MASTER’S THESIS

Similarity in Design: A Framework for Evaluation

Master’s Educational Program __Space and Engineering Systems__

Student: Alsalehi, Suhail Hasan Hael

Signature, Name

Research Advisor: Crawley, Edward F.

Signature, Name, Title

Moscow 2019

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Сходство в дизайне: Структура для оценки

Магистерская образовательная программа__ Космические и Инженерные Системы_

Студент: Аль Салехи Сухаиль Хасан Хаил

подпись, ФИО

Научный руководитель: Кроули, Эдвард Ф.

подпись, ФИО, должность

Москва, 2019

"*Aim for the moon. If you miss, you may hit a star*"

W. Clement Stone

ABSTRACT

Engineered systems are rapidly increasing in complexity. Complexity, if not managed properly, can lead to inefficient design, higher risk of failures, and/or unexpected costs and schedule delays. This is specifically true during early design. The early phase of design, conceptual design, is characterized by high levels of uncertainty and fuzziness. In this phase, designers rely mostly on expertise to evaluate solution variants. Although researchers have developed many tools for solution variants evaluation during conceptual design, these tools have some drawbacks. A major drawback is that existing tools do not account for the system architecture during evaluation. System architecture is crucial for the design process and can be captured by comparing the architectures of solution variants. Comparing architectures established the need to distinguish near and far architectures. In other words, we need to capture the similarity/distance between pairs or architectures. However, depending on the specific application, the notion of 'similarity' may greatly vary. The notion of similarity, in the design field, needs to be addressed and a general formulation for similarity in design will help in the evaluation process.

This work started by reviewing the literature on evaluation methods and similarity. Next, we identified three main aspects of a system. The first is the architectural decisions which are usually addressed during conceptual design. The second and third aspects are the crisp features and relative features. The last two aspects deal with form and function attributes, and the relationship between them. Then, we chose suitable similarity measures for the purpose of comparing system architectures and showed how such measures can be combined with the existing evaluation methods. After that, we discussed the potential applications of our proposed framework. In general, the research was carried out by reviewing the existing literature, conducting case studies and a survey study. In addition, very valuable input and guidance were constantly received from professionals in the design field. In the thesis body, we included a brief review of the literature, presented a case study on commercial aircraft, and discussed the learning and outcomes of the survey study.

To frame solution variants evaluation as a general problem, we described a framework that utilizes information about the three aspects of a system – architecture, crisp features, and relative features. First, we introduce the term "architectural distance" to reflect how different the architectures are. To calculate the architectural distance, we use binary similarity measure. Next, we demonstrated how information about architectural distance can be used along with other indicators that are calculated with evaluation methods for crisp and relative features. For crisp features, we introduced a general formulation to calculate "performance index". Performance index is an indicator that accounts for multiple form and function attributes. Next, we explained the application of fuzzy logic to evaluate relative features. The combination of these three tools makes up the proposed framework. In addition to solution variant evaluation, there is a wide range of applications for the framework. Applications can include clustering similar architectures in the design space, studying the history of innovation for a certain product, and design cataloging and documentation.

To sum up, the research distinguished three aspects of a system and proposed a framework for solution variants evaluation by comparing their architectures. We demonstrated the work and applications of the framework with a case study. In addition, we carried out a survey study and discussed its results. The framework provides powerful insights about the architecture for both novice and seasoned designers. The notion of architectural distance has a wide variety of application that includes, but are not limited to, solution variants evaluation, reducing the size of design space, design documentation and cataloging, and studying the history of innovation for products.

РЕФЕРАТ

Инженерные системы быстро усложняются. Сложность, если не будет должным образом управляться, может привести к неэффективному дизайну, более высокому риску сбоев и / или неожиданным затратам и задержкам в графике. Это особенно актуально в ранней стадии разработки и дизайна. Ранняя фаза дизайна, концептуальный дизайн, характеризуется высоким уровнем неопределенности и нечеткости. На этом этапе дизайнеры полагаются в основном на опыт для оценки вариантов решения. Хотя исследователи разработали много инструментов для оценки вариантов решения во время концептуального дизайна, эти инструменты имеют некоторые недостатки. Основным недостатком является то, что существующие инструменты не учитывают архитектуру системы во время оценки. Архитектура системы имеет решающее значение для дизайна и может быть получена путем сравнения архитектур вариантов решения. Однако в зависимости от конкретного применения понятие «сходство» может сильно различаться. Понятие сходства в области дизайна необходимо рассмотреть, и общая формулировка сходства в дизайне поможет в процессе оценки.

Эта работа началась с обзора литературы о методах оценки и сходства. Далее мы определили три основных аспекта системы. Первый — это архитектурные решения, которые обычно учитываются при концептуальном дизайне. Второй и третий аспекты - четкие черты и относительные черты. Последние два аспекта касаются атрибутов формы и функции и взаимосвязи между ними. Затем мы выбрали подходящие меры сходства с целью сравнения системных архитектур и показали, как такие меры можно комбинировать с существующими методами оценки. После этого мы обсудили возможные применения предлагаемой нами структуры. В целом исследование проводилось путем обзора существующей литературы, проведения тематических исследований и опроса. Кроме того, очень ценные материалы и рекомендации постоянно получали от специалистов в области дизайна. В теле диссертации мы включили краткий обзор литературы, представили тематическое исследование на коммерческом самолете и обсудили изучение и результаты исследования.

Чтобы представить оценку вариантов решения как общую проблему, мы описали структуру, которая использует информацию о трех аспектах системы - архитектура, четкие черты и относительные черты. Во-первых, мы ввели понятие «архитектурное расстояние», чтобы показать, насколько разные архитектуры. Для вычисления архитектурного расстояния мы используем двоичную меру сходства. Далее мы продемонстрировали, как можно использовать информацию об архитектурном расстоянии вместе с другими показателями, которые рассчитываются с помощью методов оценки четкости и относительных характеристик. Во-первых, для четких характеристик мы ввели общую формулировку для расчета «индекса производительности». Индекс производительности — это один параметр, который может учитывать несколько характеристик формы и функции. Затем мы объяснили применение нечеткой логики для оценки относительных характеристик. Комбинация этих трех инструментов составляет предлагаемую структуру. В дополнение к оценке варианта решения существует широкий спектр применений для структуры. Приложения могут включать кластеризацию аналогичных архитектур в области дизайна, изучение истории инноваций для определенного продукта, а также каталогизацию и документирование систем.

Подводя итог, исследование выделило три аспекта системы и предложил структуру для оценки. Мы продемонстрировали работу и применение структуры на примере. Кроме того, мы провели опросное исследование и обсудили его результаты. Структура обеспечивает мощные идеи как для начинающих, так и для опытных дизайнеров. Концепция архитектурной дистанции имеет широкий спектр применений, которые включают, но не ограничиваются, оценку вариантов решения, уменьшение размера пространства решений, проектную документацию и каталогизацию, а также изучение истории инноваций для продуктов.

ACKNOWLEDGEMENTS

This work is another milestone in my academic life. It marks more than 6 years since I first came to Russia from Yemen, where I grew up. Throughout these years, I have had many tough times. It is these tough times that turned out to be the most profound. I learned a big deal about engineering, life, and people. Every single moment in my life - the good ones and the bad ones - has added up to this moment. And I am very thankful and fully committed to continuing my career in academia and contributing to building a better world.

Throughout the course of my master thesis work, I was lucky to be supervised by the very best mentors. This work would not have been accomplished without the constant guidance, feedback and mentoring from many people.

In the first place, I would like to express my deepest appreciation to my supervisor, Professor Crawley (Ed). You have made a great influence on my progress both in my academic and personal life. Working with you has been a great pleasure. With every conversation, I received valuable advice and important insights. Dear Ed, thank you.

I'd like to acknowledge the assistance of my colleagues at the MIT System Architecture lab and the Skoltech Space Center. They played a decisive role in shaping this work at the early stages. I specifically would like to thank Markus and Slava. I gratefully acknowledge help from Demetrios who provided me with the database for the case study. Liza and Bruce – thank you for helping me in designing the survey study and conducting the questionnaire.

It might live thousands of kilometers away from my family, but their constant support is always a determining factor in my success. I'm deeply indebted to my twin sister, Thuria, my two brothers Hayl and Mohammed, and to my beloved mum and dad. Your unconditional support and trust have provided me with motivation and determination. Your teachings were and will always be the light in my way.

Last but not least, I would like to express my thanks to my friends. Those who believed in me and were there for me in the darkest moments. You who were my family away from home. I cannot imagine how I would have made it through these years without you.

To the beautiful people. To the end of an era. To the beginning of an era.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

DSE -	**Design Space Exploration**
MBSE -	**Model-Based System Engineering**
ARM -	**Antenna Release Mechanism**
OTU -	**Operational Taxonomic Units**

CHAPTER 1:

INTRODUCTION

Design in the broadest sense describes innovating or improving the means (either material or immaterial) for addressing human needs[1]. Central to this is the designer, a denominator for the actors in the design activity. Humans have been practicing design since ever, let that be making farming tools, creating services, developing complex systems, conducting scientific experiments and so on. Throughout history, the rapid development in technology and services has driven the need for rigor design practices. Although the study of design as science came later humans have been repeating specific practices over and over in what seems like mental guidelines or blueprints for the practice of design. Design can be best viewed and studied for engineered systems (next "systems"). To simplify, a system is a combination of interacting elements that work in synergy to collectively perform a useful function. Complex systems and the rapid technological evolution drive the need for formal approaches to design. The emerge of design research can be thought of as a result of this need. And while the research on design has contributed hugely to the current state of development, there is still space for development and gaps in the research. Complexity, fuzziness, and uncertainty are only a few issues that designers deal with in their daily practice. Addressing these issues by research is important to improve design efficiency and bring the highest benefit at the lowest cost.

1.1 Background and motivation

The design process requires significant engineering experience, intuition, and creativeness. Throughout the past century, many guidelines, methods, and tools have been developed to assess designers. In addition, design witnessed important advancements in the research on system architecture. This has opened new doors for developing general-use design tools. However, developing a unified design theory is still a big challenge. This is due to uncertainty, fuzziness, and inconsistency that accompanies the design process. Researchers are working in many frontiers to advance the field of design. One important frontier is the design space exploration. Design space exploration requires constant human input and guidance. This bounds exploration by the knowledge and abilities of the human designer, who is subject to limitations and biases.

The ever-increasing complexity requires the designer to keep many variables in mind when evaluating solution variants in the design space. The more complex the system, the harder it is to evaluate. This is especially true during early design (conceptual design) when little is known about the system parameters. Although there are many evaluation methods in the literature, they are often case-based and lack a general formulation. Moreover, most methods are not applicable during the conceptual design phase. This gap brings forward the need for a new method to compare and evaluate solution variants. Such a method should enable the designer to compare solution variants using available information about the architecture and other parameters.

In this work, we address this need and develop a framework that uses information about the architecture to evaluate solution variants. Ultimately, this will help to improve the design space exploration during conceptual design.

1.2 General objective

Design often involves comparing different solution variants and choosing between them. The decision about which solution variant to go with is usually not trivial and requires further analysis. Designers often evaluate solution variants by comparing their parameters and characteristics. However, such information can be unavailable during conceptual design. The need for additional information about the architecture itself is critical as it can provide important insights for the designer about solution variants.

