INTERNATIONAL DESIGN CONFERENCE – DESIGN 2022
https://doi.org/10.1017/pds.2022.187

Harnessing the Complexity for Vehicle System Design at the Concept Design Phase of an Aircraft

A. D. Drego ✉
Saab AB, Sweden
✉ adelia.drego@saabgroup.com

Abstract
Aircraft vehicle systems enable an aircraft to fly safely throughout a mission. Generating feasible vehicle system architectures at the aircraft concept design phase is complex. Aspects from various complex systems theories are used to provide different insights into this complexity. To address this complexity, a framework based on industrial reality that can used recursively is presented. The framework employs various design theories to harness the complexity of vehicle system design at the concept design phase of an aircraft.

Keywords: complex systems, conceptual design, complexity

1. Introduction

Aircraft vehicle systems are the systems that enable the aircraft to fly safely throughout a mission. Typically, they are the systems that transfer energy to other systems, equipment, and subassemblies in the aircraft (Steinkellner, 2011). Generally, vehicle systems that are common to most types of engine-powered aircraft include the engine, environmental control system (ECS), fuel system, electrical power system, actuation system, and landing and braking systems.

In this paper, the life cycle of an aircraft is defined by the generic life cycle model (ISO/IEC/IEEE 15288:2015) described in INCOSE (2015). Using this model, vehicle system design would typically commence at the aircraft development stage rather than at the aircraft concept stage. This paper proposes that vehicle system design be introduced at the aircraft concept stage because the function, performance, weight, and volume of vehicle systems impact those of the aircraft.

Vehicle system designers would be tasked with generating vehicle system concepts at the aircraft concept stage. Vehicle system concepts result in vehicle system architectures. The definition of a concept by Crawley *et al*. (2015) is adopted in this paper. Crawley *et al*. (2015) defined a concept as a notional mapping between function (what the system or product does) and instruments of form (what the system or product is). In this paper, an architecture is defined as a pictorial representation of the relationships between the forms and operands (the thing that the function changes). To avoid generating unfeasible architectures, the vehicle system designer must understand how the system fits into the aircraft it is being designed for.

Aircraft are complex systems. Depending on the aircraft type and the missions the aircraft is being designed to perform there are various factors that significantly affect vehicle system design. However, these factors may not be clearly defined at the concept phase of an aircraft. This makes the process for generating feasible vehicle system architectures while trying to account for these factors more cumbersome. Firstly, various complex systems theories are considered in this paper to understand these factors. Aspects from complex products and systems (CoPS) theory by Davies and Hobday (2005) are used to understand the role of an aircraft vehicle system designer at an aircraft original equipment manufacturer (OEM) that designs and builds aircraft. Aspects of normal accident theory by Perrow (1984) and system architecture theory by Crawley *et al*. (2015) are used to identify the factors that significantly influence the design choices for a vehicle system architecture. Secondly, aspects from axiomatic design theory, feature modelling method, design structure matrix (DSM) method, and the systems thinking approach are used to create a framework in this paper. The framework presented enables feasible vehicle system architecture generation by accounting for the factors that significantly affect vehicle system design but may not be clearly defined at the aircraft concept phase.

2. The role of a vehicle system designer at the concept design phase of an aircraft

To comprehend how to generate and analyse a vehicle system architecture for an aircraft, firstly the role of a vehicle system designer at an OEM that produces complex systems and products such aircraft must be understood. OEMs that produce complex systems and products are often system integrators. While system integration began as a technical, operations task in the US military in the late 1940s, nowadays it is a strategic business capability that plays a central role in the management of several high-technology projects. The significance of system integration increases as the complexity, technology, and cost of the product increases (Davies and Hobday, 2005).

Systems integration capability and performance at the strategic level can be rewarding to all prime contractors and final integrators of high-technology products, systems, networks and constructs as well as the suppliers of major components, control units and subsystems (Davies and Hobday, 2005). In CoPS theory, Davies and Hobday (2005) noted that system integration at the technological level is the underlying capability that enables new product development and market capability. Prencipe (1997) demonstrated using aircraft engines as a study case that the development of a product within a given technological family requires static (intra-generation) technological capabilities. However, even more significant to long-term competitive advantage is the dynamic (inter-generation) capabilities that are needed to envision and produce new product architectures and new product families (Davies and Hobday, 2005).

System integration capability is evident at an aircraft OEM. The inter-generational capability of Saab has resulted in the company producing five families of fighter jet aircraft over the last eighty years. The Gripen E/F are the latest versions in the latest fighter jet family at Saab, the Gripen family of fighter aircraft. The development of the first versions of the Gripen, namely A/B, commenced in the early 1980s. Due to the intra-generational and inter-generational capabilities at Saab, the general airframe configuration of the Gripen E/F is like its predecessors within the Gripen family, but the engine, most of the other systems, and the airframe structure have been updated.

