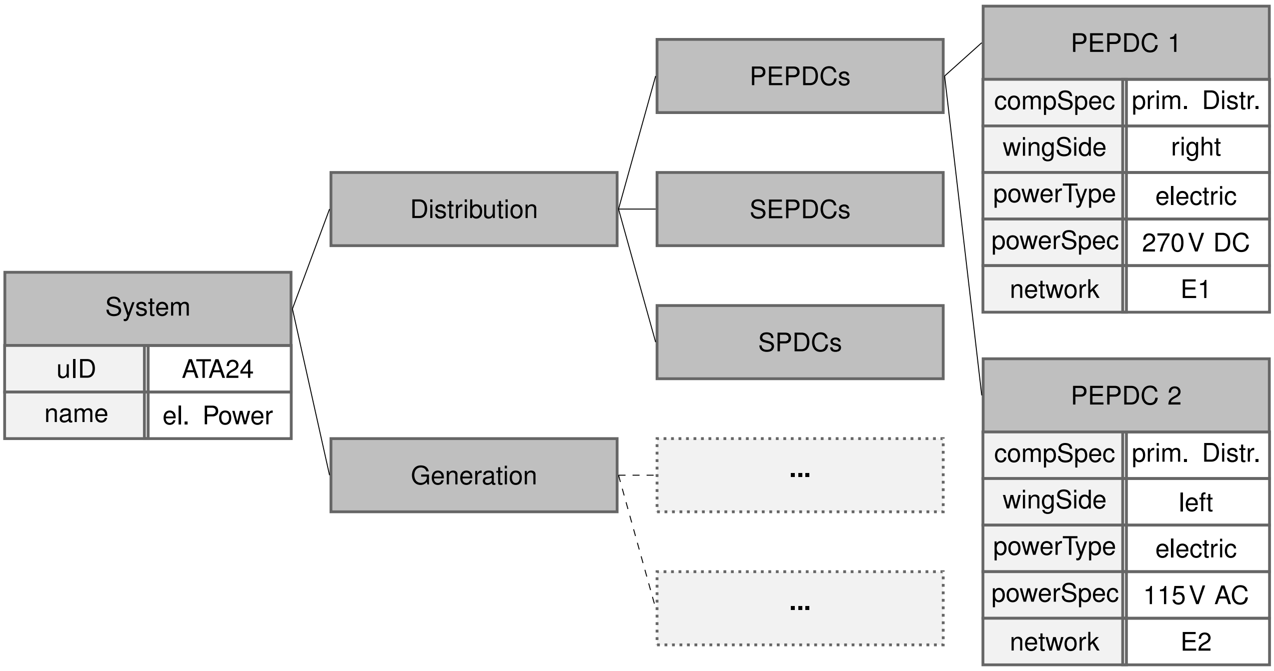


Figure 7 – Exemplary schematic excerpt of the XML-interface file from *SArA* to OSD for the EPSS.

Figure 7 highlights the hierarchical structure of the interface file, based on a previously created logical systems architecture example within *SArA*. Towards the right of the shown tree structure, a lower level with respect to the architecture decomposition is implied. Attributes of architecture components and their connections are presented for an excerpt of an EPSS. The listed attributes within the interface file provide information about component and connection properties for each logical element, such as network and power information. For example, the listed components *PEPDC 1* and *PEPDC 2* are supplied by *electric* power at 270 V DC and 115 V AC by different networks. With the information about components, connections, and properties, the systems topology can be created using the OSD framework *GeneSys*.

4. Case Study

The above mentioned methods for systems architecting are applied in the scope of the definition and evaluation of architectures for the electric power supply system (EPSS) of a hydrogen-based concept aircraft. First, the concept aircraft is introduced. Second, general requirements for the EPSS are defined, followed by defining an exemplary solution space with different proposed system architecture layouts. Last, the system architectures of the solution space are evaluated by applying the presented evaluation methods.

4.1 Reference Aircraft

To perform the systems architecture definition and evaluation process, a regional aircraft based on an *ATR72*-like aircraft platform with a seating capacity of 70 passengers and an anticipated entry into service in 2040 is proposed as reference. This reference aircraft is named *ESBEF* (German acronym for *Development of Systems and Components for Electrified Flight*) Concept Plane 1 (CP1) and is displayed in Figure 8. The *ESBEF-CP1* has ten propulsion units, each containing fuel cell systems and their peripheral systems such as thermal management and air supply. Moreover, each propulsion unit, hereafter referred to as Pod, contains a power management and distribution unit (PMAD) to control the electric power for the power train as well as for the aircraft on-board systems. Hydrogen storage is realized by two cryogenic tanks positioned in the aft fuselage. To maintain the same seating capacity as the conventional *ATR72*, the cabin configuration of the *ESBEF-CP1* has been adapted to a five-abreast seating configuration to shorten the cabin and allowing the integration of the cryogenic tanks. The systems architecture of the *ESBEF-CP1* follows a More Electric Aircraft (MEA) approach. Hence, all aircraft on-board systems are electrically powered, including an electric environmental control system, an electric de-icing system, and three separate electro-hydraulic systems [48].