The overarching aim of this thesis is to develop a rigorous framework for evaluating solution variants in the design space by comparing their architectures using similarity measures.

1.3 Literature review

As we will be developing a framework, our research will touch different topics related to design theory, conceptual design, system architecture, engineering design evaluation methods, and similarity and its applications. When studying design, talking about the design theory is inevitable. Design theory is an essential part of the design body of knowledge as it shapes the design principles, knowledge and practice. Another important topic is the conceptual design. While the design process is a broad term that covers the whole lifecycle of a system, the conceptual design is the early phase that determines the broad outline of the system. This makes it in the core of our study. In the same context as, conceptual design is system architecture. Architecting a system is a critical step in developing a good architecture. And good architectures leads to good systems. This puts the focus of our work in the early design phases.

As mentioned before comparing architectures to evaluate solution variant requires unconventional tools. For this purpose, we will explore the existing evaluation methods in engineering design before we dive into similarity. But what is design theory? What design theories are there? What do we mean by system architecture? and Afterall what is unique about the conceptual design phase? and how can we capture similarity?

Before commencing with the main body of the thesis, we will try to answer these questions and carry out a review of the existing literature. This is aimed at giving an overview of academic knowledge in the topics of design, conceptual design, system architecture, evaluation methods, similarity, and its applications.

1.3.1 Design

Design is both science and art. Hubka, like many others, saw design science as a system of logically related knowledge which should contain and organize the complete knowledge about and for designing[2]. Design as a process is thought of as a practice, mediated by scientific and engineering knowledge, aimed at transforming a set of needs into an artifact[3]. It has been a subject of study for decades and researchers have developed design theories to capture and guide the design process. These theories constitute the field of study known as Design Theory. Design theory is a subfield of design research concerned with theoretical approaches towards understanding and delineating design principles, design knowledge, and design practice.

Although a single unifying design theory is still far from reach, there have been many efforts to improve the rigor and unity of design theories. In the first glance, it might seem that the design study is not connected to practices, but this cannot be any further from the truth. Theories such as General Design Theory[4] and Axiomatic Design[5] in the 1980s, Coupled Design Process[6] in the 1990s, Infused Design[7] and C–K Theory[8] in the 2000s have paved the way for the development of many practices and design tools. In addition, these theories contributed to innovative design and the advancements of other sciences such as artificial intelligence, set theory, and engineering sciences. We will not dive into details as it is beyond the scope of this thesis.

Design research has matured since those early days, giving rise to today's body of design theories, methods, and computer tools. However, it is important to notice the gaps in the field and address them by research. Armand Hatchuel and Yoram Reich devoted 5 years of research and concluded that design research is facing four main difficulties. They are related to the lack of unity, paradigm shifts, fragmentation of design professions and the limits of empirical research for conceptual design. Our work come in touch with the fourth difficulty – empirical research for conceptual design. One important design practice of interest is the Design Space Exploration (DSE). DSE refers to systematic analysis and pruning of design points based on requirements, constraints, and parameters of interest. Ideally, the output of such activity is listing and understanding feasible solution variants for achieving a specific function. We believe that introducing new rigorous tools that support navigation of design space can bring theory even closer to practice.

1.3.2 Conceptual design

Andreasen et al. define conceptualization as the process of creating something previously unknown or unseen: a new product[1]. It is the core synthesis activity of design. Hubka's Theory of Technical Systems (1973) was one of the earliest to address conceptual design. Conceptual design is the early phase of the design process, during which the designers articulate the general outlines of function and form of the system. In this phase, designers also define formal relations and functional interactions, processes, and strategies. A distinguishing character of the conceptual design phase is the high level of uncertainty and fuzziness. Even the most seasoned designers find it difficult to work in such environment. During conceptual design, designers rely mostly on experience and gut feeling to evaluate solution variants. Such approaches have been in action for ages. However, as systems become more complex, the design process becomes more challenging for designers. One complexity-related issue is the information that a designer must have to make optimal design decisions. There is a need to concentrate information and decrease the levels of fuzziness and uncertainty during early design. One way to tackle these challenges is to utilize the available information about the architecture to compare solution variants. This can provide designers with important insights and decrease the level of uncertainty and fuzziness.

1.3.3 System architecture

The field of system architecture is a deep-rooted field within the study of design. System architecture is concerned with the big picture of the "system". A system is thought of as a set of interacting elements arranged in such a way as to fulfill a specific purpose. The Open Group Architecture framework defines system architecture as "formal description of a system, or a detailed plan of the system at component level to guide its implementation"[9]. Crawley et al.

wrote in the definition of system architecture “the embodiment of a concept: the allocation of physical/informational function to elements of form and definition of interfaces among them and with the surrounding context” [10]. To simplify, we will think of architecture as the set of decisions that define a system at its highest level of design. The rigorous framing of an "architecture" has made it possible to broaden the research to include new areas such as digitalizing many aspects of the design process and Model-Based Systems Engineering.

As we noted before, the space that has solution variants as points in it is known as the design space. Designers must navigate through the design space to look for the optimal variants. This is a complicated task that requires multidisciplinary knowledge, expertise. A review of research on multidisciplinary design optimization can be found in the work of de Weck et al.[11]. We will focus on the fact that optimal solution variants come from choosing the optimal architectures. Comparing the architectures of solution variants can give powerful insights.

1.3.4 Solution variants evaluation

Designers work to develop optimal systems that meet needs at the lowest costs. But how?... The analysis and evaluation of solution variants is the key. Many evaluation methods can be found in the literature. We will briefly discuss two of them here.

The first method was proposed by Pahl et al. in his book "Engineering Design: A Systematic Approach"[12]. This method combines two types of analysis: cost-benefit-analysis and technical and economic evaluation analysis. It starts by establishing a tree network of technical and economic criteria arranged on levels of dependence. The criteria are usually derived from the requirements, constraints and desired parameters. Next, a weight is assigned to each criterion. Then, for each solution variant, each criterion is given a value (by the designer). The aggregation of the criteria values in the tree is calculated to reach a single value for each solution variant. As a result, a pairwise comparison between resulted values can be made to reach a conclusion about the best solution variant. This method is widely used but it has a major drawback. The subjectivity element in assigning weights and evaluating individual criterions can significantly affect the output. There is no guarantee of reproducibility for the results if the same evaluation is done by another designer.

The second method was presented in several works of Vladu and Dobre [13], [14]. This method is somewhat similar to Pahl's evaluation method. The main difference is that Vladu implemented fuzzy logic[15] to assign weights and evaluate criterions (we will discuss the application of fuzzy logic in more details in chapters 2 and 3). Although this method eliminates the subjectivity from assigning a weight, the results can be affected by the choice of the membership functions and defuzzification method.

Overall, the reviewed methods do not distinguish the different aspects of a system (i.e., types of features) and are used for general evaluation. Another option for making such comparisons is quantifying the similarity/distance between architectures. Using similarity/distance measures for solution variants evaluation will be a valuable addition to the existing set of tools.

1.3.5 Similarity

In this work, we will explore how similarity can be used for evaluation. For that, a review of the literature on similarity is presented here. A similarity function is a real-valued function that quantifies the similarity between two objects [16]. Although no single definition of a similarity measure exists, similarity measures can be thought of as the inverse of distance metrics. They take on large values for similar objects and either zero or a negative value for very dissimilar objects. For example, if two pieces of data have close x, y coordinates, then their "similarity" score, the likelihood that they are similar, will be much higher than two data points with more space between them. The literature on similarity comes from various fields. In his work [17], Henkel et al discussed the notion of similarity for system biology models. They argued that systems biology models are rapidly increasing in complexity, size, and numbers. This brings up the need for software tools for the retrieval, comparison, combination, and merging of models. Henkel et al. argued that such tools need to be able to quantify the differences and similarities between computational models. This is where the similarity measure comes handy. The analogy between systems biology and engineered systems is clear and it calls for further inspection.

Many works have surveyed similarity measures. Elavarasi et al. describe the basics of the semantic similarity measure, classification of single ontology-based similarity measure and cross ontology-based similarity measure[18]. The main application of semantic similarity measures is web mining and text mining which needs semantic content matching. Choi et al. provided a survey of binary similarity and distance measures in their work "A Survey of Binary Similarity and Distance Measures"[19]. In this work, they listed 76 binary similarity and distance measures used over the last century and discussed their correlations through the hierarchical clustering technique.

The binary feature vector is the most common representations for patterns. Similarity and distance measures play a critical role in identifying patterns by clustering or classification.

Introducing similarity to the design research can open the door for new applications. The most obvious of which is clustering similar solution variants in design space and creating a design taxonomy. In the next chapter, we elaborate on binary similarity measure to use it later in the proposed framework.

1.3.6 Applications of similarity measures

Similarity measures have an important influence on the development of multiple fields of science. In this section, we will explore the applications of similarity measures in two fields, namely: biology and information science. We choose these field to present examples of similarity measures applications because their systems have common characteristics with engineering systems. For instance, systems biology models are rapidly increasing in complexity, size and numbers. The same is happening with engineering systems and models.

1.3.6.1 Biology

Biology in general and bioinformatics in part have seen many applications of similarity measures. Some of these applications are rooted deep in the field. We will briefly discuss genetic distance and sequence alignment as two examples that show the importance of similarity.

- Genetic distance

One well-developed branch of bioinformatics is the study of genetics for species. Genetic distance is a measure of the genetic divergence between species or between populations within a species, whether the distance measures time from a common ancestor or degree of differentiation.[20] Populations with many similar alleles have small genetic distances. This indicates that they are closely related and have a recent common ancestor.

Let X and Y represent two different populations for which L loci have been studied. Let X_u represent the u_{th} allele [1]at the l_{th} locus.

- Cavalli-Sforza chord distance

In 1967 Luigi Luca Cavalli-Sforza and A. W. F. Edwards published this measure. It assumes that genetic differences arise due to genetic drift only. One major advantage of this measure is that the populations are represented in a hypersphere, the scale of which is one unit per gene substitution. The chord distance D_{CH} in the hyperdimensional sphere is given by [21], [22]

$$D_{CH} = \frac{2}{\pi}\sqrt{2(1 - \sum_{l}\sum_{u}\sqrt{X_u Y_u})} \qquad (1)$$

Some authors drop the factor $\frac{2}{\pi}$ to simplify the formula at the cost of losing the property that the scale is one unit per gene substitution.

- Reynolds, Weir, and Cockerham's genetic distance

In 1983, this measure was published by John Reynolds, Bruce Weir, and C. Clark Cockerham. This measure assumes that genetic differentiation occurs only by genetic drift without mutations. it estimates the coancestry coefficient θ which provides a measure of the genetic divergence by:[23]

$$\Theta_{\omega} = \sqrt{\frac{\sum_l \sum_u ((X_u - Y_u)^2)}{2\sum_l (1 - \sum_u X_u Y_u)}} \qquad (2)$$

- Use in sequence alignment

Similarity matrices are also used in sequence alignment. As in binary similarity measures, higher scores are given to more-similar characters and lower or negative scores for dissimilar characters.