When developing an aircraft, there is a transformation in the systems integration capability of the aircraft OEM. The OEM removes itself from detailed involvement in component design and manufacturing and focuses more on system integration of various modules produced by others in the supply chain (Davies and Hobday, 2005). The role of an aircraft vehicle system designer aligns with this integration capability. Aircraft OEMs typically do not design in detail and manufacture the components and equipment contained in aircraft vehicle systems; they instead outsource these tasks to subsystem suppliers. One of the specific functions of a vehicle system designer at the concept design phase entails formulating a set of overall specifications that represent the performance of the vehicle system and its interactions with other systems. For some vehicle systems, the OEM might purchase the system commercial of the shelf (COTS) from a supplier based on specifications. In other cases, the OEM might design the architecture of a vehicle system and contract a subsystem supplier to manufacture it. The greatest challenge an aircraft vehicle system designer faces at the concept phase of an aircraft is ensuring that the vehicle system architecture has been appropriately designed to be integrated into the aircraft.

3. An aircraft vehicle system: a system within a complex system

The distinction between complex and complicated systems is considered in this paper. In this paper, the definitions by Crawley *et al*. (2015) are considered. Apparent complexity is akin to the level of complicatedness of a system. The term ‘complicated’ can be used as a measure of human ability to

https://doi.org/10.1017/pds.2022.187 Published online by Cambridge University Press

perceive and comprehend complexity. If a system is complicated, then it has high apparent complexity and complicated systems are hard for humans to comprehend (Crawley *et al*., 2015). Using this notion, an aircraft vehicle system designer would find it far less complicated to analyse the architecture of a vehicle system as a standalone system than to analyse it when it is integrated in the aircraft. However, the former is rarely ever the case.

To understand integration and 'interconnectedness' within a complex system such as an aircraft, the concepts, complex interactions and tight couplings, coined by Perrow (1984) are applicable. These two concepts are discussed below in the context of aircraft vehicle systems.

Complex interactions and tight couplings are inherent to aircraft, and they play a significant role in the design choices for aircraft vehicle systems. However, the interactions and couplings affecting design choices may not be defined at the concept stage of an aircraft. If they are not defined, then it makes it difficult for the vehicle system designer to generate feasible architectures for the vehicle system of interest.

Regarding the concept of complex interactions, aircraft consist of many interconnected subsystems that interact with each other to perform various functions. This leads to many connections between components within a subsystem as well as connections between different subsystems. At the aircraft concept stage, the connections between a vehicle system and other subsystems may be unclear. The undefined connections between a vehicle system and other subsystems pose one of the greatest challenges to a vehicle system designer at the concept design phase of an aircraft. Therefore, at the aircraft concept stage, what a vehicle system is dependent on and what depends on the system may not be defined. If they are not defined, then the possible interactions between subsystems cannot be comprehended.

At the aircraft concept stage, the internal configuration of the aircraft may not be clearly defined either. Therefore, the equipment and systems in the vicinity of a vehicle system in the aircraft may not be clearly known. The designer is unable to predict the effects the surrounding equipment and systems might have on the vehicle system. Spatial hazard analyses are used to assess the effect(s) a system or equipment has on other systems or equipment when they are physically configured near each other in the aircraft. If the internal configuration is changing or not clearly defined, the vehicle system designer requires a method to note the changes.

Regarding the concept of tight couplings, there are many tight couplings between a vehicle system and the system(s) it depends on or between the vehicle system and the subsystems that depend on it. If a vehicle system such as the ECS depends on the engine for bleed air supply, then what are the redundancies in place if the bleed air supply system fails? If a subsystem depends on the electrical power system what are the redundancies in place if the power system fails? System hazard analyses are conducted to assess the risk of a system or equipment failing in the aircraft. With a vague picture of the interconnectedness of the vehicle system with other subsystems, the vehicle system designer has to design the appropriate redundancies or buffers for safety in case of a failure. However, the systems dependent on a vehicle system or those that the vehicle system depends on might change (in some projects multiple times) during the concept phase of an aircraft. Therefore, the vehicle system designer requires a method to keep track of the subsystems that the vehicle system depends on and those that it provides to.

4. Innovation in integration: systems engineering for complex systems

Assembly is physically putting entities together while integration is the process of understanding what entities should be put together and how entities should be put together. At an aircraft OEM, integration is the process of coordinating various subsystems, subassemblies, and processes to make them work together in the final product. Integration of various entities into a complex system must be conducted in an organised manner to account for all changes that the complex system might be sensitive to. This is where a systems engineering approach is useful.

At the onset of the Cold War, the complexity and ambition of military systems were increasing in the US military. This weakened the traditional single-discipline, linear approach to system development that entailed making one part of the system, then the next, and so on. In turn, this gave rise to the systems

https://doi.org/10.1017/pds.2022.187 Published online by Cambridge University Press

engineering-integration approach that provided a systematic, multi-disciplinary approach to systems development which was concurrent and not sequential (Davies and Hobday, 2005).

One approach to developing complex systems is the integration-driven development approach described in Taxén and Pettersson (2010). Parallel incremental development of a complex system like an aircraft could employ the delta (Δ) anatomies approach shown on the left-hand side of (Figure 1). The anatomy shows all currently planned system changes (Δ) and their dependencies. The dependencies constrain the order in which changes can be done and determine the possible level of parallelism. Therefore, as shown in (Figure 1), ΔC can only be integrated and tested after ΔA and ΔB are integrated. In integration-driven development, interfaces are very important, and they should be established as early as possible for large projects, with several teams working in parallel.

