32nd Congress of the International Council of the Aeronautical Sciences
September 6-10, 2021
Pudong Shangri-La, Shanghai, China

EVALUATION ASPECT MAPPING FOR ACTUATION SYSTEM DESIGN

Christopher Reichenwallner[1], Felix Larsson[2], Alessandro Dell'Amico[2] & Petter Krus[1]

[1]Linköping university, SE-581 83, Linköping, Sweden
[2]Saab AB, SE-581 88, Linköping, Sweden

Abstract

The actuation system is one of many aircraft vehicle systems facing a possible conversion to an electrified solution. With the constant development of aircraft actuators, the design space of available actuation system architectures increases significantly. Extensive evaluation on a platform level is required to find the best choice among the available architectures. What this extensive evaluation should be centered around is however not apparent and raises the questions of what to examine and how to evaluate the possible alternatives. These questions are addressed in this paper through the formation of an ontological model, i.e., a formal description of information within a domain. The domain, also recognized as the target world, represent an aircraft with its actuation system and adjacent vehicle systems. Also included in the target world are evaluation aspects arranged in a hierarchy and centered around life cycle cost. The components within each system are modeled and described using descriptive elements, which in turn relate the components to the various evaluation aspects. These relations along with the evaluation aspects represent what is needed to be examined and acts as building blocks for future development of evaluation tools, such as mathematical models.

Keywords: Actuation system, Evaluation aspect, More electric aircraft, Systems engineering

1. Introduction

Electrification is a discussed topic in many different industries today, not least within the aircraft industry. Although centralized hydraulic servo systems for flight controls have been an enabler for modern aircraft, it is not certain that the technology will cope with future requirements on, for instance, energy efficiency and maintenance practices. Electrified actuators, such as the electromechanical actuator (EMA) and the electrohydrostatic actuator (EHA) [1], have emerged as possible alternatives for the hydraulic servo actuator (HSA). Expectations are that these alternatives share a lower power consumption compared to the conventional hydraulic solution, mainly due to the idea of power-on-demand [2, 3].

The best choice among the three available actuators is however not apparent. System requirements will impact the outcome, and extensive evaluation on a platform level is needed. Furthermore, the increased number of actuator alternatives will enable many different actuation system architectures [4, 5], which poses a challenge for system engineers in conceptual design stages. To find the optimal solution, all possible actuation system architectures should be evaluated with respect to design rules, system requirements, propagating effects to other systems and carefully selected evaluation criteria. A process for such an evaluation is proposed in Figure 1. Top level requirements, such as an aircraft design, allows for the generation of numerous actuation system architecture proposals. These proposals must be sorted with respect to design rules and safety regulations. With knowledge about aircraft performance requirements the remaining feasible proposals can be sized before they are evaluated with respect to various evaluation aspects.

There is research on the topic of actuation system evaluation which propose evaluative processes with various evaluation aspects employed. For instance, [6] propose a framework to find the most appropriate actuators with respect to mass and energy consumption for several control surfaces. This framework is extended in [7] to include geometric and aerodynamic models of an aircraft to additionally minimize the weight of a flap surface. Geometrical studies are also found in [8]. Advantages and disadvantages with control surface splitting are evaluated with respect to aircraft weight and geometric integration. The authors also present how actuator mass is affected by reducing the thermal resistance of the actuator housing and highlights the significance of cooling technologies. The framework presented in [9], and further surveyed in [10], is a three-step process for actuation system evaluation. It is composed of an architecture exploration and selection, the creation of a representative model for power flows between systems, and lastly, the composition of system models for the various represented systems. The framework evaluates the actuation system's impact on platform level with respect to aspects such as weight, fuel burned, and power extracted from the engine. It also incorporates redundancy rules required by regulations when considering the architecture exploration and selection. Additional frameworks such as [11] and [5] feature evaluation aspects such as weight, power extracted from the engine, fuel burn, efficiency and reliability.