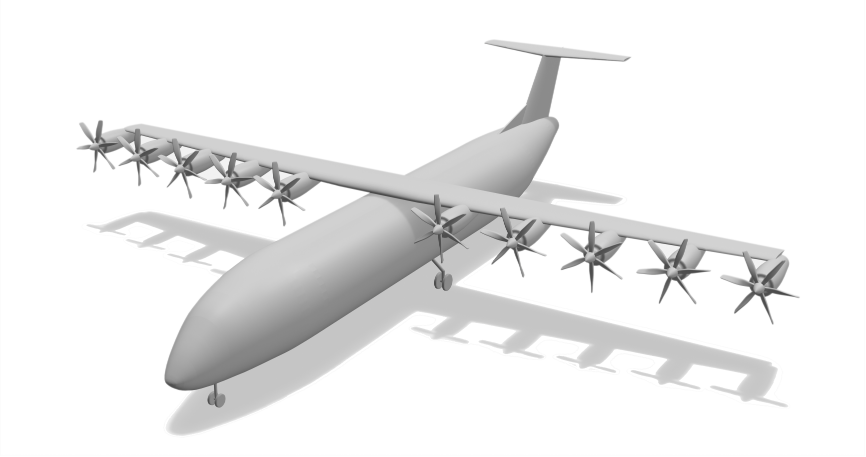


Figure 8 – Hydrogen-powered concept aircraft – *ESBEF-CP1*.

4.2 Functional Requirements for the Electric Power Supply System

In the scope of the OSD process, considered elements of the EPSS are the power sources, for example generators, power distribution units, and cables. Power sources and distribution units are sized based on maximum power requirements [38]. Electrical cables are sized according to a maximum allowed voltage drop, which is caused by losses due to the internal resistance of the cables [47]. The EPSS network is modeled by employing a graph-based approach [20]. Starting at the power sinks, i.e., electrical consumer systems, the known system parameters, such as voltage and electric power, are propagated through the electric network. All relevant components, such as power distribution units containing voltage transformers and power protection units, are sized as the algorithm continues through the network graph. Finally, the power sources are sized. This process is performed for different operation scenarios (e.g. normal operations and one engine inoperative) to be able to consider all potential worst-case scenarios in which individual components might be sized due to maximum power requirements [8, 38].

Aircraft systems with high electric power demands or high overall systems mass shares are considered as consumer systems with respect to the OSD modeling approach. Such consumer systems are the environmental control system, cabin systems, flight control system, fuel system including the hydrogen supply system, hydraulic power supply system, ice protection system, lights, and avionics systems [8, 38].

Typically, EPSS architectures are either arranged in a centralized manner or in a distributed manner [43]. Both of these layouts are visualized in Figure 9. For the central layout, the generated electric power is collected in an electric power distribution center (EPDC). Such an EPDC contains voltage transformers (e.g. transformer rectifier unit and inverter) and power protection units (e.g. circuit breaker and solid-state power controller) (cf. Figure 10). All consumer systems are directly connected to the EPDC according to the rated voltage specification. For the *ESBEF-CP1*, it is assumed that the EPDC is located in the central electrical bay near the wing box above the cabin.

As shown in Figure 9, the distributed layout includes more power distribution units compared to the central layout. Distributed layouts have been developed as part of modern aircraft platforms [43]. Generated power is collected in a primary electric power distribution center (PEPDC). It is assumed that the PEPDCs are located in the central electrical bay as well. The PEPDCs contain circuit breakers for power protection [43]. All consumer systems, which require high electrical loads, are directly connected to the PEPDCs [49]. Consumer systems, which require low electrical loads, are connected to the secondary power distribution centers (SEPDCs). Cabin and cargo consumer systems which require low electrical loads and which are considered as non-essential are supplied by the secondary power distribution boxes (SPDBs) [8, 49]. The SEPDCs and SPDBs are connected to the PEPDC [8, 49]. Both the SEPDCs and SPDBs contain voltage transformers and solid-state power controllers (SSPCs) for power protection (cf. Figure 11) [43, 49]. One SEPDC is located in the central electrical bay. The second SEPDC is located in the avionics deck below the cockpit. It is assumed that in total four SPDBs are distributed throughout the cabin.