The systems engineering approach lets engineers break down a system into smaller, manageable subsystems, assemblies and subassemblies while simultaneously developing interface specifications for each component before they are constructed. This approach provides engineers with flexibility in system development by reducing subsystem interactions and (unanticipated) emergent behaviour that adversely affects the design and final functioning of the system (Davies and Hobday, 2005).

The implication of several design changes is reduced with the systems engineering approach since it allows engineers to freeze the system design at the most appropriate point (Davies and Hobday, 2005). The right-hand side of (Figure 1) is an illustration of the systems engineering approach employed in the development of an aircraft at Saab. Development of each Δ by a different subsystem or subassembly team is concurrently conducted and fed into the overall system development as indicated by the black arrows. At each system version in (Figure 1), relevant feedback (indicated with the blue arrows) from the overall system is provided to a team(s) responsible for a particular subsystem or subassembly. The information exchange during this parallel incremental development process helps reduce the number of unknown interactions as well as the unanticipated, undesirable emergent behaviour as the system is integrated.

This integration-driven approach is appropriate at the detailed development phase of an aircraft when the specifications and configuration of the complete airframe is defined. However, as established in §3, the factors that drive the design choices for a vehicle system architecture may not be defined at the aircraft concept stage. But the chosen vehicle system architecture must still be feasible for the aircraft it is being designed for. With system integration being a key capability of an aircraft OEM, the focus for sustained development at an OEM would be innovation in integration even at the aircraft concept stage.

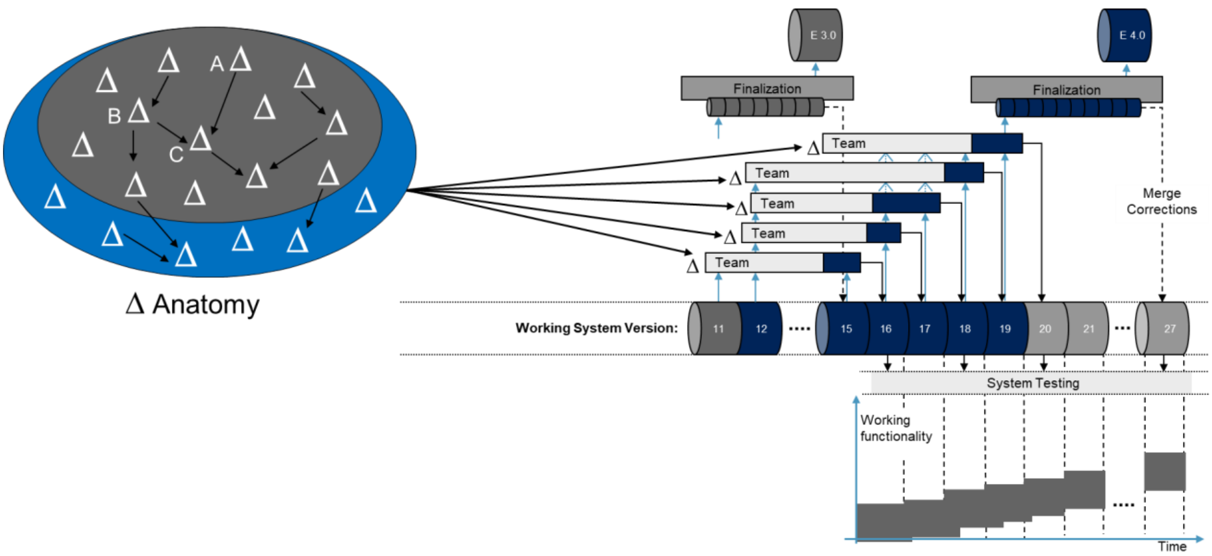

Figure 1. An illustration of the integration-driven development approach adapted from Taxén and Pettersson (2010)

https://doi.org/10.1017/pds.2022.187 Published online by Cambridge University Press

5. A framework for harnessing the complexity for vehicle system design at the concept design phase of an aircraft

A framework for harnessing the complexity for vehicle system design at the concept design phase of an aircraft is proposed in this section. The framework proposed is based on industry experiences of an aircraft OEM. The ECS of a manned fighter aircraft is used as a case study to present this framework.

The framework employs aspects from axiomatic design theory presented in Suh (2001) and Suh (2007), DSM method described in Browning (2001) and Eppinger and Browning (2012), feature modelling method described in Meinicke *et al*. (2017) and Kang *et al*. (1990), and the systems thinking approach presented in Crawley *et al*. (2015).

The framework consists of eight steps presented in §5.1 to §5.8 in the same sequence that they need to be carried out in order to generate feasible vehicle system architectures. The framework can be used recursively to account for the factors that significantly affect vehicle system design but might change during the concept phase of an aircraft.