[1] An allele is a variant form of a given gene

Nucleotide similarity matrices are used to align nucleic acid sequences. Because there are only four nucleotides commonly found in DNA (Adenine (A), Cytosine (C), Guanine (G) and Thymine (T)), nucleotide similarity matrices are much simpler than protein similarity matrices. For example, a simple matrix will assign identical bases a score of +1 and non-identical bases a score of −1.

A more complicated matrix would give a higher score to transitions (changes from a pyrimidine such as C or T to another pyrimidine, or from a purine such as A or G to another purine) than to transversions (from a pyrimidine to a purine or vice versa). The match/mismatch ratio of the matrix sets the target evolutionary distance.[24] The +1/−3 DNA matrix used by BLASTN is best suited for finding matches between sequences that are 99% identical; a +1/−1 (or +4/−4) matrix is much more suited to sequences with about 70% similarity. Matrices for lower similarity sequences require longer sequence alignments.

Amino acid similarity matrices are more complicated because there are 20 amino acids coded for by the genetic code, and so a larger number of possible substitutions. Therefore, the similarity matrix for amino acids contains 400 entries (although it is usually symmetric). The first approach scored all amino acid changes equally. A later refinement was to determine amino acid similarities based on how many base changes were required to change a codon to code for that amino acid. This model is better, but it doesn't consider the selective pressure of amino acid changes. Better models considered the chemical properties of amino acids.

One approach has been to empirically generate the similarity matrices. The Dayhoff method used phylogenetic trees and sequences taken from species on the tree. This approach has given rise to the PAM series of matrices. PAM matrices are labeled based on how many nucleotide changes have occurred, per 100 amino acids. While the PAM matrices benefit from having a well understood evolutionary model, they are most useful at short evolutionary distances (PAM10–PAM120). At long evolutionary distances, for example, PAM250 or 20% identity, it has been shown that the BLOSUM matrices are much more effective.

The BLOSUM series were generated by comparing several divergent sequences. The BLOSUM series are labeled based on how much entropy remains unmutated between all sequences, so a lower BLOSUM number corresponds to a higher PAM number.

1.3.6.2 Information science – edit distance

Edit distance is a way of quantifying how dissimilar two strings (e.g., words) are to one another by counting the minimum number of operations required to transform one string into the other. Edit distances find applications in natural language processing, where automatic spelling correction can determine candidate corrections for a misspelled word by selecting words from a dictionary that have a low distance to the word in question.

Given two strings a and b on an alphabet Σ (e.g. the set of ASCII characters, the set of bytes [0..255], etc.), the edit distance $d(a, b)$ is the minimum-weight series of edit operations that transforms a into b. One of the simplest sets of edit operations is that defined by Levenshtein in 1966.[25]

Insertion of a single symbol. If $a = uv$, then inserting the symbol x produces uxv. This can also be denoted $\varepsilon \rightarrow x$, using ε to denote the empty string. Deletion of a single symbol changes uxv to uv $(x \rightarrow \varepsilon)$. Substitution of a single symbol x for a symbol y ≠ x changes uxv to uyv $(x \rightarrow y)$.

In Levenshtein's original definition, each of these operations has unit cost (except that substitution of a character by itself has zero cost), so the Levenshtein distance is equal to the minimum number of operations required to transform a to b. A more general definition associates non-negative weight functions $w_{ins}(x)$, $w_{del}(x) and\ w_{sub}(x, y)$ with the operations.[25]

Additional primitive operations have been suggested. A common mistake when typing text is a transposition of two adjacent characters, formally characterized by an operation that changes $uxyv$ into $uyxv$. For the task of correcting OCR output, merge and split operations have been used which replace a single character into a pair of them or vice versa.[26]

Other variants of edit distance are obtained by restricting the set of operations. Longest common subsequence (LCS) distance is edit distance with insertion and deletion as the only two edit operations, both at unit cost.[27] Similarly, by only allowing substitutions (again at unit cost), Hamming distance is obtained; this must be restricted to equal-length strings.[27] Jaro–Winkler distance can be obtained from an edit distance where only transpositions are allowed.

In conclusion for the section about the applications of similarity measures, we believe that introducing similarity to design will bring useful applications to the design theory and practice.

1.4 Specific thesis objective

Gaps were identified in the literature on evaluation methods during conceptual design. Given these gaps and the overarching thesis objective presented in Section 1.2. This has led to the development of the specific thesis objective, which is to support solution variants evaluation in the design space during conceptual design by:

1- Identifying the different aspects of a system and the features that belong to each aspect

2- Developing a framework for comparing solution variants architectures and quantifying the similarity between them

3- Demonstrating the possible application of such a framework and how it can be integrated with existing evaluation methods.

1.5 Research methodology

To explore the possibility of using similarity to compare architectures of solution variants, we relied on qualitative and quantitative methods to collect data, develop tools of the framework and verify the results. This work relied mainly on two methods:

1.5.1 Case study

After careful consideration of the literature, we started studying various systems with different levels of complexity. We limited the study to engineered systems for two reasons. First, it is easier to study tangible systems and generalize to cover other systems (such as services). Second, the availability of a good database for commercial aircraft. The commercial aircraft database was collected from various sources by Demetrious Kellari (MIT SAL) in his master thesis research[28]. The case study also involved studying the architectures of various systems from different domains. For that, we listed the main features and observed the repeated patterns and commonalities that later led to identifying three distinguished aspects of a system in the conceptual design phase. We also reviewed the literature on comparison methods from non-engineering fields to look for methods that can be adapted to comparing architectures of engineered systems. After eliminating inapplicable methods, we tested various and assessed their applicability and consistency throughout the range of systems that we studied. Finally, the resulted comparison tool was applied to the example of commercial aircraft as a case study and proof of concept work.

1.5.2 Survey study

The survey study was a critical part of this work as it helped in shaping the research question and tackling the right issues. When we started doing this research, we had a very abstract idea about similarity in design. Neither the problem nor applications were known for sure. This idea was studied and improved further to become the seed of this work. This was done thanks to tons of discussions, comments, feedback (direct and indirect). The major theme of this part of the work can be identified as a survey study. Below we elaborate on two aspects of the survey study, namely interviews and questionnaire.

a. Interviews

In the early stage of this work, we carried out multiple interviews and discussions with experts from the field of system architecture and design as well as graduate students with engineering background and expertise. This helped to shape the direction of the research, articulate the problem and identify the possible applications. This later was continued in a form of feedback on the framework and the intermediate conclusions through the research time. Interviews were also the base for creating the questionnaire as we relied on direct feedback during the design stage of the survey. More on the questionnaire below.

b. Questionnaire

Once the research started taking its shape, we conducted a survey study to get expertise from a wider range of audience. The need for the questionnaire raised when we started studying which parameters are most important to consider for evaluation. In addition, the data from participants could help in verifying system aspects that we identified as well as the comparison methods.

The questionnaire, in addition to instructions, consisted of three types of questions. The types are one-answer multiple-choice questions (MCQ) multiple-answer MCQ and direct (text) questions. We used standardized scales (Likert scale), question formatting and strategies to eliminate biases and avoid common mistakes. The majority of the population of this survey are affiliated with the MIT SDM program and the Skoltech Space Center. This group has background and expertise in engineering and design.

The information we received from the questionnaire was very valuable to the development of the framework. Chapter 4 detail the survey study and the questionnaire in part.

1.6 Thesis overview

The rest of this thesis is organized as follows. In chapter 2, we distinguish three main aspects of a system -architecture, crisp features, and relative features. Next, we explore the notion of similarity and show that similarity measure can be used for engineered systems. This was done with an example of antenna release mechanisms (ARMs). Then, we review two existing evaluation methods that rely on crisp and relative features. In chapter 3, we present and detail the proposed framework for solution variants evaluation by comparing their architectures. Chapter 4 covers the survey study and the discussion of its results. In chapter 5, we demonstrate the application of the framework with a case study on commercial aircraft. Finally, we conclude with the summary and recommendations in chapter 6.

CHAPTER 2:
SIMILARITY MEASURES AND SOLUTION VARIANTS EVALUATION

In chapter 1, we showed that similarity is of great importance to several fields of study and have many practical implementations. Introducing the notion of similarity to design will improve the evaluation by introducing information about the similarity between architectures. This chapter starts by distinguishing three aspects of a system – architecture, crisp features, and relative features. Next, it explores the existing similarity and distance measures. This will serve as a background for the framework. We will show such measures can be used to compare the architectures of solution variants in the design space.

Ultimately, we expect one of three outputs. First, find similarity measures that can be applied for comparison of different engineered systems. Second, show that none of the existing similarity measures can be used as a tool for comparison. Third, show that we can combine some measures and/or make changes to make them applicable to engineered systems. In the first case, we will show how we can apply these measures in the case of engineered systems. We will need to propose a new measure in the second case. In the last case, we will show how the existing measures can our goal of comparing architectures.

In the rest of this chapter, a distinction between architecture, crisp features, and relative features are made in 2.1. In 2.2, we describe the notion of similarity and how it can be captured mathematically. A problem example is presented in 2.3 and is used throughout the chapter. In 2.4, we introduce the operational taxonomic units as the start point for calculating similarity. Then, few binary measures were reviewed and compared. After that, we showed how the system architecture can be represented as a binary vector of architectural decisions. Having the binary vector, we show how binary similarity/ distance can be calculated. Before concluding the chapter in 2.6, we discuss two evaluation methods. The first is Pahl's evaluation method and the second is the application of fuzzy logic.

2.1 Aspects of engineered systems

In this work, three aspects of a system are distinguished. The first aspect is the architecture. Architecture *is" the embodiment of concept, the allocation of function to elements of form, and definition of relationships among the elements and with the surrounding context.*[10]*"*. The second aspect is the crisp features. Under crisp features, we understand features that have a specific real value that can be measured with a measurement tool. The value of such features does not rely on the opinion of the designer. Examples of such features are the range of an airplane, elasticity of a liquid and so on. The third aspect is concerned with another category of features – relative features. We define relative features as features that cannot be measured, or their measurement is dependent on the designer's opinion. Such features include the aesthetic aspect of a system, estimations of crisp features under uncertainty (no accurate measurement values are available).

The distinction between these three aspects is of great importance to this thesis. Although the literature does not state clearly the proposed distinction, our literature review showed that the existing evaluation methods work on the latter two aspects – crisp and relative features.

The benefit of considering the architecture comes from the fact that during conceptual design little information is known about crisp and relative features. Nevertheless, we will show how information about crisp and relative aspects can be utilized to bring a universal value to the evaluation of solution variants.

2.2 The mathematical notion of similarity

We have established that similarity has great importance in many fields. For instance, there are geometric methods in mathematics for evaluating similarity that are used in trigonometry, congruence, and homothety. In biology, different similarity measures are used to measure genetic variation over populations. Another important problem in molecular biology is assessing the similarity between the sequences of proteins. Measures of similarity can also be found in the fuzzy set theory, which has applications in medicine, management, and meteorology. Semantics is another field with numerous measures of similarity where topological among other methods are applied. In addition, graph theory is one of many methods used for assessing similarities in taxonomy. The concept of similarity also has a fundamentally crucial role in the study of psychology. [16] For instance, in some psychological experiments, participants are asked to assess the similarity of pairs of objects. Similarity also plays an important role in the modeling of various psychological tasks in theories of the recognition, identification, and objects categorization. Generally, these theories assume that more similar a pair of objects are, the more likely they are to be confused with one another. A review or even a listing of similarity measures and all their applications is almost impossible. Instead, here we will focus on a few measures and explain them with an example.