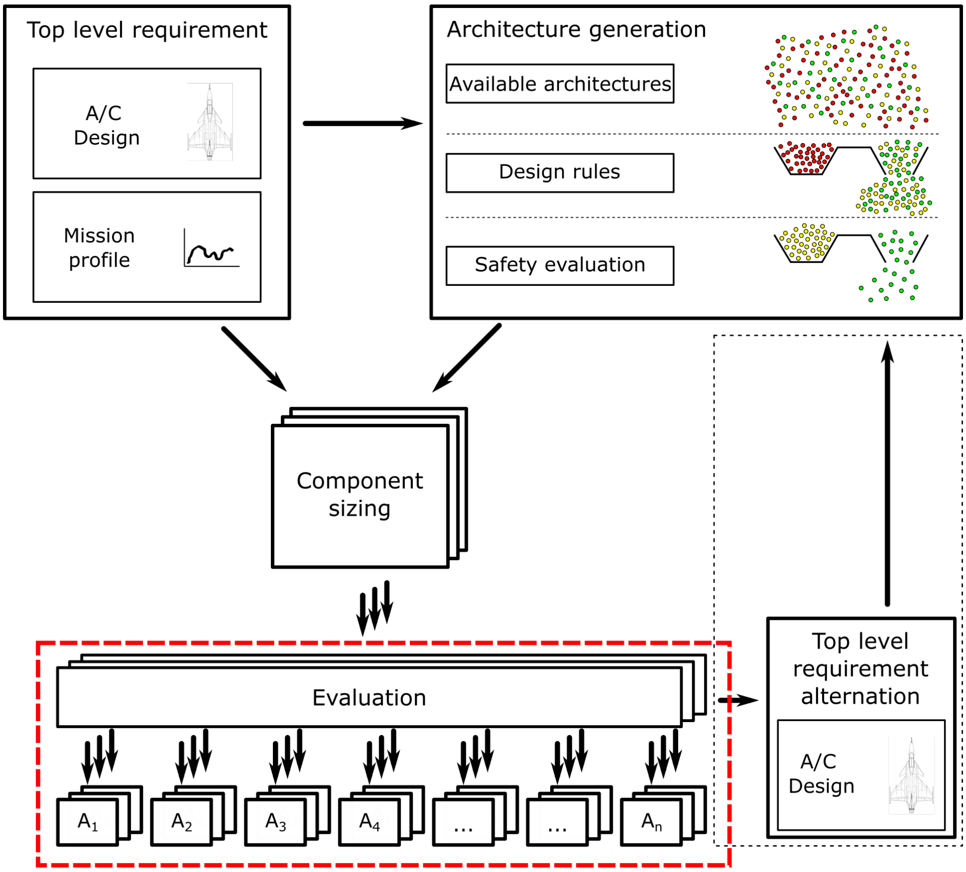


Figure 1 – Proposed process for actuation system evaluation. Actuation system architectures are generated for a set aircraft design and mission profile. The architectures are then evaluated against different evaluation aspects.

As shown among the presented frameworks there are many different evaluation aspects that may be included when considering evaluation of actuation systems. Deciding among them is a delicate task. Enough information, and more importantly, the correct information needs to be captured when the intention is to find an optimal solution for a platform. The weight aspect is highly represented among the previous mentioned frameworks, followed by fuel burn and power extracted from the engine. However, motivations for the selected evaluation criteria and why they would give an optimal solution are often lacking. The effect of adjacent aircraft systems likewise. This paper therefore aims at identifying possible evaluation aspects and show how these relate to a common reference for a comparison on equal terms. This is accomplished in an ontological approach in which an evaluation aspect hierarchy, concentrated around life cycle cost, is established. An actuation system, and its adjacent systems, are modeled to convey how these systems could possibly relate to the different evaluation aspects in the hierarchy. The work in this paper will serve as a groundwork for future quantitative models for the evaluative section highlighted in Figure 1.

The paper will continue with an introduction to ontology modeling and the modeling procedure used in this work. Chapter 3 will then explain the ontology model created with the modeling procedure. The chapter will firstly introduce the resulting class structure of the complete ontology model before presenting in detail the composition of classes and how these are related to other classes through various properties. Lastly presented in the chapter is the evaluation aspect hierarchy as modeled in the ontology. The paper finishes with a case study which exemplifies how system components relate to the different evaluation aspects in the hierarchy.

2. Ontology modeling as a tool for information representation

Ontology modeling as a tool for systems engineering is an unconventional method that appears to be receiving more attention [12, 13]. An example of ontology modelling used for exploration of system-of-system solutions can be found in [14], and [15] presents a design of an aircraft using ontology modelling. An ontology is defined as an "explicit specification of conceptualization" [16]. Conceptualization in this definition refers to the entities within the domain of interest, and the relationships that exist between them. Elaborated to the field of engineering, an alternative definition proposed by [17] is "a system (systematic, operational and prescriptive definitions) of fundamental concepts and relationships which shows how a model author views the target world, and which is shared in a community as building blocks for models". This latter definition reflects very well what is important in systems engineering. To describe a complex system is challenging but crucial in order to understand which means are necessary to realize the success of a system [18, 19].

Two popular ontology languages are the Resource Description Language (RDF) and the Web Ontology Language (OWL) [20]. OWL is built on RDF but adds a dimension by its inclusion of a descriptive logic reasoner. An ontology built in OWL is composed of classes, individuals and the properties that hold between them, as illustrated in Figure 2. The three elements together describe the perceived view of the target world.

There are typically two types of properties found in OWL, namely, object properties and data properties. Object properties relate two classes and/or individuals to one another, as Figure 2 illustrates. Data properties on the other hand relate classes and/or individuals to values of data. Furthermore, there are generally two types of classes found in OWL. These are categorized as primitive classes and defined classes. The difference between them lies in the type of conditions which describe them. The primitive class is described by *necessary conditions*, implying that if an individual is a member of a class, it is a necessity for that individual to fulfill the conditions describing the class. Deduction in the opposite way is however not possible. With other words, an individual that is described with the *necessary conditions* of a primitive class cannot be concluded to be a member of that class. A defined class on the other hand offers that extra direction of deduction. This is possible by describing the defined class with so called *sufficient* and *necessary conditions*. The *sufficient conditions* imply that the conditions are sufficient to conclude that an individual, which is described with the conditions, must in fact be a member of the defined class. More on properties, classes and how they are structured can be read in [21].

Properties and classes in combination with the descriptive logic reasoner can be used to, if structured correctly, check the ontology for any inconsistencies. Moreover, the reasoner can be used to infer more knowledge out of simpler information such as automatic classification of individuals and relationship establishment between classes. As an example, using the elements found in Figure 2 and assuming that an EMA is not known to be an actuator, the reasoner can infer that the individual *EMA* must indeed belong to the class *Actuator* due to its property *Actuates surface*. To let the reasoner deal with this sort of classification, instead of solving it manually, would be one possible strategy when composing models in an ontology. It should however be noted that the reasoner can constitute a limitation for larger ontologies due to its requirement of large computational resources, as indicated in [22].