5.1. Defining the vehicle system attributes using axiomatic design theory

The first step in the framework is to define the attributes for the vehicle system in the stakeholder domain. In this framework, the terms, customer domain and customer attributes from axiomatic design theory are replaced with stakeholder domain and system attributes, respectively. This is because vehicle system design is at a lower level of abstraction with respect to complete aircraft design. In this framework, the stakeholders of a vehicle system are considered to be a collective group including the overall aircraft designer and the engineers responsible for systems that depend on the vehicle system of interest.

The vehicle system designer typically receives vaguely-defined attributes from the stakeholder(s) of the system. It is the responsibility of the vehicle system designer to interpret those attributes in the context of the vehicle system of interest and then translate those attributes into well-defined functional requirements for the system. The system attributes for an ECS of a manned fighter aircraft are listed in the first block labelled 'Stakeholder Domain' in (Figure 4).

For the case study at hand, system attributes are divided into two categories, 'Flight Critical' and 'Mission Critical'. With this division, the physical means to fulfil flight critical requirements can be separated from those that fulfil mission critical requirements. Therefore, the ECS could eventually consist of two physically-separate systems with each system not straining the other during combat missions. In a combat mission, typically the ECS performs functions to keep the aircraft and pilot safe while also ensuring that the mission systems (dependent on it) operate at the required capacity. Hence, separating the ECS of fighter aircraft into two systems is preferred for tactical and aircraft safety reasons.

5.2. The system thinking approach for identifying the vehicle system boundary

One of the tasks of the system thinking approach entails identifying the entities of the system, their form and function, and the system boundary and context (Crawley *et al*., 2015). A vehicle system is connected to various other subsystems and equipment in the aircraft that either depend on the vehicle system or provide to it. Defining the boundary of the vehicle system makes it clear as to what is contained in the system and what belongs outside the system. The systems and equipment that provide or depend on the vehicle system of interest are to be placed outside this boundary and they make up the 'context' of the system. The system thinking approach is applied to define the boundary of the ECS as shown in (Figure 2). Interpreting the system attributes defined in (Figure 4), the systems and subassemblies that may depend on the ECS are indicated on the left-hand side of the system boundary and marked as 'Downstream'. The systems indicated on the right-hand side of the system marked 'Upstream' are what the vehicle system may depend on. In the aircraft development stage, this categorization could be implemented, for example in a SysML-based tool (Friedenthal *et al*., 2015).

https://doi.org/10.1017/pds.2022.187 Published online by Cambridge University Press

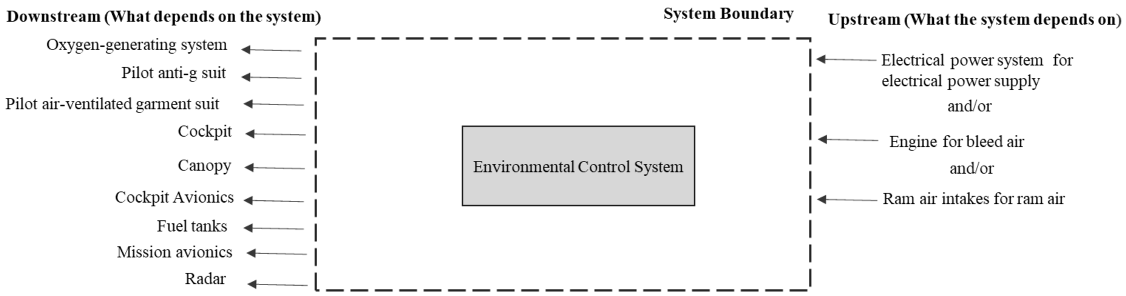

Figure 2. The system boundary and the upstream systems (what the system depends on) and downstream systems (what depends on the system) for the ECS case study in this paper

5.3. Feature modelling and the system thinking approach for identifying the main aspects of a vehicle system

According to Suh (2001) when using axiomatic design theory, once the customer (system) attributes are defined, they must be translated into functional requirements in a solution-neutral environment. A solution-neutral environment is one in which the functional requirements must be defined with no consideration for something that has already been designed or what the design solution should be (Suh, 2001). However, a solution-neutral environment might be counter intuitive for a vehicle system designer at an aircraft OEM. The designer might need to comply with the system integration strategy of the OEM that might include buying COTS components. Therefore, the designer will require a good grasp of the contemporary systems that are in development or in use. This paper suggests a pseudo solution-neutral environment by creating a schematic of all the main aspects of the vehicle system of interest by studying existing systems.

The main aspects of an aircraft vehicle system can be identified using the system thinking approach. One of the tasks of the system thinking approach is to identify the system, its form and function. In this paper, the term 'form' and the term design parameter from axiomatic design theory are considered to be synonymous. Function is made up of a process and an operand. The part of the function that is pure action or transformation is a process and it modifies the state of an operand (Crawley *et al*., 2015). Depending on the vehicle system, the operand(s) could be fuel for the fuel system or electrical current for the electrical power system. For the ECS, the operand could be ram air, bleed air, and/or polyalphaolefin (PAO). For the actuation system, it could be oil or air. Using FeatureIDE (METOP GmbH, 2020), a feature model of the main aspects that are inherent to an ECS of a manned fighter aircraft was created and it is shown in (Figure 3).