How similarity is captured?

Similarity is usually represented and applied using similarity measures. Similarity measures are represented by mathematical functions. A similarity measure assigns a value calculated by a mathematical function to a pair of objects under comparison. If we represent the properties, of the objects that we compare, in a property space, then similarity can be thought as the inverse distance. In other words, objects with small distances are considered more similar (close) than objects with larger distances (far)[29]. It is a common practice to normalize the similarity value (say "σ") to yield values between 0 and 1. In this case, a similarity $\sigma = 1$ implies that two objects are identical (i.e., indistinguishable) with regard to the properties considered, while entirely different objects will have a similarity $\sigma = 0$. Formally, a normalized similarity measure for a set X is defined as a function which assigns two objects $x_1, x_2 \in X$ a value $\sigma \in [0,1]$. This σ is called the 'similarity value'. Most similarity measures are symmetrical and satisfy the triangle inequality[29].

As it is difficult to consider all the aspects of an object (system), similarity measures are usually evaluated for specific aspects of the objects of interest. The idea is to choose the main aspects that you want to compare for the objects and map the real objects to their models (see Figure 1).

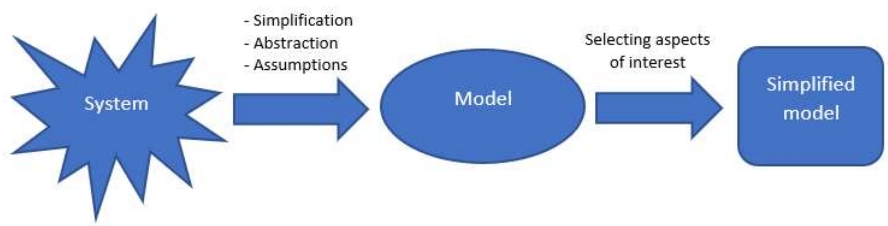


Figure 1: creating models to compare systems

2.3 Example problem

To demonstrate the work of similarity measures, for the rest of the chapter we will use an illustrative example with antenna release mechanisms (ARMs) concepts for a CubeSat. ARM is a mechanism used to deploy the antenna after the CubeSat is deployed to orbit. ARM also stores and protect the antennas during launch. After the satellite is deployed and put to orbit, the ARM releases the antenna and they become ready to send and receive signals. The need for such a mechanism comes from the space limitation in launchers as antennas can take much space.

CubeSats have been gaining popularity in the past two decades and the need for ARM is driving engineers to come up with new architectures all the time. In general, ARM has many functions but for simplicity, we will focus on three functions. First, store the antennas during transportation and launch. Second, release the antennas once in orbit. Finally, fix the antennas' position to guarantee to send and to receive signals.

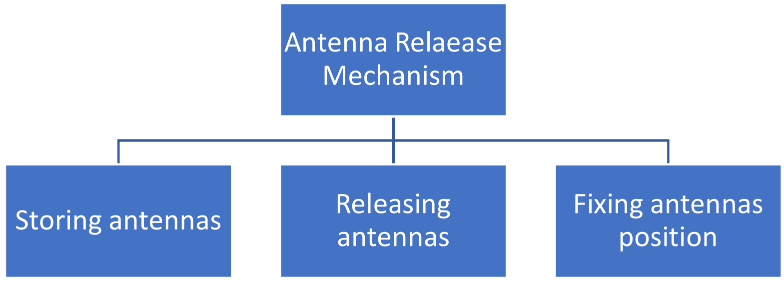


Figure 2- Functional decomposition for ARM

2.4 Binary similarity

The class of problems that binary similarity measure can help solve is wide. For instance, binary similarity measures are at their core of many methods that are used to solve pattern analysis problems, such as classification and clustering. The performance of these methods relies mainly on the similarity measure used to solve the problem. Over the decades, researchers have taken elaborate efforts to develop similarity measures. Many of these measures are being actively used in various fields.

For example, the Jaccard similarity measure was used for clustering ecological species [30], and Forbes proposed a coefficient for clustering ecologically related species[31], [32]. The binary similarity measures were subsequently applied in chemistry [33], image retrieval[34], geology[32], ethnology[35], taxonomy[36], and biology [37], [38]. Recently, some binary similarity measures have been actively used to solve the identification problems in biometrics such as fingerprint[39], iris images[40], and handwritten character recognition [41]. Many papers [42]–[45] discuss their properties and features.

2.4.1 Operational Taxonomic Units

To illustrate the work of binary similarity measures, let's assume that two objects, x and y are represented by binary feature vector form. Let n be the number of features (i.e., dimension of the feature vector). Operational Taxonomic Units **(OTU)** (Table 1)[46] is a popular way to express binary similarity measures is in a 2×2 contingency table where $\boldsymbol{a}$ is the number of features where the values of x and y are both 1 (or presence), meaning '*positive matches'*, $\boldsymbol{b}$ is the number of attributes where the value of x and y is (0,1), meaning 'x *absence mismatch*es', $\boldsymbol{c}$ is the number of attributes where the value of x and y is (1,0), meaning 'y *absence mismatches*', and $\boldsymbol{d}$ is the number of attributes where both x and y have 0 (or absence), meaning '*negative matches*'. The diagonal sum $a + d$ represents the total number of matches between x and y, the other diagonal sum $b + c$ represents the total number of mismatches between x and y. The total sum of the 2 × 2 table, $a + b + c + d$ is always equal to n.

Table 1. OTUs expression of binary instances x and y

Object/system/product	$\boldsymbol{x}$	$\bar{\boldsymbol{x}}$	**Sum**
$\boldsymbol{y}$	$a = x * y$	$b = \bar{x} * y$	$a + b$
$\bar{\boldsymbol{y}}$	$c = x * \bar{y}$	$d = \bar{x} * \bar{y}$	$c + d$
Sum	$a + c$	$b + d$	$n = a + b + c + d$

2.4.2 Binary similarity measures

In See Appendix A (**Table** and **Table**) we collected some binary similarity and distance measures used over the last century. Here, we will only address a few of these measures that are more relevant to our research.

Next, we elaborate on some of the binary similarity measures. Namely, Jaccard similarity coefficient, Jaccard distance, Dice coefficient, Faith coefficient, Simpson coefficient, Braun & Banquet coefficient, and Goodman similarity coefficient. Then, we will apply these measures to our example problem.

- **Jaccard similarity coefficient**

Jaccard coefficient is a measure used for comparing the similarity between objects. The Jaccard coefficient is defined as the size of the intersection divided by the size of the union of the sample sets. i.e., number of common features divided by the number of all the features the objects we compare have :

$$S_{Jaccard} = \frac{a}{n - d} = \frac{a}{(a + b + c)} \tag{3}$$

- **Jaccard distance**

Jaccard distance measures dissimilarity between objects is complementary to the Jaccard coefficient and is obtained by subtracting the Jaccard coefficient from 1, or, equivalently, by dividing the difference of the sizes of the union and the intersection of two sets by the size of the union

$$D_{Jaccard} = 1 - \frac{a}{n-d} = \frac{b+c}{(a+b+c)} \tag{4}$$

- **Dice similarity coefficient (DSC)**

It is different from the Jaccard index which only counts true positives once in both the numerator and denominator. DSC is the quotient of similarity and ranges between 0 and 1.[47] It can be viewed as a similarity measure over sets.

$$S_{Dice} = \frac{2a}{2a+b+c} \tag{5}$$

- **Faith coefficient**

Faith included the negative match but only gave the half credits while giving the full credits for the positive matches in the equation :

$$S_{Fati} = \frac{a+0.5d}{a+b+c+d} \tag{6}$$

- **Simpson coefficient**

This index is independent of the number of entries on the larger of two lists to be compared. If the smaller list has k percent of its entries appearing also in the larger list, then the value of the k/100 no matter how extreme the difference in list sizes may be[48].

$$S_{Simpson} = \frac{a}{\min(a+b, a+c)} \tag{7}$$

- **Braun & Banquet coefficient**

Similar to Jaccard similarity measure, Braun & Banquet similarity measures take the value 0 when x and y have no common elements, and the value 1 when x and y are equal. However, in intermediate cases $S_{Braun\&Banquet} \geq S_{Jaccard}$[49].

$$S_{Braun\&Banquet} = \frac{a}{\max(a+b, a+c)} \tag{8}$$

2.4.3 Architectural decisions as binary features

To deliver a specific function, elements of form have to be assigned for each function. This process can be considered as decision-making to assign form to function. Each function corresponds to one or more architectural decisions. For each architectural decision, there can be one or more options (alternatives). In Table 2, we illustrate the relationship between functions, architectural decisions, and options. The form results can be identified after the decision-making process. I.e., the form can be identified by the set of the architectural decision taken to fulfill the functions. For example, the grey shaded area in Table 2 represents a set of architectural decisions for one solution variant.

Table 2. Representing functions, architectural decisions, and options using Pugh's matrix

Architectural decisions	**option 1**	**option 2**	**option …**
	Function 1		
Arch. decision 1	Option 1	Option 2	…
Arch. decision 2	Option 1	Option 2	…
…	…	…	…
	Function 2		
Arch. decision 1	Option 1	Option 2	…
…	…	…	…
	Function …		
Arch. decision 1	Option 1	Option 2	…
…	…	…	…

Binary similarity measures are only applicable to binary features. In other words, features that can only either exist or not exist. In our case, the options of architectural decisions can be considered as binary features. This is due to the fact that these features exist in one of two states, true or false (1 or 0).

2.4.4 Application of binary similarity measure

Applying binary similarity measure to a pair of architectures will give us a value between 0 and 1 that shows how similar these architectures are to each other. If we calculate the distance ($distance\ = 1 - similarity$), then we can think of architectural distance as the architectural cost of converting one architecture into the other architecture.

We will demonstrate the notion of architectural distance with an example. Table 3[2] lists some functions, corresponding architectural decisions, and options for antenna release mechanism for a CubeSat. For simplicity, we only listed three functions, seven architectural decisions in total and no more than three options for each decision. We also will not assign weights for functions at this point. Instead, we will treat all functions and options as equally important.

Table 3. Architectural decisions for ARM

Architectural decisions	**Option 1**	**Option 2**	**Option 3**
Function 1: Storing antenna			
Antenna storing location	inside	outside	
Antenna type	foldable	rigid	
Number of doors	0	2	4
Function 2: Releasing antenna			
technique	melting a thread	magnetic force	
Function 3: Fixing antennas position			
technique	melting rope	magnetic field	
Other			
Max number of monopole antennas	2	3	4
Antenna deployment indicator	yes	No	

Now that we have chosen the architectural decisions, we will use to compare the ARMs, we can present the corresponding decisions for ARMs. For example, the ARM designed by GOMSpace has the following set of architectural decisions:

- Antenna storing locations is outside
- The antenna is rigid

[2] This table is not intended to be an exhaustive list of the functions, architectural decisions or options. We only use this example for illustration purposes.