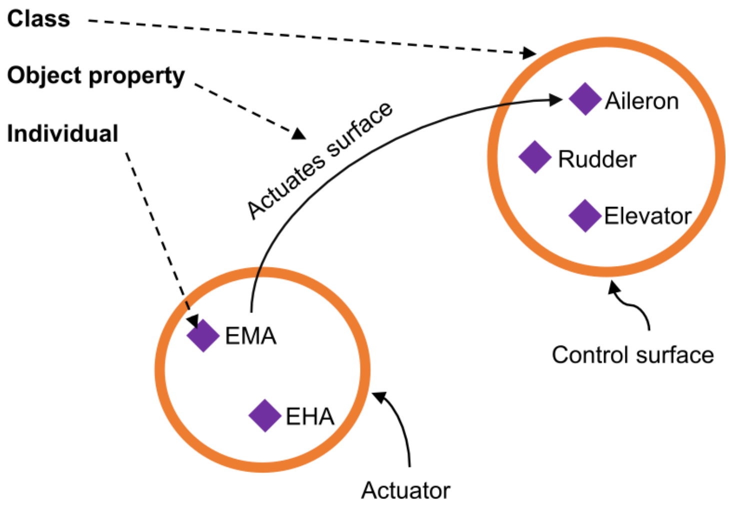


Figure 2 – Actuator individuals *EMA* and *EHA*, and control surface individuals *Aileron, Rudder* and *Elevator* as part of the two classes *Actuator* and *Control surface*. The *EMA* individual has the *Actuate surface* property and is related to the *Aileron* individual.

2.1 Modeling procedure

The target world in this paper was modeled through a four-step process. Although the steps are seemingly sequential it can be imaginable that the process is in fact iterative. Applying the process and its steps for one fraction of the target world at a time, or creating all the classes for the complete target world at first, is up to user preferences.

Step 1

The first step in the process involves the establishment of classes for the intended target world. These classes may represent various components, systems and structures and even non-physical evaluative information such as attributes, functions and capabilities. The classes should be arranged in a hierarchy of classes and sub-classes. For the reasoner to function properly, it is important to disjoint classes that are in fact unequal.

Step 2

The second step entails the definition of properties. As elaborated earlier, there are two types of properties. However, for the intended use in this work, the object properties were assumed to be enough. The intention was not to quantify descriptive attributes, such as weight, only to convey their relations.

Step 3

As a third step in the process, once the classes and properties are defined, is the description of the classes. At this step, class relations are established by using the previously identified and defined object properties. Moreover, classes are categorized as either primitive or defined type. In this work, the defined classes and the descriptive logic reasoner came useful especially for the evaluation aspect hierarchy and its association to system components. This is further elaborated in Chapter 4.

Step 4

The fourth and last step in the process involves the creation of individuals. These individuals are the most specific concepts in the target world and should be assigned to their corresponding classes. Furthermore, the individuals should be described in similar manner as the description of classes in the previous step. The description of the individuals should be done in accordance with the necessary conditions describing their corresponding classes.

3. Implementation of the target world

The ontology model was built in an iterative manner using the four-step modeling process presented above. It was modeled using the open-source software Protégé, which is based on OWL [23]. The target world was an aircraft of arbitrary type and size. It was found suitable to describe the aircraft by its structure, system and system components. The corresponding classes for these elements are found in the left column in Figure 3. In terms of systems and components, it was decided not only to include an actuation system, but also to include representations of supply systems and cooling systems as these are vital for a fully functional actuation system. The middle column incorporates classes of descriptive character. Classes for descriptive elements such as attributes, cooling solutions and various requirements constituted an integral part for the description of system components and their correlation to evaluation aspects. Lastly, seen in right column is the hierarchy of evaluation aspects, here presented in terms of life cycle cost. The resulting class hierarchy with its classes and their usefulness will be further described in this chapter.

Figure 3 – Class hierarchy for the target world.

3.1 Composition of classes

The descriptive element classes found in the middle column in Figure 3 above had a central role in the composition of the other classes, especially for the aircraft system component classes and the life cycle cost classes. Through the descriptive elements and the defined object properties from step 2 in the modeling procedure, it was possible to describe the remainder of classes. An example using the actuator class will follow.

Generally, aircraft system component classes were assumed to be described by, at least, static and dynamic attributes. The definitions of the two concepts are as follows:

- **Static attributes** – Attributes that are constant over the course of a mission profile. For instance, weight.
- **Dynamic attributes** – Attributes that are dependent on how the component is used. For instance, power consumption.

Other descriptive elements such as various requirements have an impact on these attributes, as suggested in Figure 4. While performance requirements have an, conceivably, obvious impact on a components weight and size, functional requirements do not. Functional requirements are often overlooked, although they are usually the most difficult to convey [24]. For different components to be fully comparable, it is a necessity for them to fulfill the same requirements.