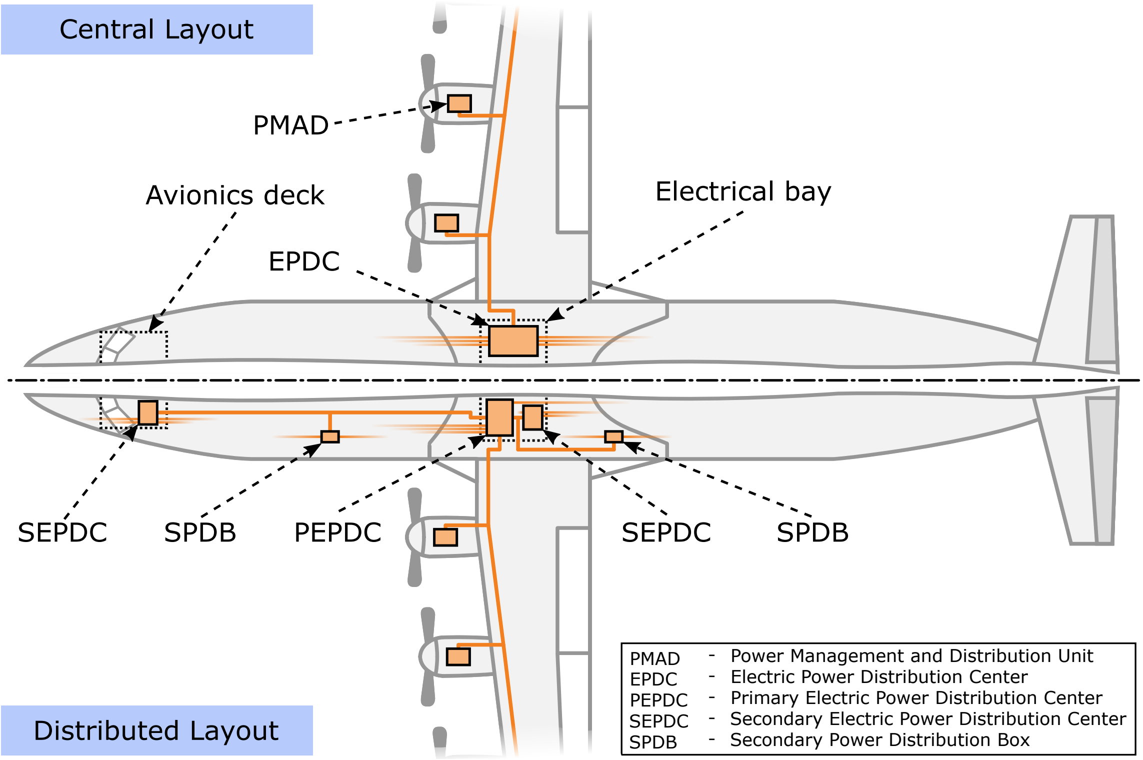


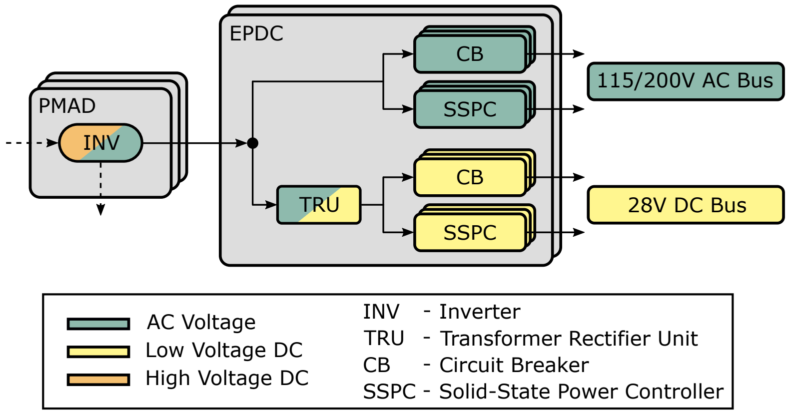
Figure 9 – Considered EPSS architecture layouts for the *ESBEF-CP1*.

4.3 Definition of a Solution Space

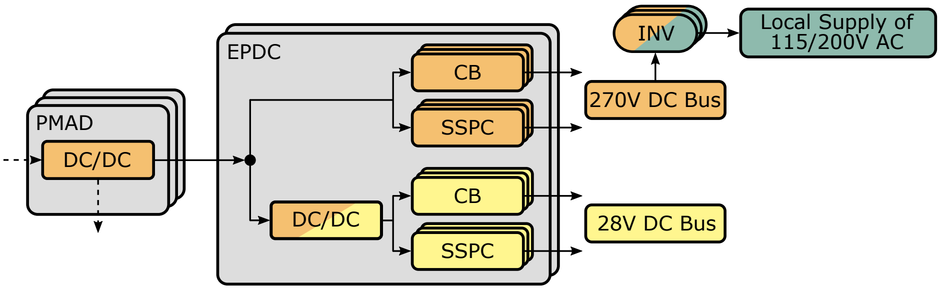
A solution space for different system architecture variants for the EPSS is created by both adapting the layout of the system architecture (i.e. central and distributed) and adapting the voltage specifications. Although the central layout has been replaced by the distributed layout in new aircraft developments, the central layout is considered for the *ESBEF-CP1* due to the following reasons:

- Central location of the electrical bay near the wing box leads to shorter cable lengths compared to larger aircraft typically having the EPDC installed in an electrical bay in the front of the fuselage [8, 11]
- Cabin power supply by distributed SPDBs might not be advantageous due to the short cabin length with 14 seat rows and due to the already central position of the electrical bay above the cabin

For each of the considered system layouts, the voltage specification of the main bus bars is either set to 115/200 V three-phase alternating current (AC) or 270 V high voltage direct current (HVDC). In both cases, 28 V direct current (DC) bus bars are installed to supply, for example, electronic and communication systems. Hence, four system architecture definitions are proposed in total. The central layout with the two voltage specifications are shown in Figure 10. It is assumed that all voltage transformer and power protection units are located in the EPDC. Figure 11 illustrates the distributed layout accordingly. It is assumed that 28 V DC is only generated in the SEPDCs and SPDBs. Both Figure 10 and Figure 11 show the required components of the system architecture for normal operations.