- No doors
- Releasing the antenna by melting a thread
- Can be mounted on the top/bottom of the CubeSat
- The maximum number of monopole antennas is 4
- No deployment indicator

We list the architectural decisions for four ARMs in Table 4

Table 4. Architectural decisions for 4 chosen CubeSat ARMs

ARM	GOM Space[50]	Aalto 1[51]	GOS	ISIS[52]
Function 1: Storing antenna				
Antenna storing location	2	1	1	1
Antenna type	2	1	1	1
Number of doors	1	2	2	3
Function 2: Releasing antenna				
Technique	2	2	3	2
Function 3: Fixing antennas position				
Mounting location	2	2	3	2
Other				
Max number of monopole antennas	3	3	3	3
Antenna deployment indicator	2	1	1	2

It is important to note that we only considered the binary features of the ARM, namely, the architectural decisions. This is because binary similarity measures can only be applied to binary features. For instance, it is of no value to apply binary similarity measures to performance or any metric feature as they are expressed in values that do not in a manner of 1 or 0 (i.e., exists or does not exist).

Returning back to the example, let's assign the notation x to the GOS ARM and y to the Aalto ARM. The OTUs of binary instances for the pair "GOS and Aalto" can be seen in Table 5.

Table 5. OTUs for ARMs, where x-GOS; y-Aalto ARMs

Object/system/product	$\boldsymbol{x}$	$\boldsymbol{x}$	Sum
$\boldsymbol{y}$	$\boldsymbol{a = 7}$	$\boldsymbol{b = 1}$	$\boldsymbol{a + b = 8}$
$\boldsymbol{\bar{y}}$	$\boldsymbol{c = 1}$	$\boldsymbol{d = 9}$	$\boldsymbol{c + d = 10}$
Sum	$\boldsymbol{a + c = 8}$	$\boldsymbol{b + d = 10}$	$\boldsymbol{n = 18}$

Table 6 shows the results of using five binary similarity measures to compare two of the ARMs from Table 4 the GOS ARM and the Aalto ARM. We can see that the results depend on the choice of the measure. However, the change in the results is mostly in the magnitude. Next, in our work, we will use Jaccard similarity measure as it is simpler to implement and does not account for negative agreements(see section 2.3.2).

Table 6. Applying similarity measures to compare ARMs

Similarity measure	GOS ARM and Aalto ARM
Jaccard similarity coefficient	0.63
Dice coefficient	0.77
Faith coefficient	0.56
Simpson coefficient	0.83
Braun & Banquet coefficient	0.71

In Figure 3 we can see the branching diagram that uses the neighbor-joining algorithm to cluster the objects by their relative distance. Each branch (leaf) represents architecture and each node represent the common root for the architectures under share. The path length equals the calculated distance between architectures (**Remember! distance = 1 - similarity**). We used the Jaccard distance coefficient to compare ARMs. We can notice that the path length between the Aalto 1 ARM and GOS is 0.22 which is also the Jaccard distance between them.

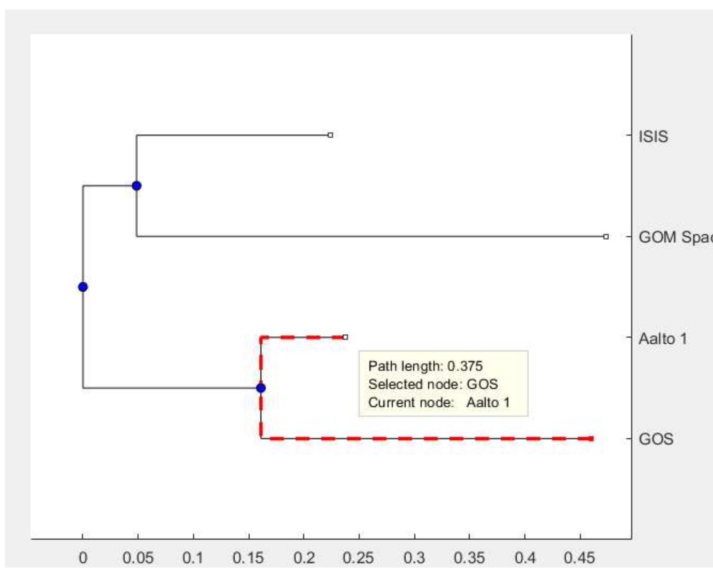


Figure 3. Branching diagram showing the distance between the ARMs

Binary similarity measures are easy to apply and can be used directly to compare the architectural decisions of engineered systems. However, most of these measures treat features equally. In other words, binary similarity can be used only for binary features. This is applicable to architectural decisions.

2.5 Cross-field analysis of evaluation methods

Evaluation of simple systems with few components and features is a relatively easy task in comparison to the real system with high complexity. The evaluation task becomes exponentially more difficult as more components and features are introduced. The most popular evaluation method was proposed by Pahl et al. in his book: "Engineering Design - A Systematic Approach" [12]. This method combines two types of approaches: cost-benefit-analysis and technical and economic evaluation. The main drawback of Pahl's method is its subjectivity as it depends on estimation from the designer. Addressing the subjectivity of this method was the question of many research papers throughout the years. Vladu et al.[14] proposed using fuzzy logic to reduce subjectivity in their 2013 paper " Application of the fuzzy logic in the evaluation of solution variants in the product development process ". They also presented a case study for an application on Mechanical transmission[13]. The existing literature on the evaluation of solution variants was briefly discussed in chapter 1. Here we will detail the use of these two methods. This is done in order to show later how our framework can be integrated with existing evaluation methods easily.

2.5.1 Pahl's Evaluation method

The method consists in the establishment of a tree network of technical and economic criteria arranged on levels of dependence and which are mainly derived from the requirements list and from general constraints; such a tree network with four levels of dependency is presented in Figure 4. Criterion 1 from the first level is detailed using the criteria 11 and 12 from level 2, each of those two criteria is further detailed until the criteria from the last level are established. Thus, the criteria are arranged into levels of decreasing complexity. The criteria with the lowest complexity form the evaluation criteria used for the assessment of the solutions. "For the quantitative appreciation of each criterion are used two weighting factors. A weighting factor is a real, positive number ranging on a scale from 0 to 1 (or the scale can be chosen from 0 to 100). The two factors are the node weighting factor, n, which indicate the relative contribution of the criterion to the associated sub-group with respect to the superior criterion and the level weighting factor, l, which indicates the relative importance of a criterion at a particular level with respect to the criterion from the first level (for example, Criterion 1 in Figure 4). The node weighting factors are determinate by human experts. The sum of the node weighting factors of a sub-group at any level must always be equal to 1. For example, at the level 2 the sub-group associated to the criterion from the first level: $n11 + n12 = 0.8 + 0.2 = 1$; at the level 3 the first sub-group associated to the first criterion of the level 2: $n111 + n112 + n113 + n114 = 0.3 + 0.2 + 0.2 + 0.3 = 1$ and so on.

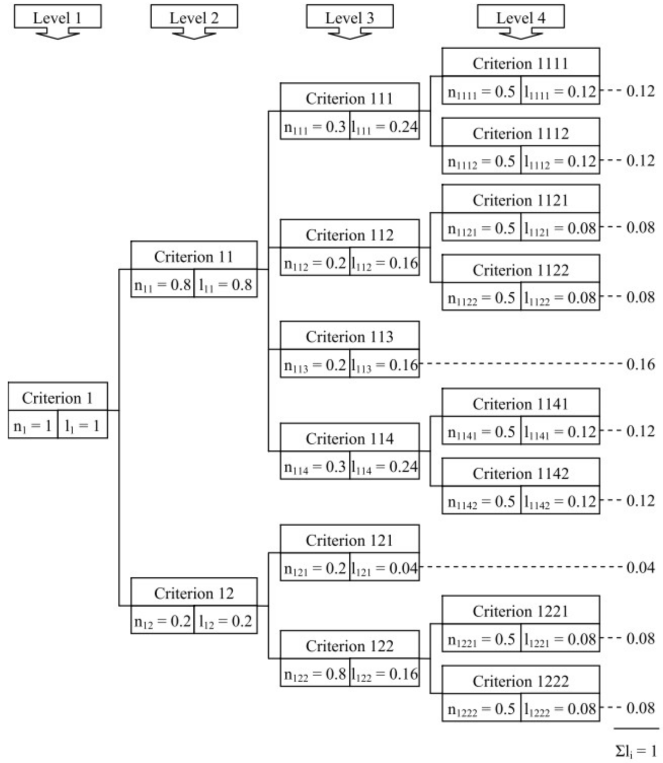


Figure 4. Example of tree network criteria with weighting factors[12]

The level weighting factor is determined by the product between the node weighting factor of the considered criterion and the level weighting factor of the superior associated criterion. The sum of the level weighting factors of all evaluation criteria (at the lowest levels) is equal to 1 ($\Sigma l_i = 1$). Thus, a percentage weighting can be attached to all the evaluation criteria.

The evaluation of the solution variants consists of:

1. analysis of each solution variant:
   a. for each evaluation criterion will be assigned a chosen assessing value according to a value scale, usually from 0 to 10 (Table 7) by the human experts;
   b. calculation of the final overall values (unweighted and weighted);
2. comparison of the solution variants. The solution variant with the highest overall values (unweighted and weighted) will be the optimal solution.

The unweighted overall value will be calculated with the equation:

$$v_\Sigma = \sum_{i=1}^{n} v_i \tag{9}$$

where: v_Σ - unweighted overall value of the solution variant;

v_i - chosen assessing value of the evaluation criterion i

The weighted overall value will be calculated with the equation:

$$(lv)_{\Sigma} = \sum_{i=1}^{n} l_i v_i \,, \qquad (10)$$

where $(lv)_{\Sigma}$ - weighted overall value of the solution variant;

l_i - level weighting factor of the evaluation criterion i

Table 7. Recommended assessing value scale for evaluation criteria[12]

Chosen assessing value	Signification	Chosen assessing value	Signification
0	Absolutely useless	6	Good with few drawbacks
1	Very inadequate	7	Good
2	Weak	8	Very good
3	Tolerable	9	Exceeding the requirement
4	Adequate	10	Ideal
5	Satisfactory		

In chapter 3 we will derive a general formulation of Pahl's method. This will be done to show how such method can be integrated into our framework.

2.5.2 Similarity for relative features using fuzzy logic

The fuzzy logic theory[53] has the ability to deal with the inexact and vague information that appears in the evaluation process like the uncertain assessing values and weighting factors because of the subjective character of this assessing.

Vladu et al. describe Pahl's approach to assessing the value of the node weighting factors as " imprecise, uncertain, inexact, ambiguous."[14] Thus, they propose an approach that calculates the overall values for each solution variant using fuzzy logic that will calculate assessing weighting values for each variant.

This means that the fussy logic will be applied to all sub-groups of criteria and for the associated superior criterion creating in this way a network tree of fuzzy logic systems which will be used to calculate the assessing weighting global value for each solution variant.

The procedure of fuzzy processing is realized in the following steps(see Figure 5): problem identification, fuzzification, rule base, defuzzification, and interpretation.