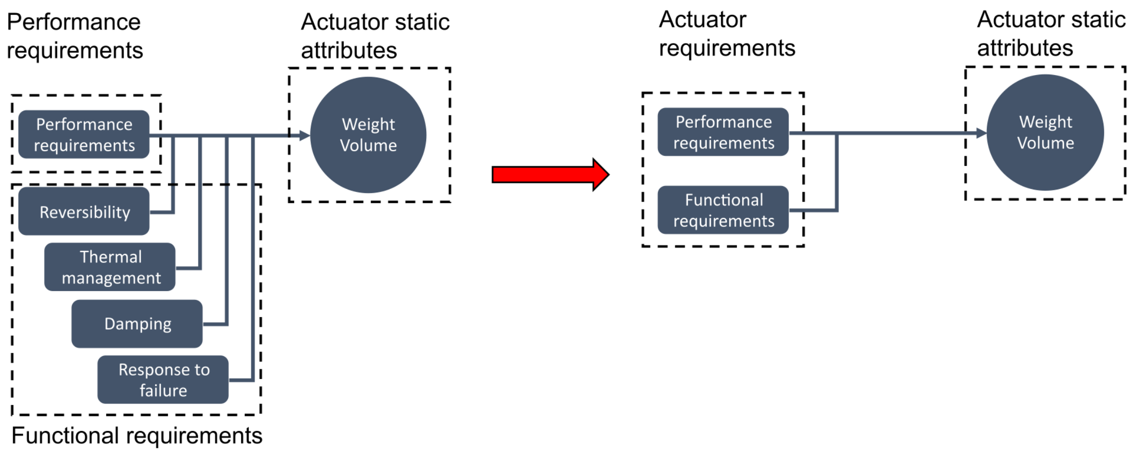


Figure 4 – Relation between various requirements and static attributes. An example applied for an actuator.

A visual representation of the *Actuator* class is seen in Figure 5 and will be used to illustrate the inclusion of descriptive elements in the ontology. Object properties, displayed on the right side in the figure, relate the *Actuator* class to different descriptive element classes. The *Actuator* class was modeled as a primitive class, suggesting it to be necessary for its members to be described by the related descriptive elements, i.e., the necessary conditions. As suggested in Figure 4, a components attributes are consequences of the requirements set on the component. This is illustrated in the figure by the *influence* property displayed among the object properties on the right side. In essence, this view of an actuator is the building block that represent models, as per described in the ontology definition provided by [17]. What is conveyed is the need for evaluative mathematical models which describe the relationship between the various requirements and attributes.

It is important to note that the *Actuator* class in Figure 5 is still the general view of an actuator. This class will include individuals as instances of the class, in accordance with the fourth step in the modeling procedure. An illustration of an actuator individual is found in Figure 6. This individual is a member of the *Electric actuator* class and is described with various descriptive elements. Liquid cooling as cooling solution and weight as static attribute, to mention a few. The descriptive elements are in turn members of the different descriptive element classes, just as the actuator individual is a member of the *Electric actuator* class.

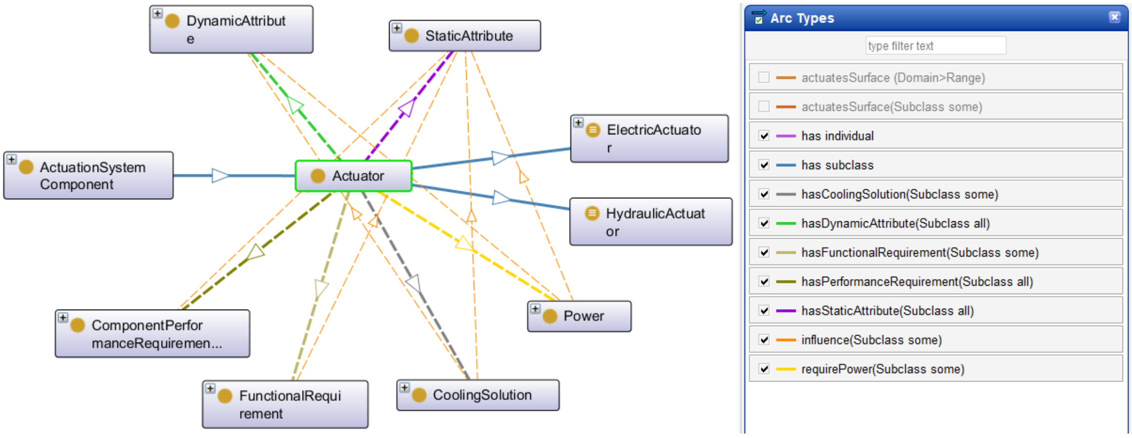


Figure 5 – The Actuator class with its relation to other classes.