(a) Central architecture - Three-phase alternating current as main voltage specification.

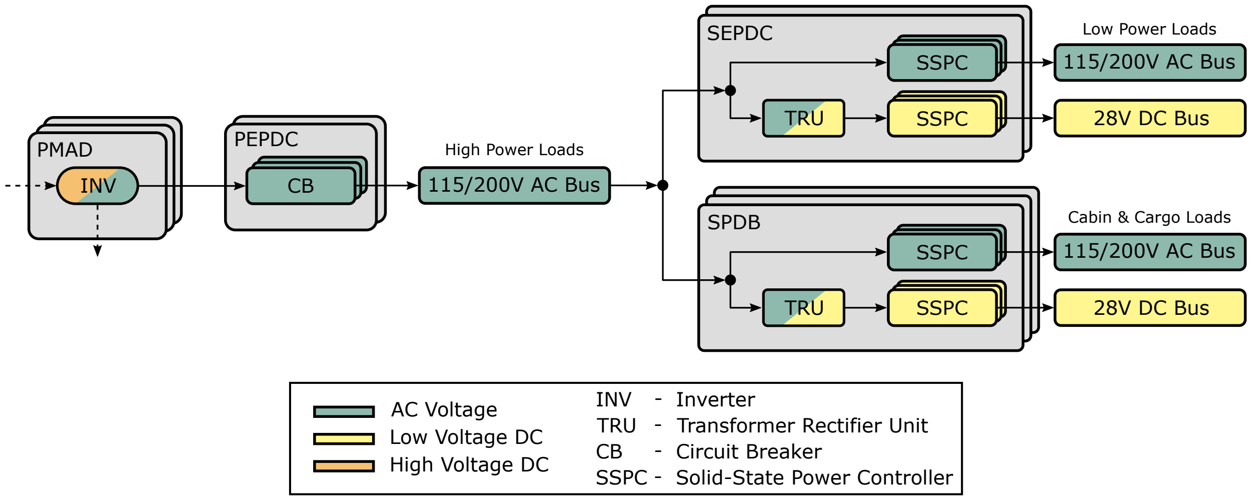


(b) Central architecture - HVDC as main voltage specification.

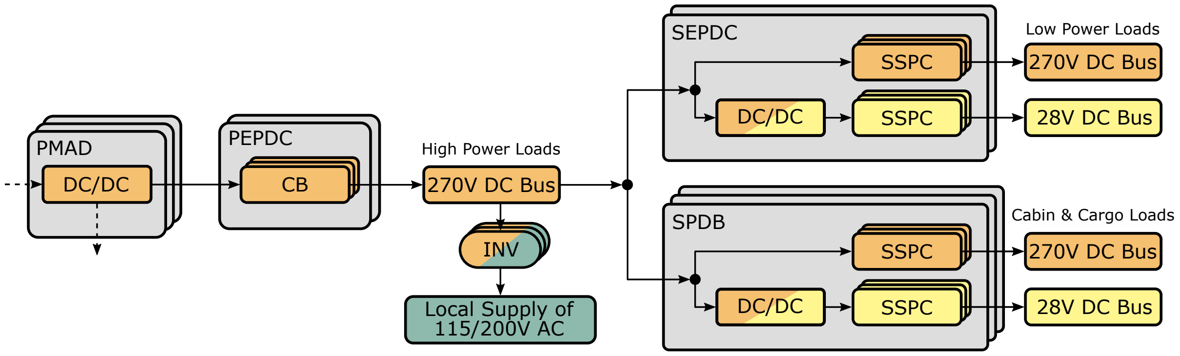
Figure 10 – Logical architecture variants of a central EPSS architecture layout (normal operations).

The reason for the consideration of a 115/200 V three-phase AC network for a hydrogen-based aircraft is to keep this particular architecture variant comparable to the ones of conventional aircraft which typically have three-phase AC networks on board. Inverters in the PMADs generate three-phase AC with constant frequency from direct current generated by the hybrid fuel cell systems. 270 V HVDC is considered for the main bus bars as then thinner cables are required for the same power compared to 115/200 V three-phase AC [11]. However, as shown in Figure 10b and Figure 11b, the HVDC supply requires an inverter for some consumer systems, such as electric motor pumps, to be able to locally provide three-phase AC.