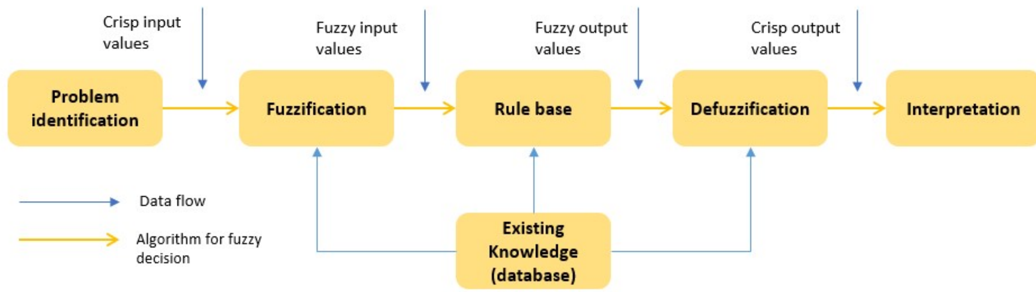


Figure 5. Fuzzy processing

Let's demonstrate the application of fuzzy logic for comparing objects by an example. We will use the example of ARMs. To create the fuzzy logic system for this example, we follow four important steps[14]:

a) The type of the fuzzy inference system (Mamdani type[54]) and the defuzzification method (centroid method);

Table 8. Fuzzy properties of ARMs

ARM	GOM Space	Aalto 1	GOS	ISIS
Fuzzy properties				
Easy to test	2	3	2	3
Easy to mount	1	4	4	2

b) The input and output variables (two input variables and one output variable). The input variables are the values for two criteria: "easy to mount" and "easy to test";

c) The membership functions attached to each variable (all the input variables will have five triangular membership functions represented in Figure 6 and the output variable will have five triangular membership functions represented in Figure 7);

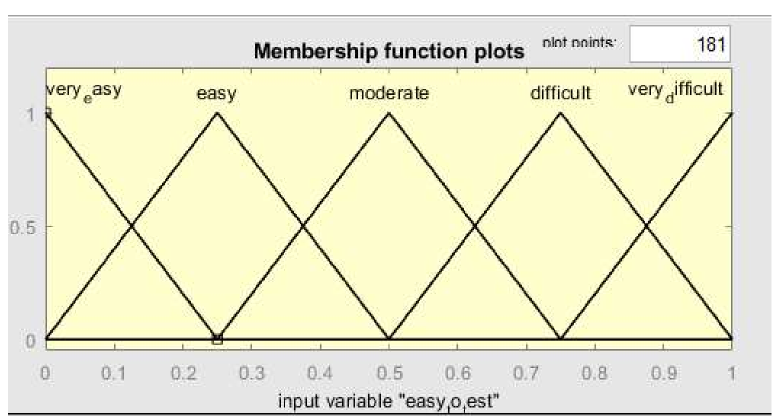


Figure 6. Representation of the five triangular membership functions for the input variables

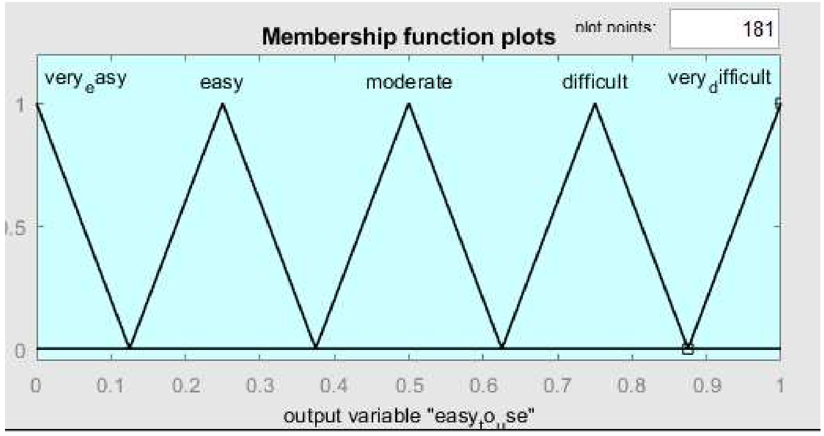


Figure 7. Representation of the five triangular membership functions for the output variables

d) The rule base consists of if-then sentences that the designer assigns according to the requirements and information available.

For the problem example, we have created few rules (See Figure 8 and Figure 9)so that easiness of testing is more important than easiness to mount because the ARM will be tested more times than it will be mounted.

1. If (easy_to_test is very_easy) and (easy_to_mount is very_easy) then (easy_to_use is very_easy) (1)
2. If (easy_to_test is easy) and (easy_to_mount is easy) then (easy_to_use is easy) (1)
3. If (easy_to_test is moderate) and (easy_to_mount is moderate) then (easy_to_use is moderate) (1)
4. If (easy_to_test is difficult) and (easy_to_mount is difficult) then (easy_to_use is difficult) (1)
5. If (easy_to_test is very_difficult) and (easy_to_mount is very_difficult) then (easy_to_use is very_difficult) (1)
6. If (easy_to_test is very_difficult) then (easy_to_use is difficult) (1)
7. If (easy_to_mount is very_difficult) then (easy_to_use is difficult) (1)
8. If (easy_to_test is very_easy) and (easy_to_mount is not very_difficult) then (easy_to_use is easy) (1)
9. If (easy_to_test is difficult) and (easy_to_mount is not very_difficult) then (easy_to_use is difficult) (1)
10. If (easy_to_test is easy) and (easy_to_mount is moderate) then (easy_to_use is easy) (1)
11. If (easy_to_test is very_easy) and (easy_to_mount is moderate) then (easy_to_use is easy) (1)
12. If (easy_to_test is difficult) then (easy_to_use is difficult) (1)
13. If (easy_to_test is very_difficult) then (easy_to_use is very_difficult) (1)

Figure 8. Rule base for ARM easiness of use

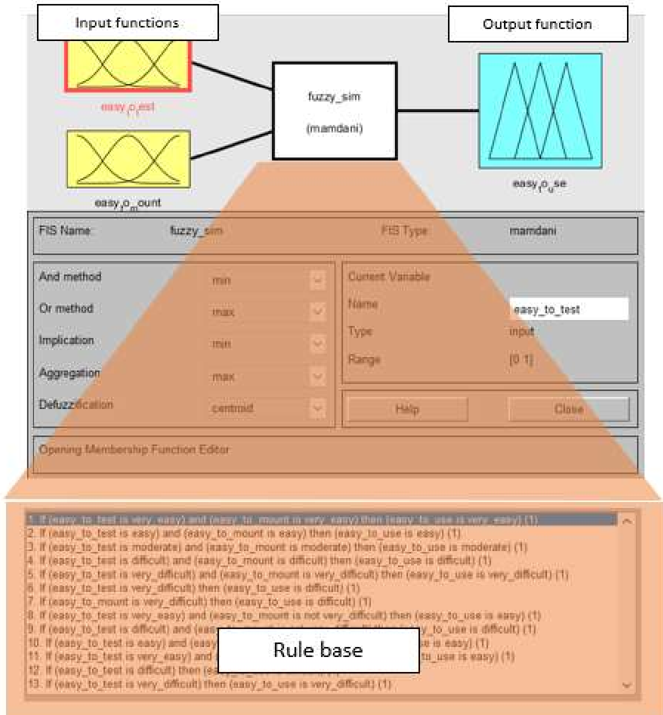


Figure 9. Rules as represented in MATLAB fuzzy logic toolbox

e) Finally, we apply the logic system to our example to define the easiness of use for each ARM individually first and then calculate the similarity between them. Table 8 shows two fuzzy features: easiness of testing and easiness of mounting. We will apply the logic system to both GOS ARM and Aalto.

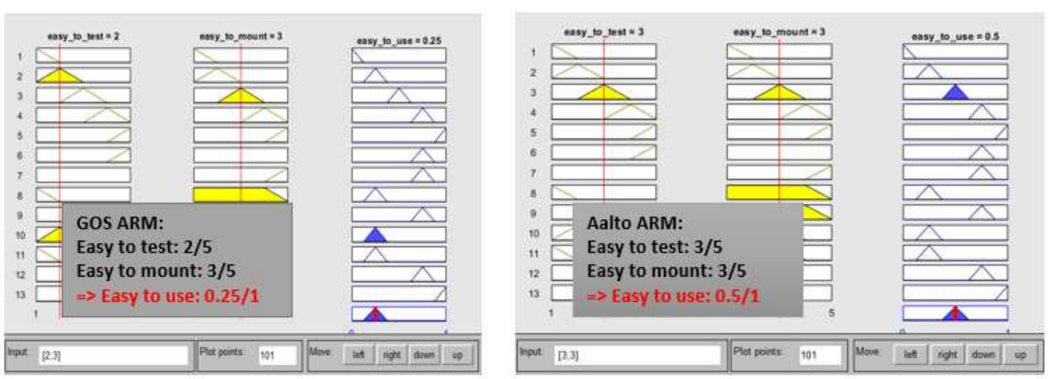


Figure 10. Fuzzy logic system applied to the ARM example (GOS and Aalto ARMs)

From the results above (See Figure 10) we note that Aalto ARM (0.5/1) is 0.25 (or 25%) more "difficult to use" than GOS ARM (0.25/1).

The approach above provides a less subjective approach to estimating a relative feature (easiness of use) of a system (ARM). This approach can be utilized to generate indicators that account for relative features.

2.6 Conclusion

This chapter started by distinguishing three aspects of a system. The aspects are the architecture, crisp features, and relative features. The rest of the thesis uses this to develop the framework with a focus on the architecture. Next, the work of binary similarity measures was explained along with two evaluation methods from the literature. We introduced the use of binary similarity methods to evaluate the dissimilarity between architectures. This led to the introduction of the term "architectural distance". Architectural distance will be the basis for the proposed framework. Finally, we detailed the work of Pahl's evaluation method and Vladu's evaluation method with the application of fuzzy logic.

CHAPTER 3:
THE PROPOSED FRAMEWORK

The literature review in chapter 1 helped to identify gaps in the research on evaluation methods during conceptual design. Existing evaluation methods rely on system parameters and characteristics (crisp and relative features). Such information usually is not available during conceptual design. The idea of introducing a new method to evaluate solution variants by comparing their architectures was proposed. The literature review revealed that similarity measures are of great importance in several fields. The notion of similarity in design was introduced and discussed in chapter 2. Furthermore, binary similarity measures were explained with a problem example. It was shown how architecture can be presented as a binary vector of architectural decisions. This allowed for the use of binary similarity measure for comparing architectures and getting a value of similarity. A normalized value of similarity $s \in [0,1]$ can be used to calculate distance $d \in [0,1]$. From there, the term "architectural distance" was introduced. Architectural distance is a value that shows how different two architectures are. This information adds important insights to the evaluation process. Chapter 2 also detailed the work of Pahl's evaluation method and Vladu's evaluation method. The two methods are used for evaluating solution variants but do not account for architecture. However, they come handy when some features of solution variants are known.

3.1 Introduction

A major development in the history of design in general and system architecture in part was the introduction of frameworks. "*A framework is a collection of views and viewpoints. A view is a model, diagram, table, picture or document that describes one or a few aspects of the system that are of relevance to one or more stakeholders, such as functional decomposition, concept of operations, or interfaces between the system components*"[55]. A viewpoint is the template and set of guidelines to construct a view. Generally, the purpose of frameworks is to improve the design process by providing a common approach across teams and codifying an institutionalizing best practice. In this chapter, we propose a framework for evaluating solution variants by comparing their architectures. The framework compares systems using aspects and features chosen by the user.