The ECS of various currently-operating fighter aircraft as well as commercial aircraft were analysed to create the feature model in (Figure 3). The feature groups, 'Refrigeration' and 'Heat Exchanging', represent the design parameters (components or subsystems) of an ECS. The classification under the heat exchanging group was adopted from Shah and Sekulic (2003). The feature groups, 'Air Supply' and 'Heat Transportation', represent the operands of an ECS. All four feature groups are set as mandatory since at least one alternative from each group is needed to fulfil the system attributes.

The detail in the ECS feature model in (Figure 3) could also be expanded to include other branches such as safety valves and sensors that are inherent safety features in an ECS. However, for the purpose of demonstrating the fundamental aspects of an ECS, the feature model in (Figure 3) is sufficient.

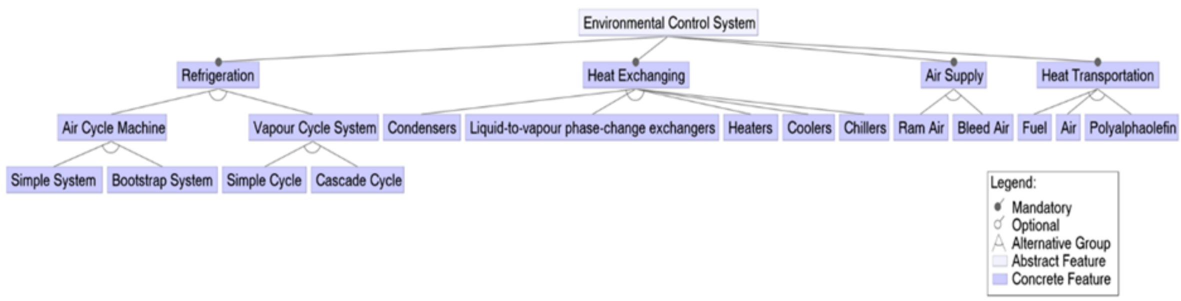

Figure 3. Feature model created using FeatureIDE (METOP GmbH, 2020) for the ECS case study in this paper

https://doi.org/10.1017/pds.2022.187 Published online by Cambridge University Press

5.4. Axiomatic design theory for mapping the vehicle system attributes to functional requirements of the system

Once a feature model of the vehicle system of interest is created, then the system attributes can be translated into functional requirements. The feature model serves as a guide for translating attributes to functional requirements. For example, the four main groups in the feature model in (Figure 3) can be used to realise what functions the ECS would be able to perform. The functional requirements of the ECS are defined in the block labelled 'Functional Domain 1' in (Figure 4) based on the system attributes defined in the stakeholder domain. Note that a single system attribute may not necessarily translate to a single functional requirement. For the case study at hand, the first attribute, SA1 that is 'pilot comfort and safety' translates into the first three functional requirements, FR1 through FR3. SA2, SA3, SA4, and SA5 translate into FR4, FR5, FR6, and FR7, respectively. SA6 and SA7 that are mission critical translate into FR8 and FR9, respectively.

Clearly-defined functional requirements of a vehicle system are possible only through communication between the vehicle system designer and the stakeholder(s) of the system. All flight critical attributes are translated into pneumatic functional requirements and all mission critical attributes are translated into non-pneumatic functional requirements.

All mission critical attributes could also be fulfilled by pneumatic means. However, this may not be ideal for the ECS of a modern-day fighter aircraft. Typically, the 'cooling power' demand of a military-grade aircraft radar is high. Along with low temperatures, the volume flow rate of air may need to be very high to dissipate across the complete surface area of the radar. Hence, it would be more practical to fulfil mission critical attributes through non-pneumatic means.

All pneumatic requirements, FR1 through FR7 can be lumped into a single functional requirement called FR1. Similarly, FR8 and FR9 can also be lumped into a single functional requirement called FR2. The new FR1 and FR2 are defined in the block labelled 'Functional Domain 2' in (Figure 4). Therefore, the two main functions of the ECS are to provide conditioned air to subsystems and subassemblies and to cool subsystems by non-pneumatic means. The main functions can be mapped to the main forms of the system.

5.5. Identifying the input constraints on the vehicle system using axiomatic design theory

Once the functional requirements for a vehicle system have been defined then all the input constraints on the system must be noted. This step must be conducted prior to mapping functional requirements to design parameters. Suh (2007) defines constraints as non-functional requirements and classifies them as either input or system constraints. All proposed designs must satisfy input constraints while system constraints are specific to a given design. With the knowledge of input constraints in hand, the vehicle system designer can make appropriate design choices and avoid generating unfeasible concepts of a vehicle system. System constraints can be identified after the architecture of a vehicle system concept is generated.

Typically, input constraints that are quantifiable for vehicle systems are managed by budgets. Usually, measurable constraints include weight, volume, cost, safety, and reliability. Safety and reliability of the system are usually measured as a probability of failure. Depending on the vehicle system and the choice of design parameters there might be budgets for other resources the system requires. For example, if the ECS requires electrical power to operate then it may be provided with an electrical power budget or vice versa. The ECS may provide the electrical power system with a budget based on its own needs.