Additional variants of EPSS architectures can be considered. This includes three-phase AC with a variable frequency or a voltage level of 230/400 V as it has been introduced in several system architectures of MEA [11]. However, a voltage with variable frequency cannot be generated by inverters and an increase of the voltage level to 230/400 V does not lead to a significant change of the considered architecture layouts as shown in Figure 10a and Figure 11a. Hence, a system architecture with a voltage level of 230/400 V is not further considered in the scope of this case study. Nevertheless, assuming constant power, cables of a 230/400 V network can be designed thinner than cables of a 115/200 V network due to the higher voltage level and the resulting smaller currents [11].



(a) Distributed architecture - Three-phase alternating current as main voltage specification.



(b) Distributed architecture - HVDC as main voltage specification.

Figure 11 – Logical architecture variants of a distributed EPSS architecture layout (normal operations).

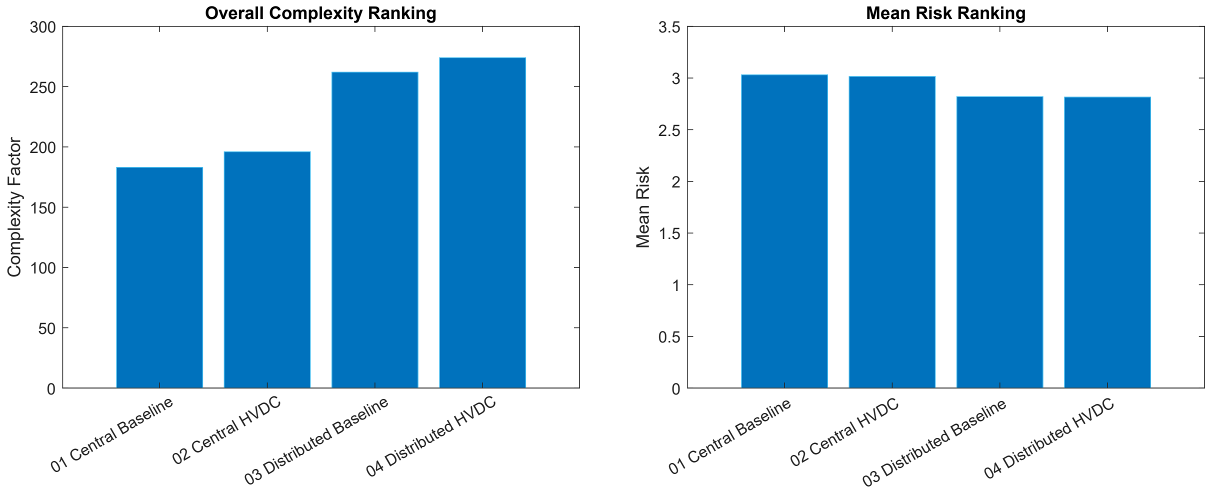
4.4 Evaluation of the EPSS Architecture Variants

All four EPSS architecture variants given in the previous chapter were modeled using a graphical MBSE tool. Although most MBSE tools offer a user-friendly interface, models of complex systems architectures often present challenges as they become difficult to understand [54]. To this end, integrated functions that support the user with evaluation of the created models are essential for examining critical systems architecture configurations. Using evaluation scripts based on the metrics given in Subsection 3.4, a ranking of the four proposed architecture variants was performed according to complexity and risk of redesign. The values for complexity factor and mean risk are absolute values and have not been scaled.

Figure 12a indicates that using HVDC for the main bus bars only causes a small increase of complexity compared to the central baseline architecture, while both distributed architectures generally show higher complexity than the central architecture variants. However, since the distributed concepts promise a reduction of system mass due to shorter cable lengths [8], excluding them from further investigation only based on complexity as a proxy metric for cost, could lead to a misjudgment. Therefore, the need for an additional proxy metric arises, indicating the architectural potential with respect to novel technology concepts and supporting experience-based evaluations. The derivation of such a metric can be challenging, since it is probably highly dependable on the system under investigation (i.e. its requirements and boundary conditions). Thus, it is part of further research.

All four proposed architectures are at TRL 9, as only technologies from existing aircraft systems architectures are incorporated. Hence, the risk factor is strongly dependent on the amount of interfaces. As Figure 12b shows, this leads to an advantage for distributed architectures, because the interfaces are more evenly distributed on the integrated components. Examining mean risk seems promising to get a rough overview of which architecture configurations might cause less major design iterations. Nevertheless, visually highlighting elements with high risk factors in an automated way to reveal locally concentrated risk areas within the systems architecture may be much more important for decision support. This includes highlighting interfacing elements which are directly affected by an architectural redesign. However, the latter is not relevant for the present study as with modeling only the EPSS and neglecting the consumer systems, especially the cross-system interfaces are out of scope. This aspect, as well as only having components at TRL 9, results in no increased risk areas.

Moreover, due to very strict requirements for the EPSS, only few architecture variants of a design space are feasible and valid [45]. Therefore, all architecture variants are close to each other in the ranking. Further research should investigate the applicability of the introduced metrics on larger problems, i.e., systems architectures with several subsystems, such as the EPSS connected to relevant consumer systems. Hereby, weaknesses due to complexity and risk of redesign of the proposed architecture variants may become more evident.