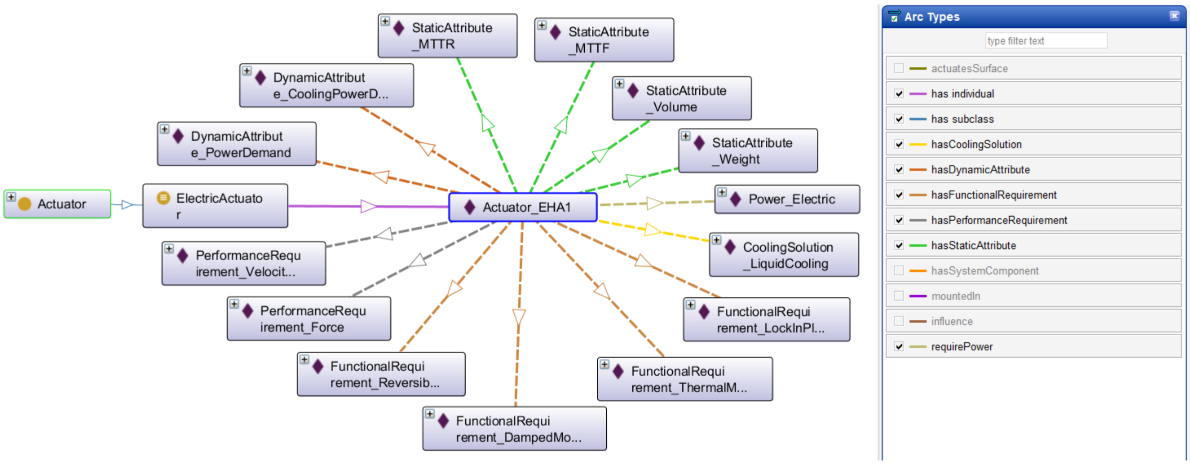


Figure 6 – One actuator individual related to descriptive elements through object properties.

3.2 Evaluation aspect hierarchy

The evaluation aspect hierarchy was created with the concept of life cycle cost (LCC) as a baseline. LCC analysis can serve as valuable input to decision making for system engineers and project teams in conceptual stages. It can aid in understanding how design decisions will impact the total cost for a system and thus support trade studies that covers a broader span of a systems lifetime [18, 25]. Although there are many definitions on how LCC should be expressed, it was chosen in this work to express LCC in terms of acquisition cost, operational cost, maintenance cost and disposal cost. Parallels can be drawn to existing evaluation frameworks which, for most examples, merely covers evaluation aspects associated with operational cost. Product qualification and installation practices, as examples for acquisition costs, and maintenance frequency and time, as examples for maintenance costs, are however some aspects that could be included to obtain the border span analysis.

Figure 7 depicts the LCC hierarchy as modeled in the ontology. This illustration is a visual representation of the LCC class and its sub-classes found in the right column in Figure 3. The hierarchy was established in a "is-a" manner, hence the suffix *contributor* in the class names. With that in mind, the figure should be read from right to left, and as an example, this would equate saying that the power extracted from the turbine, here represented by the *TurbinePowerTakeOffContributor* class, is a contributor to fuel burn, represented by the *FuelBurnContributor* class. Likewise, it can be furthered said that fuel burn is a contributor to operational cost, which in turn, is one out of four cost contributors to LCC.

The LCC serves as a common reference for comparison among the different cost categories in the hierarchy. Just as Figure 5 depicted the view of an actuator and its need for evaluative mathematical models, every class in the LCC hierarchy represent the same need, but in several senses. Models are needed to evaluate the quantities represented by the different evaluation aspect classes and relate these quantities to aspects on higher levels. The power extracted from the turbine needs to be converted into consumed fuel, for instance. The other facet of needed models is the relation between component attributes and the quantities represented by the different evaluation aspects. For example, a component's volume, when mounted in a wing, could possibly impact the shape of the wing. To evaluate how this penalizes aerodynamic drag, it is first needed to relate the volume attribute to a quantity better suited for drag evaluation, in this case, a drag coefficient. How the attributes identified in this work relate to the different evaluation aspects is illustrated in Figure 7.

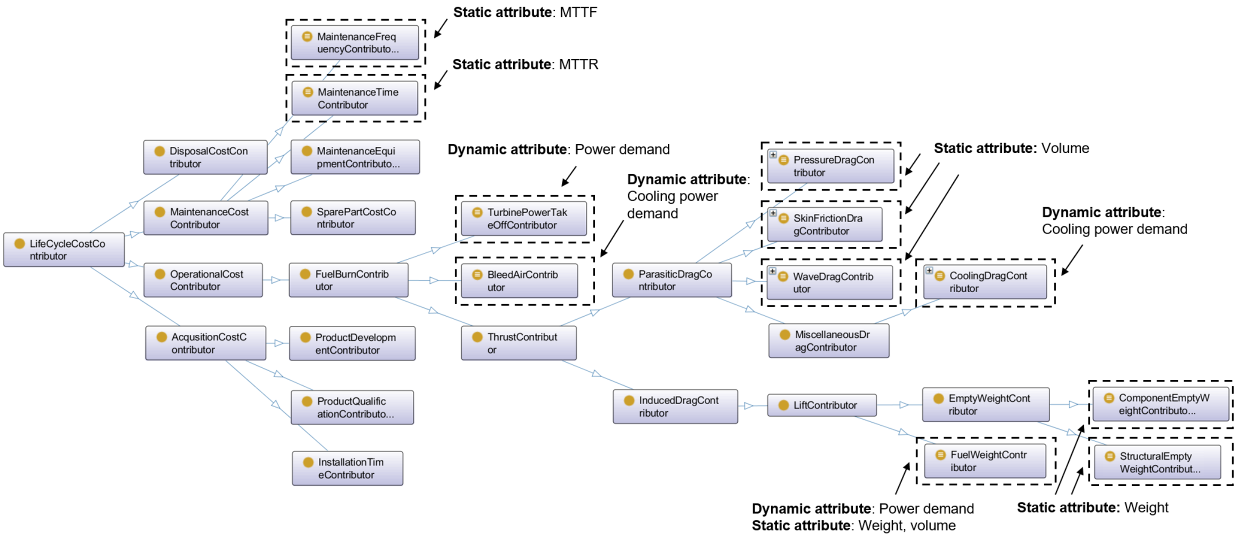
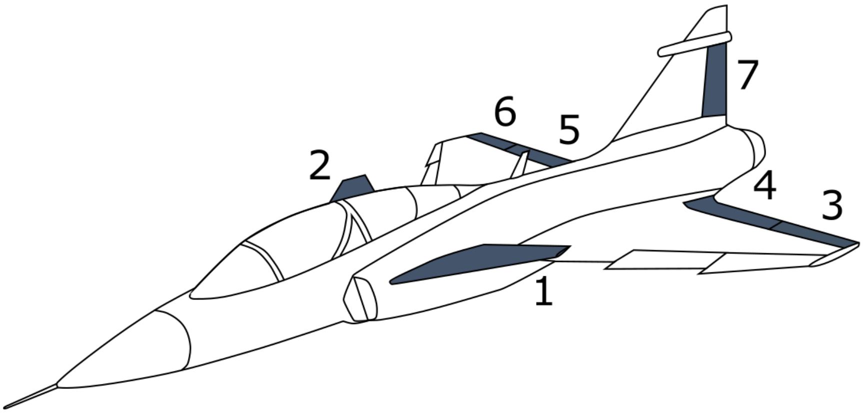