For the proposed framework, we make the following assumptions:

- Although the framework can be generalized, it is only presented for tangible engineered systems.

- The overall function (i.e., the delivered value) of pairs of architectures is the same. I.e., The framework does not compare architectures with different overall functions. For example, a car and a piano have different overall functions – transporting passengers and creating a melody.

This chapter starts by introducing some terms and detailing the framework step by step in section 3.2. Next, in section 3.3, we discuss some of the possible applications for such framework. Finally, we conclude with a short summary and discuss the limitations and recommendations in section 3.3.

NOTE: All the examples in this chapter are exclusively used for illustration.

3.2 Architectural distance – the framework

Before commencing to the framework, we will define some terms used in this chapter. The terms are architectural decision and similarity measure.

Architectural decisions are *"decisions about the main system architecture, the main entities of function and form, and their relationships*" [55]. Architectural decisions are also defined as "*the decisions that address architecturally significant requirements; they are perceived as hard to make and/or costly to change*"

A similarity measure or similarity function is "*a real-valued function that quantifies the similarity between two objects.*" Distance or dissimilarity is the opposite of similarity. If the output value s of a similarity measure between two objects is normalized between $s \in [0,1]$, where 0 means the two objects are totally different and 1 means totally similar or the same, then a distance between these objects will be: distance $d = 1 - s$. This means that two objects with a distance $d = 1$ are totally different and those with distance $d = 0$ are the same. This way it is easy to move between distance and similarity measures. For our proposed framework we will use normalized distance measure, where distance takes value $d \in [0,1]$.

The framework will be introduced as an algorithm of four steps. In step 1, the designer identifies the context (i.e., the big picture). This includes identifying the design space and the architectures he/she intend to evaluate. In step 2, the designer identifies the high-level functions for each architecture. In addition, he/she identifies the sets of architectural decisions for the two architectures they intend to compare. Step 3 is concerned with calculating the distance between the pair of architectures (architecture #1 and architecture #2). Finally, in step 4, the designer can utilize the results to make important decisions. He/she can consider architectural distance along with other indicators to evaluate the pair of solution variants. The overall outline of the proposed framework can be explained in the following steps(see Figure 11):

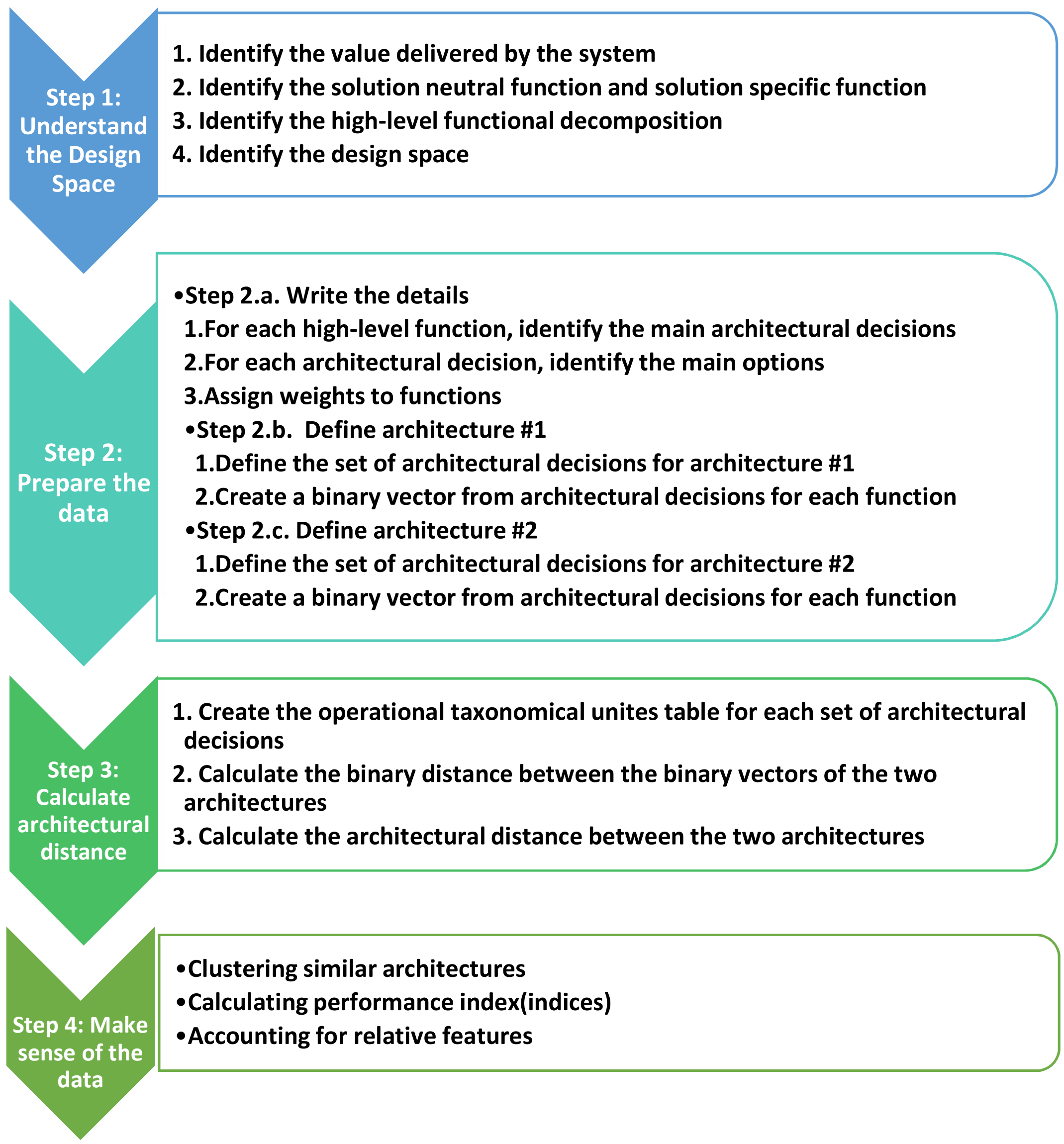


Figure 11. The proposed framework

In general, the framework utilizes information about system architecture. However, we will show how existing measures can utilize information about crisp features and relative features. The output is one (or more) indicator(s) that reflect the three aspects of the pair of solution variants. These indicators are the architectural distance between the architectures and one or more indicators for crisp and relative features. Note: information about crisp and relative features might not be available during conceptual design.

Next, we detail each step and provide illustrations. We only provide examples where illustration is needed.

Step 1: Understand the design space

In this step, the designer identifies the context of solution variants (for the intended system), collects information and defines the requirements. To achieve that we recommend the following plan of action:

1- Identify the value delivered by the system

The very first thing a designer should understand is the needs of the stakeholder. This will lead to identifying and understanding the delivered value. Value is the benefit at cost. "Architecture is function enabled by form"[10]. The designer should have a clear understanding of what is the value to be delivered.

2- Identify the solution neutral function and solution-specific function

Solution-neutral function is the function of a system stated without reference to how the function is achieved. The solution-specific function is a specialization of solution-neutral function (see example below).

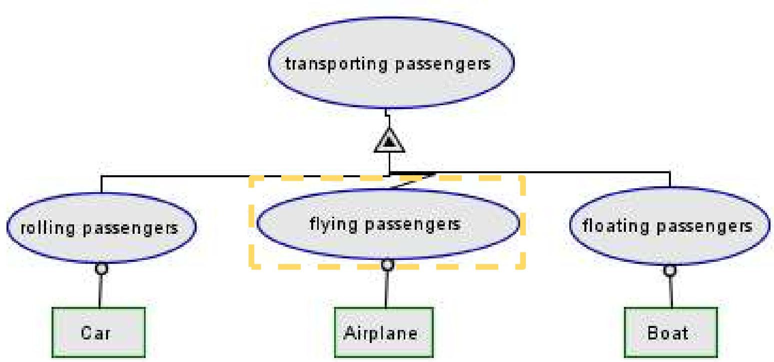


Figure 12. solution-neutral function and solution-specific function

3- Identify the high-level functional decomposition

The high-level functional decomposition shall include all the functions that contribute to the value. See example of a high-level decomposition of an aircraft.

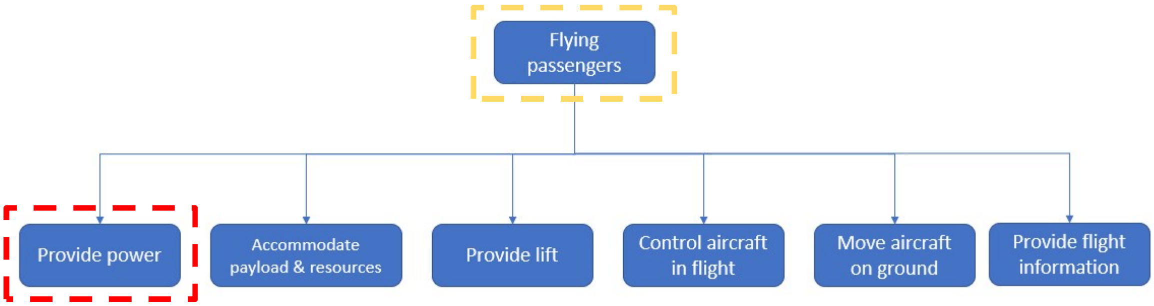


Figure 13. High-level functional decomposition for the function flying passengers

Depending on the required precision of the evaluation, the designer can go one or more levels down in the functional decomposition. Figure 14 shows the functional decomposition of the high-level function "provide power".

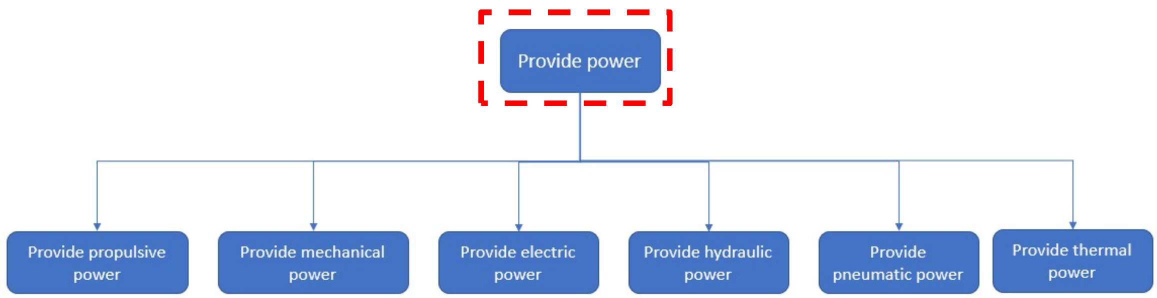


Figure 14. Second level functional decomposition for the function "provide power"

4- Identify the design space

The solution variants under evaluation can be part of a bigger pool of solutions that meet stakeholders' needs. Given that, the designer shall study the system in its bigger context. One possible method to expand the context of the evaluation is to conduct a literature review and/or survey the market for the existing solutions that can meet the needs and requirements.

Step 2: Prepare the data

Step 2.a. Write the details

1- For each high-level function, identify the main architectural decisions

In order to deliver a function, several architectural decisions should be made. Architectural decisions result in assigning elements of form to the function and defining the formal and functional interactions.