(a) Ranking of logical architecture after evaluation of overall architecture complexity.

(b) Ranking of logical architectures after evaluation of mean architecture risk.

Figure 12 – Proxy metric evaluation ranking for logical architecture variants of the EPSS.

5. Conclusion and Future Work

Novel aircraft concepts based on disruptive technologies like hydrogen-powered fuel cell systems in standalone engines aim to reduce aviation emissions towards climate neutrality. Due to these novel concepts and technologies, systems architectures complying with given top-level aircraft requirements (TLARs) still need to be defined. Furthermore, technology concept studies need to be performed to determine systems architecture performance, for instance, mass, power demand, and fuel consumption on aircraft level. Therefore, a seamless process chain from TLARs on aircraft level to technology concept studies on overall systems design (OSD)-level is necessary. Within such a seamless process chain several systems architecture variants are investigated based on a design space exploration. However, only valid and promising variants are considered at OSD-level.

Different definitions for key terms related to systems architecting, such as system, architecture, and architecting, are discussed based on several references and concluded to a definition which is used in the scope of this paper. To differentiate architecture abstraction levels, three definitions are elaborated:

- Functional architectures describe system functions, which are needed to accomplish an objective, and functional interdependencies.
- Logical architectures specify logical system components and their interactions conducting only a technology pre-selection.
- Physical architectures are characterized by high fidelity models of physical components and their connections providing the systems functions.

The primarily knowledge-based *SArA* methodology includes a process for assisting during systems architecting, i.e., in creating and evaluating systems architecture variants. The process starts by clearly analyzing and defining a given problem, focusing mainly on a TLAR-based approach, but also considering systems and functions-based approaches. The second step of *SArA* consists of the knowledge-based assisted exploration of the design space and the generation of architecture variants. The created architecture pool is validated based on safety and reliability methods. The evaluation is based on two proxy metrics: complexity for costs and TRLs for risk of high redesign effort. Lastly, the generic parametric interface file to the OSD framework *GeneSys* is presented, enabling a seamless tool chain.

SArA is applied to evaluate proposed system architecture variants of the electric power supply system for a regional hydrogen-based concept aircraft. The presented architectures consider central and distributed layouts differing in voltage specification. The evaluation of complexity of all four architecture variants indicate that using additional high voltage direct current main bus bars leads to an increase in complexity. However, distributed architecture variants are strongly penalized in the ranking, although these architectures have the potential to significantly decrease cable length resulting in a reduced system mass. Therefore, it is concluded that a third evaluation metric measuring the potential of an architecture is needed, especially with respect to disruptive technology concepts. Considering mean risk of high redesign effort all variants ranked similarly due to the fact that comparable architectures have already been certified for flight on-board aircraft.

This paper serves as a basis for further investigations in the area of knowledge-based systems architecting and for creating a seamless process chain in context of aviation as part of the aircraft conceptual design phase. An aspect, which has not yet been included in *SArA*, is a suitable method for assisting experts in conserving their knowledge. Thereby, existing knowledge does not need to be redeveloped. Furthermore, the validation of architecture variants is currently performed manually within *SArA*. Further research is necessary to include a feasible, automated safety and reliability validation method in *SArA*.

6. Contact and Acknowledgement

To contact the authors, please send an email to nils.kuelper@tuhh.de, jasmin.broehan@tuhh.de, t.bielsky@tuhh.de, and frank.thielecke@tuhh.de. The results of the presented paper are part of the work in the research project *ESBEF* which is the German acronym for *Development of Systems and Components for Electrified Flight* and is supported by the Federal Ministry for Economic Affairs and Climate Action in the national LuFo VI program. Any opinions, findings, and conclusions expressed in this document are those of the authors and do not necessarily reflect the views of the other project partners.

7. Copyright Statement

References

[1] Air Transport Action Group. Waypoint 2050: Balancing growth in connectivity with a comprehensive global air transport response to the climate emergency: vision of net-zero aviation by mid-century. 2021.

[2] Airbus. ZEROe: Towards the world's first zero-emission commercial aircraft. 2022, URL: https://www.airbus.com/en/innovation/zero-emission/hydrogen/zeroe.

[3] Alder M, Moerland E, et al. Recent advances in establishing a common language for aircraft design with CPACS. *Aerospace Europe Conference*, Bordeaux, France, 2020.

[4] Altfeld H-H. *Commercial aircraft projects: managing the development of highly complex products*. Ashgate, 2011, ISBN: 9780754677536.

[5] Annighöfer B. Model-based architecting and optimization of distributed integrated modular avionics. Dissertation, Hamburg, Germany, Hamburg University of Technology, 2015.

[6] Armstrong M J. A process for function based architecture definition and modeling. Master Thesis, Atlanta, USA, Georgia Institute of Technology, 2008.

[7] Armstrong M J. Identification of emergent off-nominal operational requirements during conceptual architecting of the more electric aircraft. Dissertation, Atlanta, USA, Georgia Institute of Technology, 2011.