Figure 7 – Evaluation aspect hierarchy based on life cycle costs. Classes represent the need for evaluative mathematical models.

4. Case study

The classes in the LCC hierarchy consist of both primitive and defined classes. As elaborated on in Chapter 2, defined classes can be used in combination with the descriptive logic reasoner to automatically classify individuals. This functionality can thus be used to associate and represent any system component which by any means will impact the evaluation aspect in question.

To examine the classification facilitated by the descriptive logic reasoner, a simple case study was performed. This case study included modeling of an aircraft by means of individuals in Protégé. As mentioned in the modeling procedure description, the individuals represent the most specific concepts in the target world, which for this application were structural components and system components associated with an actuation system. It was assumed that the aircraft had a delta-canard configuration with a total of seven control surfaces, in accordance with Figure 8. The aircraft had one actuation system consisting of five EHA:s and two EMA:s. Due to assumed safety requirements, each actuator was supplied by at least two supply systems. Similarly, each supply system was supplied by at least two power supply units, i.e., electrical generators, hydraulic pumps and liquid cooling pumps. Furthermore, it was assumed that the EMA:s were cooled by means of ambient air while the EHA:s were cooled using a liquid cooling system and the fuel as a heat sink. The individuals describing the aircraft were assigned properties according to the necessary conditions in their corresponding classes, as elaborated in Chapter 3.1.



Control surface	Actuation system
1	EHA
2	EHA
3	EMA
4	EHA
5	EHA
6	EMA
7	EHA

Figure 8 – Delta-canard aircraft with seven control surfaces and one actuation systems.