The cooling requirement for cockpit avionics (FR7) could be fulfilled by non-pneumatic means. However, for pilot safety it should only be fulfilled through pneumatic means. This is to prevent any hazard of liquid leakage in the cockpit with a non-pneumatic cooling system. If mission avionics were located in the cockpit then they would also be cooled by pneumatic means. This is an example of an input constraint for system safety.

Input constraints could also be specified by the stakeholders of the system. For example, the life support systems for the pilot may require a minimum mass flow rate of air from the ECS within a specific temperature range to keep the pilot comfortable during all phases of a mission. The operating conditions for a fighter aircraft are harsher than those for a commercial aircraft. In a single mission, a fighter aircraft

https://doi.org/10.1017/pds.2022.187 Published online by Cambridge University Press

may fly at high speed at low altitude and at high speed at high altitude along with performing other combat manoeuvres. While a commercial aircraft may typically fly at high speed at cruising altitude for most of its mission. Therefore, the operating conditions of the aircraft also affects the design choices for vehicle systems. In summary, to ensure that the vehicle system architecture is feasible for the aircraft it is to be integrated into, several hundred input constraints may have to be fulfilled. Ensuring architecture feasibility may require risk-informed decision making which can be done in accordance with Dezfuli *et al*. (2010).

5.6. Axiomatic design theory for mapping vehicle system functional requirements to design parameters

Once the input constraints on the vehicle system are listed, then the functional requirements can be mapped to the design parameters of the system. For the case study at hand, FR1 and FR2 defined in Functional Domain 2 in (Figure 4) are mapped to DP1 and DP2, respectively. DP1 and DP2 are defined in the block labelled 'Physical Domain' in (Figure 4). Using the ECS feature model in (Figure 3), FR1 is decomposed into FR1.1 and FR1.2 and they are mapped to DP1.1 and DP1.2, respectively. This process is repeated for FR2. Therefore, with the mapping of functional requirements to design parameters, a concept for an ECS of a manned fighter aircraft is presented in the Physical Domain block in (Figure 4).

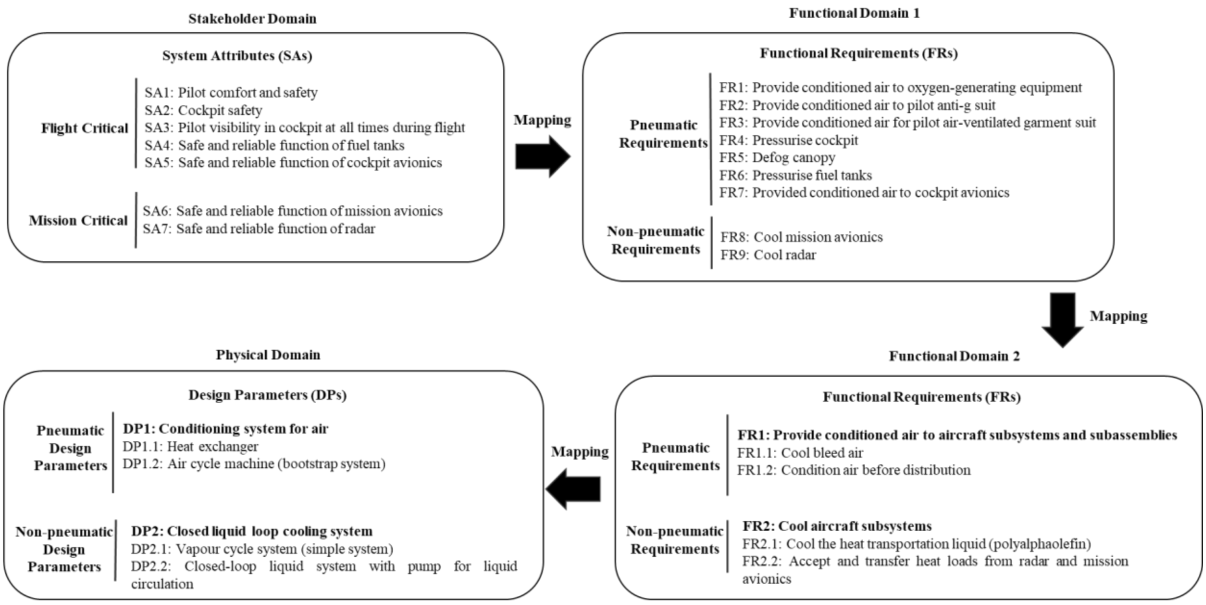

Figure 4. Mapping of System Attributes (SAs) in 'Stakeholder Domain' to Functional Requirements (FRs) in 'Functional Domain (1)' to 'Functional Domain (2)' to Design Parameters (DPs) in 'Physical Domain' for the ECS case study in this paper

5.7. Axiomatic design theory for analysing a vehicle system concept for functional independence and creating the vehicle system architecture

In axiomatic design theory, the design matrix of a concept must either be diagonal for an uncoupled design or triangular for a decoupled design to satisfy the independence axiom (Suh, 2007). With vehicle system concepts, the design matrix is more likely to be triangular rather diagonal since one functional requirement might need multiple design parameters in order to be satisfied. This can be noted in the design matrix shown in (Figure 5) for the ECS concept at hand. The bleed air must be cooled by the heat exchanger before being conditioned by the air cycle machine. Similarly, the PAO must first be cooled by the vapour cycle system before being transported around the closed-loop system.