[8] Bielsky T, Juenemann M and Thielecke F. Parametric modeling of the aircraft electrical supply system for overall conceptual systems design. *German Aerospace Congress*, Bremen, Germany, 2021.

[9] Bornholdt R. Systemübergreifende Analyse und Bewertung von Architekturvarianten neuartiger Flugzeugsysteme anhand von Sicherheits- und Betriebsaspekten. Dissertation, Hamburg, Germany, Hamburg University of Technology, 2021. DOI: 10.2370/9783844078831.

[10] Bothe H-H. *Fuzzy Logic: Einführung in Theorie und Anwendungen*. Vol. 2, Springer, 1995, ISBN: 9783662073575.

[11] Brombach J, Lücken A, et al. Comparison of different electrical HVDC-architectures for aircraft application. *2012 Electrical Systems for Aircraft, Railway and Ship Propulsion*, Bologna, Italy, 2012, DOI: 10.1109/ESARS.2012.6387380.

[12] Bussemaker J H and Ciampa P D. MBSE in architecture design space exploration. In: *Handbook of Model-Based Systems Engineering*. Ed. by Madni A M, Augustine N and Sievers M. Cham, Germany: Springer International Publishing, 2020, pp 1–41, ISBN: 978-3-030-27486-3. DOI: 10.1007/978-3-030-27486-3_36-1.

[13] Bussemaker J H, Ciampa P D and Nagel B. System architecture design space exploration: an approach to modeling and optimization. *AIAA AVIATION 2020 Forum*, 2020, DOI: 10.2514/6.2020-3172.

[14] Bussemaker J H, Ciampa P D and Nagel B. System architecture design space modeling and optimization elements. *32nd Congress of the International Council of the Aeronautical Sciences*, Shanghai, China, 2021.

[15] Chakraborty I. Subsystem architecture sizing and analysis for aircraft conceptual design. Dissertation, Atlanta, USA, Georgia Institute of Technology, 2015.

[16] Clarkson P J, Simons C and Eckert C. Predicting change propagation in complex design. *Journal of Mechanical Design*, Vol. 126, No. 5, 2004, pp 788–797, ISSN: 1050-0472. DOI: 10.1115/1.1765117.

[17] Crawley E, Weck O de, et al. MIT engineering systems monograph. Ed. by The ESD Architecture Committee. 2004.

[18] Dahmann J S, Fujimoto R M and Weatherly R M. The department of defense high level architecture. *Proceedings of the 29th conference on Winter simulation*, Atlanta, USA, 1997, pp 142–149, DOI: 10.1145/268437.268465.

[19] de Tenorio C. Methods for collaborative conceptual design of aircraft power architectures. Dissertation, Atlanta, USA, Georgia Institute of Technology, 2010.

[20] Dunker C. Rechnergestützter Entwurf von elektro-hydraulischen Energieversorgungssystemen für Verkehrsflugzeuge. Dissertation, Hamburg, Germany, Hamburg University of Technology, 2017.

[21] Eclipse Foundation. Let yourself be guided with ARCADIA: A comprehensive methodological and tool-supported model-based engineering guidance. 2022, URL: `https://www.eclipse.org/capella/arcadia.html`.

[22] Eisenhut D, Moebs N, et al. Aircraft requirements for sustainable regional aviation. *Aerospace*, Vol. 8, No. 3, 2021, pp 61, DOI: `10.3390/aerospace8030061`.

[23] European Commission. Flightpath 2050: Europe's vision for aviation: report of the high-level group. Luxembourg, 2011.

[24] Finn J, Nuzzo P and Sangiovanni-Vincentelli A. A mixed discrete-continuous optimization scheme for cyber-physical system architecture exploration. *IEEE/ACM International Conference on Computer-Aided Design*, Austin, USA, 2015, DOI: `10.1109/ICCAD.2015.7372573`.

[25] Fioriti M, Boggero L, et al. Automated selection of the optimal on-board systems architecture within MDO collaborative environment. *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Denver, USA, 2017, DOI: `10.2514/6.2017-3150`.

[26] Garg T, Eppinger S, et al. Using TRLs and system architecture to estimate technology integration risk. *Proceedings of the 21st International Conference on Engineering Design*, Vancouver, Canada, 2017.

[27] Garriga A G, Govindaraju P, et al. A modelling framework to support power architecture trade-off studies for more-electric aircraft. *Transportation Research Procedia*, Vol. 29, 2018, pp 146–156, ISSN: 2352-1465. DOI: `10.1016/j.trpro.2018.02.013`.

[28] Hoelzen J, Silberhorn D, et al. Hydrogen-powered aviation and its reliance on green hydrogen infrastructure: Review and research gaps. *International Journal of Hydrogen Energy*, Vol. 47, No. 5, 2022, pp 3108–3130, ISSN: 03603199. DOI: `10.1016/j.ijhydene.2021.10.239`.