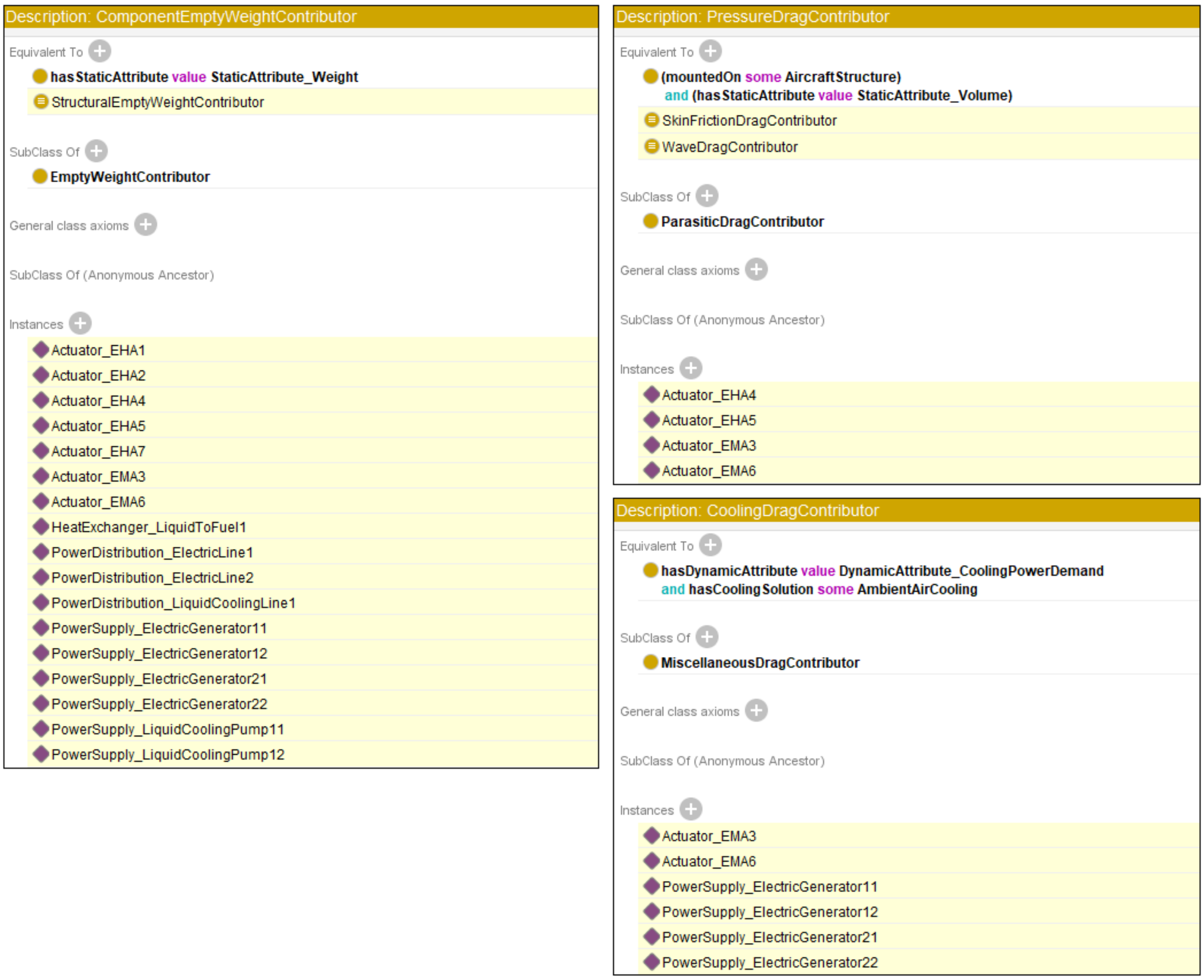


Figure 9 – Individual classification with respect to evaluation aspects of component empty weight (left), pressure drag (top right) and cooling drag (bottom left).

With individuals created for every structure and system component it was possible to invoke the descriptive logic reasoner to associate the system components with the various evaluation aspects. The reasoner searches for the individuals which fulfill the necessary and sufficient conditions set in the evaluation aspect classes and classifies them accordingly. This led to the classification of individuals found in Figure 9. The figure presents three evaluation aspect classes from the LCC hierarchy, namely the *ComponentEmptyWeightContributor* class, the *PressureDragContributer* class and the *CoolingDragContributor* class. The necessary and sufficient condition to be included in the *ComponentEmptyWeightContributor* class is to have a weight static attribute, for the *ParasiticDragContributor* class to have a volume static attribute and be large enough to not fit within the aircraft structure, suggested by the *mountedOn* property, and for the *CoolingDragContributor* class to have a cooling power demand and be cooled by means of ambient air.

Since every system component will naturally have a weight static attribute, all of them are classified as members of the *ComponentEmptyWeightContributor* class. The key result from this classification is rather the clarification of which components need to be evaluated in terms of weight for the system solution. If a liquid cooling system is added for the sole purpose of maintaining actuator temperature, then this cooling system must be considered as a weight penalty for the actuation system. The same reasoning applies for the *CoolingDragContributor* class. There are four power supply units as members of this class. If these units are included solely to supply the actuation system with power, then their demand of cooling power must be considered a penalty for the actuation system. If they are added to partly supply the actuation system with power, then this penalty must be weighed accordingly. As for the *PressureDragContributor* class it displays the system components which will impact the aerodynamic drag of the aircraft due to its volume and position of installation.

5. Discussion

The procedure used for modeling the target world in this paper can be considered a generic procedure that would suit implementations of other target worlds just as good. What potentially could be added as a first step in the process is a thorough exploration of already existing relevant ontologies. For instance, the aircraft ontology found in [15] could have been used as a basis on which the ontology created in this paper was built upon. The aircraft ontology by [15] would in such a case act as an upper ontology with the purpose of defining general guidance for how properties and relations should be structured. This would allow for more domain-specific ontologies to be implemented, with assurance of interoperability among them. One joint ontology could be favorable for further analysis of dependencies among the different aircraft systems and/or for the discovery of implicit knowledge with the aid of the descriptive logic reasoner.

In essence, there is no correct way of creating an ontology since relations can be represented in many ways. The created ontology in this paper does, for instance, omit quantitative data, although several of the descriptive elements would intuitively convey such information. Quantitative data can be represented in Protégé by insertion of data properties. The resulting evaluation aspect hierarchy and the description of classes would not have changed if data properties had been used. However, the intended use of the created ontology was not to quantify describing elements such as various attributes and performance requirements. The describing elements were instead used as a mean to describe the view of system components and associate the system components with the various evaluation aspects. Therefore, it was deemed sufficient to model the describing elements as individuals and relate these to system components using object properties.

The resulting evaluation aspect hierarchy in Figure 7 is built upon the assumption of breaking down LCC into four categories. The hierarchy could possibly be shaped differently, depending on how the categories are defined. Maintenance cost could, for instance, be argued to be a sub-category to operational cost as it is most likely a consequence of direct use of a system. Such an alternation would however not affect the main idea of studying a solution's impact on different aspects bound to cost. Furthermore, some evaluation aspects in the hierarchy are more difficult than others to relate to different component attributes. Maintenance equipment, product development and product qualification to mention a few. There are no direct attributes that will influence these categories, but they are equally important to evaluate if a holistic view is sought.

The holistic view in correlation to system components can be questioned, as there are certainly more components to be included in each system than presented in this paper. Some examples could be energy storages, such as batteries and accumulators, or different safety equipment, such as pressure relief valves in the hydraulic domain. This is an issue of fidelity level and whether the omitted components are expected to have relatively large impact on the different cost categories. However, this uncertainty is not to be answered in this paper. Further studies involving quantification is needed to reach a conclusion.