Satisfying the independence axiom enables the vehicle system concept to be realised into an architecture. To satisfy the independence axiom, the ECS design parameters must be realised in the same sequence presented in the design matrix in (Figure 5). This is also an indication of the sequential order in which the design parameters (components) of the system must be connected. Using the ECS design matrix, the architecture of the ECS was generated and is shown in (Figure 5).

https://doi.org/10.1017/pds.2022.187 Published online by Cambridge University Press

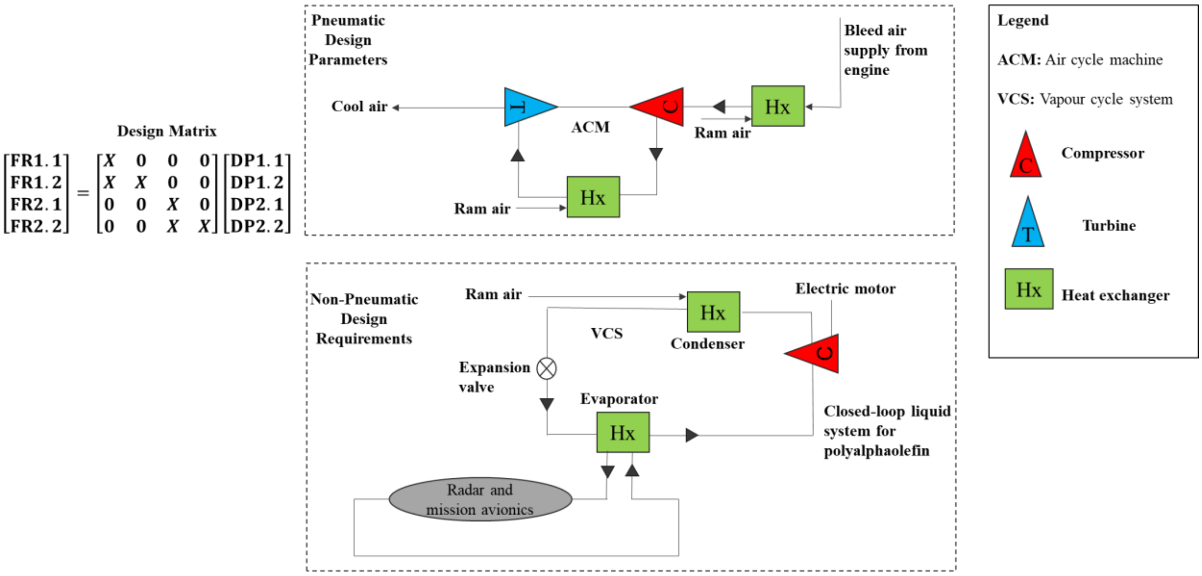

Figure 5. The design matrix and the architecture of the ECS concept generated in this paper

5.8. Design structure matrix method for hazard analyses of the vehicle system

A hazard analysis of a vehicle system is the final step in this framework and it is a two-step process. In the first step of a system hazard analysis, a component/subsystem DSM of the vehicle system of interest is created. The DSM would consist of what the system depends on, what is contained in the system, and what depends on the system. For the case study at hand, a DSM was created with the support of (Figure 2) and the ECS architecture in (Figure 5). The DSM is shown in (Figure 6) and it is a clear indication that the ECS consists of two separate systems - one to fulfil flight critical functions and the other to fulfil mission critical functions. The extent of the effect of the dependency could be quantified with an integer scale in the DSM. In the second step of a system hazard analysis, the component/subsystem DSM created for a vehicle system concept can be used to identify the redundancies for the system itself and/or the systems that depend on it. The choice of redundancies is usually a collective effort that includes the designer of the vehicle system of interest and the stakeholder(s) of the system. A spatial hazard analysis could be conducted in a similar way to the system hazard analysis.

1: Engine (for bleed air supply)
2: Heat exchanger (for bleed air)
3: Air cycle machine
4: Oxygen-generating system
5: Pilot anti-g suit
6: Pilot air-ventilated garment
7: Cockpit
8: Canopy
9: Cockpit avionics
10: Fuel tanks
11: Electrical power system (for power supply)
12: Vapour cycle system
13: Closed-loop liquid system
14: Mission avionics
15: Radar

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	x	x	x	x	x	x	x	x	x					
2		2	x	x	x	x	x	x	x	x					
3			3	x	x	x	x	x	x	x					
4				4											
5					5										
6						6									
7							7								
8								8							
9									9						
10										10					
11											11	x	x	x	x
12												12	x	x	x
13													13	x	x
14														14	
15															15

Figure 6. A component/subsystem DSM for the ECS concept generated in this paper

6. Concluding remarks

This paper proposes that vehicle system design commence at the aircraft concept stage rather than at the aircraft development stage when it typically does. However, the factors that significantly affect vehicle system design may not be defined at the aircraft concept stage. To harness the complexities of vehicle system design at the aircraft concept stage, a framework that is recursive and based on industrial reality

https://doi.org/10.1017/pds.2022.187 Published online by Cambridge University Press

is presented. The framework could become an established method in the future through the inclusion of techniques to predict the emergent behaviour of a vehicle system when integrated in an aircraft.