[29] International Council of Systems Engineering. *INCOSE systems engineering handbook: a guide for system life cycle processes and activities*. Vol. 4, Wiley, 2015, ISBN: 9781118999400.

[30] Jackson D W. Robust aircraft subsystem conceptual architecting. Dissertation, Atlanta, USA, Georgia Institute of Technology, 2013.

[31] Johansson C, Derelöv M and Ölvander J. How to use an optimization-based method capable of balancing safety, reliability, and weight in an aircraft design process. *Nuclear Engineering and Technology*, Vol. 49, No. 2, 2017, pp 404–410, ISSN: 17385733. DOI: `10.1016/j.net.2017.01.006`.

[32] Judt D M and Lawson C. Methodology for automated aircraft systems architecture enumeration and analysis. *12th AIAA Aviation Technology, Integration, and Operations Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Indianapolis, USA, 2012.

[33] Judt D M and Lawson C. Development of an automated aircraft subsystem architecture generation and analysis tool. *Engineering Computations*, Vol. 33, No. 5, 2016, pp 1327–1352, ISSN: 0264-4401.

[34] Juenemann M, Thielecke F, et al. Methodology for design and evaluation of more electric aircraft systems architectures within the AVACON project. *German Aerospace Congress*, Darmstadt, Germany, 2019, DOI: `10.25967/480197`.

[35] Kececioglu D. *Reliability engineering handbook*. Vol. 2, Prentice-Hall, 1991, ISBN: 0137723024.

[36] Kinnunen M J. Complexity measures for system architecture models. Dissertation, Cambridge, USA, Massachusetts Institute of Technology, 2006.

[37] Kleiner S and Kramer C. Model based design with systems engineering based on RFLP using V6. In: *Smart product engineering*. Ed. by Abramovici M and Stark R. Springer, 2013, pp 93–102, ISBN: 978-3-642-30816-1.

[38] Koeppen C. Methodik zur modellbasierten Prognose von Flugzeugsystemparametern im Vorentwurf von Verkehrsflugzeugen. Dissertation, Hamburg, Germany, Hamburg University of Technology, 2006.

[39] Liscouet-Hanke S. A model-based methodology for integrated preliminary sizing and analysis of aircraft power system architecture. Dissertation, Toulouse, France, Institut National des Sciences Appliquées de Toulouse, 2008.

[40] Maier M W. Architecting principles for systems-of-systems. *Systems Engineering*, Vol. 1, No. 4, 1998, pp 267–284, DOI: `10.1002/(SICI)1520-6858(1998)1:4<267::AID-SYS3>3.0.CO;2-D`.

[41] Maier M W and Rechtin E. *The art of systems architecting*. Vol. 3, CRC Press, 2009, ISBN: 1420079131.

[42] Mankins J C. Technology readiness levels. 1995, URL: `https://aiaa.kavi.com/apps/group_public/download.php/2212/trls_mankinspaper_1995.pdf`.

[43] Nya B, Brombach J, et al. Weight evaluation of cabin power architecture on smaller civil aircraft. *AST 2011*, 2011.

[44] Pfennig M. Methodik zum wissensbasierten Entwurf der Antriebssysteme von Hochauftriebssystemen. Dissertation, Hamburg, Hamburg University of Technology, 2012.

[45] Raksch C. Eine Methode zur optimalen Redundanzallokation im Vorentwurf fehlertoleranter Flugzeugsysteme. Dissertation, Hamburg, Hamburg University of Technology, 2013.

[46] Richards M G, Hastings D E, et al. Systems architecting for survivability: limitations of existing methods for aerospace systems. *Conference on Systems Engineering Research*, Hoboken, USA, 2008.

[47] SAE Society of Automotive Engineers. Aircraft electrical systems: ARD50055. 1998.

[48] Schefer H, Fauth L, et al. Discussion on electric power supply systems for all electric aircraft. *IEEE Access*, Vol. 8, 2020, pp 84188–84216, DOI: `10.1109/ACCESS.2020.2991804`.

[49] Terorde M, Wattar H and Schulz D. Phase balancing for aircraft electrical distribution systems. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 51, 2015, pp 1781–1792, ISSN: 0018-9251.
[50] The Engineering Society for Advancing Mobility Land Sea Air and Space. ARP 4761: Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment. 1996.
[51] The MathWorks. System Composer: Getting Started Guide. URL: `https://www.mathworks.com/help/pdf_doc/systemcomposer/index.html`.
[52] Ulrich K T and Eppinger S D. *Product design and development*. Vol. 5, McGraw-Hill, 2012, ISBN: 0073404772.
[53] Voirin J-L. Method and tools to secure and support collaborative architecting of constrained systems. *27th Congress of the International Council of the Aeronautical Sciences*, Nice, France, 2010.
[54] Watkins C B, Varghese J, et al. System architecture modeling for electronic systems using MathWorks System Composer and Simulink. *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, IEEE, 10/11/2020 - 10/15/2020, pp 1–10, ISBN: 978-1-7281-9825-5. DOI: `10.1109/DASC50938.2020.9256753`.