6. Conclusion

In coherence with the ontology definition provided by [17], this paper has shown how an ontology model can be used to convey a problem of high complexity for the purpose of using it as a reference for future model development. Various evaluation aspects have been identified and to get a broader span analysis of an actuation system more evaluation aspects than those bound to operational cost must be examined. Furthermore, system components in close relation to an actuation system must also be included in the evaluation.

7. Contact Author Email Address

Corresponding author: Christopher Reichenwallner, christopher.reichenwallner@liu.se.

8. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

[1] Jianming L, Zhiyuan Y, Yuping H and Zhiguo. A Review of electromechanical actuation system for more electric aircraft. *International Conference on Aircraft Utility Systems*, Beijing, China, 2016.

[2] Maré J-C. *Aerospace Actuators 2*. 1st edition, ISTE Ltd and John Wiley & Sons Inc, 2017.

[3] Zhiqiang B, Yu L and Zidong L. Flight control system architecture comprehensive evaluation based on electrical actuators for more electric aircraft. *Recent Advances in Aerospace Actuation Systems and Components*, Toulouse, France, 2018.

[4] Trochelmann N, Thielecke F, Behr R M and Hamm M. Electro-hydraulic system architectures for MEA – comparison of central and zonal power packages, *Recent Advances in Aerospace Actuation Systems and Components*, Toulouse, France, 2018.

[5] Qi H, Fu Y, Qi X and Lang Y. Architecture optimization of more electric aircraft actuation system. *Chinese Journal of Aeronautics*, vol 24, pp. 506-513, 2011.

[6] Safavi E, Gopniath V, Ölvander J and Gavel H. Conceptual optimization of aircraft actuator systems. *Recent Advances in Aerospace Actuation Systems and Components*, Toulouse, France, 2012.

[7] Safavi E, Gopniath V, Ölvander J and Gavel H. Multidisciplinary optimization of aircraft vehicle system for conceptual analysis. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Dallas, USA, 2013.

[8] Delbecq S, Budinger M, Leray Dand Piaton J. Optimization of primary flight control actuation system using parametric sizing models of actuators, power electronics and structural analysis. *Recent Advances in Aerospace Actuation Systems and Components*, Toulouse, France, 2018.

[9] Garriga A G, Govindarajua P, Ponnusamya S S, Cimminob N and Mainini L. A modelling framework to support power architecture trade-off studies for more electric aircraft. *Transportation Research Procedia*, Romania, Vol. 29, pp 146-156, 2018.

[10] Garriga A G, Mainini L and Ponnusamy S S. A machine learning enabled multi-fidelity platform for the integrated design of aircraft systems. *Journal of Mechanical Design*, vol 141, 2019.

[11] Chakraborty I, Mavris D N, Emeneth M and Schneegans A. A methodology for vehicle and mission level comparison of more electric aircraft subsystem solutions: Application to the flight control actuation system. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 229, 2015.

[12] Chourabi O, Pollet Y and Mohamed B A. Ontology based knowledge modeling for system engineering projects. *Research Challenges in Information Science*, 2008.

[13] van Ruijven L C. Ontology for systems engineering. *Conference on Systems Engineering Research*, Rotterdam, Netherlands, 2013.

[14] Knöös Franzén L, Staack I, Jouannet C and Krus P. An ontological approach to system of systems engineering in product development. *The 10th Aerospace Technology Congress*, Stockholm, Sweden, 2019.

[15] Ast M, Glas M and Roehm T. Creating an ontology for aircraft design. *Deutscher Luft- und Raumfahrtkongress*, 2013.

[16] Gruber, T. What is an ontology?. Available: `http://www-ksl.stanford.edu/kst/what-is-an-ontology.html`. Accessed 2021-05-04.

[17] Kitamura Y. Roles of ontologies of engineering artifacts for design knowledge modeling. 2006.

[18] INCOSE. About systems engineering. 2019. Available: `https://www.incose.org/about-systems-engineering`. Accessed 2021-05-04.

[19] Department of defense – systems management college. Systems engineering fundamentals. Virginia, USA, Pp. 31-34. 2001

[20] Kalibatiene D and Vasilecas O. Survey on ontology languages. *Perspectives in Business Informatics Research - 10th International Conference*, Riga, Latvia, 2011.

[21] Horridge M, Knublauch H, Rector A, Stevens R, Wroe C, Jupp S, Moulton G, Drummond N and Brandt S. A practical guide to building OWL ontologies using Protégè 4 and CO-ODE tools edition 1.3. *University of Manchester*, Manchester, England, 2011.

[22] Lampoltshammer T and Wiegand S. Improving the computational performance of ontology-based classification using graph databases. *Remote Sensing*, 2015.

[23] Stanford University. An open-source ontology editor and framework for building intelligent systems. Available: `https://protege.stanford.edu/`. Accessed 2021-05-04.

[24] Maré J-C. *Aerospace Actuators 1: Needs, Reliability and Hydraulic Power Solutions*. 1st edition, ISTE Ltd and John Wiley Sons Inc, 2016.

[25] Hjelm M. A Journey of change in the aircraft industry. 2016. Available: `https://saabaircraftindustry.com/en/roads-to-new-capability/low-life-cycle-cost/effective-maintenance-concepts/`. Accessed 2021-05-04.