Acknowledgements

The author would like to thank the Swedish government agency, VINNOVA for funding the research presented in this paper. The author would like to extend her gratitude to Gunnar Holmberg for his contributions to this paper. The author would like to thank Hampus Gavel, Robert Hällqvist, and Sören Steinkellner for reviewing this paper.

References

Browning, T. (2001), "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions", *IEEE Transactions on Engineering Management*, Vol. 48 No. 3, pp. 292-306. doi:https://ieeexplore.ieee.org/document/946528

Crawley, E., Cameron, B. and Selva, D. (2015), *System architecture: strategy and product development for complex systems*, Pearson Education Limited, Essex. doi:https://www.pearson.com/us/higher-education/program/Crawley-System-Architecture-Strategy-and-Product-Development-for-Complex-Systems/PGM30308.html

Davies, A. and Hobday, M. (2005), *The business of projects: managing innovation in complex products and systems*, Cambridge University Press, New York. doi:https://www.cambridge.org/core/books/business-of-projects/B8701FF628BEF7B5D8378549FDE12092

Dezfuli, H., Stamatelatos, M., Maggio, G., Everett, C., Youngblood, R. et al (2010), NASA Risk-Informed Decision Making Handbook, Office of Safety and Mission Assurance NASA Headquarters. doi: https://ntrs.nasa.gov/citations/20100021361

Eppinger, S. and Browning, T. (2012), *Design Structure Matrix Methods and Applications*, MIT Press, Cambridge. doi:https://mitpress.mit.edu/books/design-structure-matrix-methods-and-applications

Friedenthal, S., Moore, A. and Steiner, R. (2015), *A Practical Guide to SysML: The Systems Modeling Language*, Morgan Kaufmann-Elsevier Inc., Waltham. doi:https://www.sciencedirect.com/book/9780128002025/a-practical-guide-to-sysml

INCOSE (International Council on Systems Engineering) (2015), *Systems Engineering Handbook: A Guide for Systems Life Cycle Processes and Activities*, John Wiley & Sons, Inc., Hoboken. doi:https://www.incose.org/products-and-publications/se-handbook

Kang, K., Cohen, S., Hess, J., Novak, W. and Peterson, A. (1990), *Feature-Oriented Domain Analysis (FODA) Feasibility Study*, Software Engineering Institute Carnegie Mellon University, Pittsburgh. doi:https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=11231

Meinicke, J., Thüm, T., Reimar, S., Benduhn, F., Leich, T. et al (2017), *Mastering Software Variability with FeatureIDE*, Springer, Cham. doi:https://link.springer.com/book/10.1007/978-3-319-61443-4

METOP GmbH (2020), *FeatureIDE*. [online] METOP GmbH. Available at: https://www.featureide.de/index.php (accessed 08.07.2021).

Perrow, C. (1984), *Normal accidents: Living with high-risk technologies*, Princeton University Press, Princeton. doi:https://press.princeton.edu/books/paperback/9780691004129/normal-accidents

Prencipe, A. (1997), "Technological competencies and product's evolutionary dyamics: a case study from the aero-engine industry", *Research Policy*, Vol. 25 No. 8, pp.1261-1276. doi:https://www.sciencedirect.com/science/article/abs/pii/S0048733396009006?via%3Dihub

Shah, R. K. and Sekulic, D. P. (2003), *Fundamentals of Heat Exchanger Design*, John Wiley & Sons, Inc., Hoboken. doi:https://onlinelibrary.wiley.com/doi/book/10.1002/9780470172605

Steinkellner, S. (2011), *Aircraft vehicle systems modeling and simulation under uncertainty*, [*Licentiate Dissertation*], Linköping University. doi:http://www.diva-portal.org/smash/get/diva2:415979/FULLTEXT01.pdfLINK

Suh, N. (2001), *Axiomatic Design: Advances and Application*, Oxford University Press, New York

Suh, N. (2007), "Ergonomics, axiomatic design and complexity theory", *Theoretical Issues in Ergonomics Sciences*, Vol 8 No. 2, pp. 101-121. doi:https://www.tandfonline.com/doi/abs/10.1080/14639220601092509

Taxén, L. and Pettersson, U. (2010), "Agile and Incremental Development of Large Systems", *7th European Systems Engineering Conference (EuSEC 2010)*, Stockholm, Sweden, May 23-26, 2010. doi:http://liu.diva-portal.org/smash/record.jsf?pid=diva2%3A755592&dswid=-1884