For example: the function provide propulsive power (Figure 13), can have architectural decisions that include (but are not limited to):

- Engine type
- Number of engines
- Engine location

2- For each architectural decision, identify the main options

For the architectural decisions; engine type, number of engines and engine location, there are many options. A list of few options for engine configuration is presented in Table 9

Table 9. Architectural decisions and their options for the function propelling payload

"Propulsion	3. Propulsion System Configuration (Propelling Payload)					
Engine Type	Piston Prop	Turboprop	Turbofan	Turbojet		
Number of Engines	1	2	3	4		
Engine Location	Inside Vertical Tail & underwing	Side of Fuselage Aft of wing before tail	Side of Fuselage & in Vertical Tail	Under Wing	In Wing	Above Wing"

3- Assign weights to functions

The weighting of each function can be obtained with many different methods such as the analytic hierarchy process (AHP) method, tabulated judgment method and so on. The tabulated judgment method is used in this work. In this method, the weighting factor for each function is calculated by pairwise comparison of the relative importance between two functions.

Here we will choose four functions to describe the procedure of obtaining the weightings (Table 10). For the function "provide power" concerns with six second-level functions but we will focus on four (for simplicity). The functions we will address are "provide propulsive power", "provide mechanical power", "provide hydraulic power" and "provide electrical power" (Figure 14). If the weighting functions of these four second-level functions with respect to the function "provide power" are to be decided, we can use this method by labeling these four second-level functions as FI, F2, F3, and F4, respectively. This procedure is described step by step as follows.

a. Put the functions on the row and column to form a 4 x 4 square matrix.

b. Compare pairwise the relative importance between two functions and give scores to them with the following criteria.

- If both the second-level functions have the same importance on their upper function, then both functions are rated at 2 points. For example, the functions F1 and F have the same significance on their related high-level function, so they are both rated 2 (See Table 10).
- If one second-level function is more important than the other second-level function on their main function, then the important one is rated 3 and the other is rated 1. For example, the function F2 is more important than F1, so the element at the position intersected by F2-row and Fl-column is given 1. Pairs (F2, F3) and (F3, F4) have the same condition.
- If one function is much more important than the other, then the important one is rated 4 and the other is rated 0. For example, the function F~ is much more important than F3, so the element intersected by Fl-row and F3-column is given 4, while the element intersected by F3-row and Fl-column is given 0.
- The diagonal elements in the pairwise comparison matrix represent the comparison of the function with itself, so they are not taken into consideration.

Table 10. Tabulated judgement method for 4 functions

Feature	F1	F2	F3	F4	K	weights
F1	0	1	4	2	7	0.29
F2	3	0	1	3	7	0.29
F3	0	3	0	1	4	0.17
F4	2	1	3	0	6	0.25
Significance: F1 =F2 >F4>F3						

c. Summing up the scores on each row after the pairwise comparison matrix has been finished, we can obtain sub-total scores (Ki) for each function, e.g. 7, 7, 4, and 6 in this example.

d. Summing up these sub- total scores, we can obtain the overall total scores as

$$\sum_{i=1}^{4} K_i = 24 \tag{11}$$

For an n-function system, the total sum can be checked with the following formula:

$$\sum_{i=1}^{n} K_i = 2(n^2 - n) \tag{12}$$

e. Calculate the weighting function for each function using the following formula:

$$W_i = \frac{K_i}{\sum K_i} \tag{13}$$

where W_i denotes the weighting function of i_{th} function.

Given this, we have assigned weights to functions. The steps should be applied to all the levels of decomposition.

Step 2.b. Define architecture #1

1- Define the set of architectural decisions for architecture #1

In case of evaluation for a few architectures, this can be done manually. Otherwise, for exploring large design space, we recommend using the compatibility matrix to filter the feasible designs first.

To illustrate, below we created a table with the set of architectural decisions of two aircraft architectures. The two aircraft are Boeing 727 and DC-10-15. In this case, each set consists of three architectural decisions (see Table 11 and Table 12)

Table 11. Architectural decisions for the engine of B727 and DC—10-15

	Boeing 727	**DC-10-15**
Engine Type	Turbofan	Turbofan
Number of Engines	3	3
Engine Location	Side of Fuselage & in Vertical Tail	Inside Vertical Tail & underwing

Table 12. Encoded architectural decisions for two aircraft B727 and DC-10-15

	Boeing 727	**DC-10-15**
Engine Type	3	3
Number of Engines	3	3
Engine Location	3	1

2- Create a binary vector from architectural decisions for each function

To create the binary vector for each function, all the options shall be listed, and a binary value shall be assigned in accordance with the architectural decisions for each architecture alternative. Table 13 shows the binary vector for the function propelling payload (engine configuration)

Table 13. Binary vectors for engine configuration of the three aircrafts

option	**B727**	**DC10**
Piston prop	0	0
Turbo prop	0	0
Turbofan	1	1
Turbojet	0	0
1 engine	0	0
2 engines	0	0
3 engines	1	1
4 engines	0	0
EL1	0	1
EL2	0	0
EL3	1	0

EL4	0	0
EL5	0	0
EL6	0	0

Step 2.c. Define architecture #2

1- Define the set of architectural decisions for architecture #2

In case of evaluation for a few architectures, this can be done manually. Otherwise, for exploring large design space, we recommend using the compatibility matrix to filter the feasible designs first.

To illustrate, below we created a table with the set of architectural decisions of two aircraft architectures. The two aircraft are Boeing 727 and DC-10-15. In this case, each set consists of three architectural decisions (see Table 11 and Table 12)

2- Create a binary vector from architectural decisions for each function

To create the binary vector for each function, all the options shall be listed, and a binary value shall be assigned in accordance with the architectural decisions for each architecture alternative.

Step 3: Calculate the architectural distance between the two architectures

1- Create the operational taxonomical unites table each set of architectural decisions

In chapter 2, we elaborated on the operational taxonomical units table (OUT). Here we will illustrate its work with an example. Let x and y be two architectures to be evaluated. The OUT can be seen in Table 14

Table 14. Operational taxonomical unites

Object/system/product	$\boldsymbol{x}$	$\boldsymbol{\bar{x}}$	**Sum**
$\boldsymbol{y}$	$a = x * y$	$b = \bar{x} * y$	$a + b$
$\boldsymbol{\bar{y}}$	$c = x * \bar{y}$	$d = \bar{x} * \bar{y}$	$c + d$
Sum	$a + c$	$b + d$	$n = a + b + c + d$

For the two aircraft engine architectures, we selected in the previous steps, OUT values are presented in Table 15

Table 15. OUT values for three aircrafts

Object/system/product	$B727$ & $DC10$
a	2
b	1
c	1
d	9

2- Calculate the binary distance between the binary vectors of the two architectures

As discussed in chapter 2, there exist many binary similarity and distance measures and the results are affected by the choice. We decided to use the Jaccard distance coefficient to calculate the binary distance between the architecture alternatives. Jaccard distance coefficient is calculated using the following equation:

$$D_{Jaccard} = 1 - \frac{a}{n-d} = \frac{b+c}{(a+b+c)} \tag{14}$$

Table 16. Architectural distance between the three aircraft architectures

Object/system/product	$B727$ & $DC10$
$Jaccard\ distance$	0.5

We recommend using the Jaccard distance measure for the following reasons:

- Its simplicity
- Our case studies on different examples showed consistent results when using the Jaccard distance measure in comparison to other measures.
- It does not account for the similar negatives (the features where the option is not used by any of architecture alternative)

If we have more than one function, which is the case usually, then for each function "$\boldsymbol{k}$", we calculate the distance between the pairs "$\boldsymbol{i}$"&"$\boldsymbol{j}$". The result of this step is the calculated set of distances $\boldsymbol{d_{ij}^{f_k}}$ for the sets of architectural decisions that correspond to functions.

3- Calculate the architectural distance between the two architectures

For each pair of architectures, calculate the overall distance by adding distances for each function multiplied by the corresponding weights. In chapter 5, the full implementation of the framework is demonstrated.

$$D_{ij} = w_1 \times d_{ij}^{f_1} + \cdots + w_k \times d_{ij}^{f_k} + \ldots + w_n \times d_{ij}^{f_n} \tag{15}$$

where $w_k - weight\ for\ function\ k, k = 1, \ldots, n$

Step 4: Make sense of the results

Now that we have calculated the architectural distance, we can proceed to make sense of the data that we have. The architectural distance accounts for the first aspect of a system – the architecture. The last step in this guideline is understanding the meaning of architectural distance and making use of in the design process. As we argued before, architectural distance can be thought of as the architectural cost of changing one architecture into another or the cost of changing one or more decision(s). The designer can use the results in many ways. This step can be done in many ways. Here we recommend three ways to make use of the results. First, we will talk about clustering similar architecture. Then we will present general formulations for calculating additional indicators that are based on crisp and relative features. For that, we use a generalized version of the performance index and the application of fuzzy logic. We will not go into the details in this chapter as these two approaches have been discussed in chapter 2 and will be elaborate more by the case study in chapter 5.

➢ Clustering similar architectures

There are many clustering algorithms that can be used to create clusters of architectures according to their architectural distance. Clustering algorithms include neighbor-joining algorithm, K-Means Clustering, Mean-Shift Clustering, and many others. Explaining the work of these clustering algorithms is out of the scoop of this thesis. However, in chapter 5, we provide an example of how clustering similar architectures can reduce the solution variants in design space. Consequently, this reduces the required computational power for any calculations and help the designer to understand how the changes in architecture affect design metrics.

- Performance index and architectural distance

Information about architecture can be used along with information about crisp features and relative features. In this section, the performance index is discussed as a way to account for crisp features. The notion of performance index has been around for a while. Many works have proposed different formulation for its calculation. Its main essence is to combine some system metrics (metrics that have meaning when combined) into one index that shows the overall performance with respect to the chosen metrics. Accounting for crisp features is beyond the scope of our framework. However, due to its importance, we detail the process of calculating the performance index as a method to account for crisp features during evaluation. Below we describe the process for calculating performance index.

1- List all the known crisp features.

These can be form attributes, function attributes or other attributes depending on what is known and what is important for evaluating the architecture alternatives. For example, for aircrafts performance metrics include engine type, engine name, number of engines, cruise velocity, price, capacity, …etc. Table 17 shows an example of crisp features for aircraft.

Table 17. Example of crisp features for aircraft engine CFM56-5

Crisp feature	Value
Engine Type	3
Thrust per engine [lbf]	27,000
List Price in 2015 [USD mill]	97

2- Group the features into sets that make sense from the comparison point of view.

For example, form properties such as shape, size, and weight can be grouped in one group that accounts for the system dimension, while thrust, weight can be grouped as one parameter – thrust-weight ratio.

3- Assign a weight to each feature

Use the tabulated judgment method (see step 2.a item 3) to assign weights the attributes in each group from the previous item. Each feature F_j will have a weight of w_j. All weights should add up to 1 (i.e., $\sum w_j = 1$)

4- Use the formula to calculate the Performance index

There are many formulations that combine parameters and characteristics in one index. For the purpose of this framework, we propose using the general formulation:

$$Performance\ index = w_1 \times \frac{F_{1i} - F_{1\min}}{F_{1\max} - F_{1\min}} + \mathrm{w}_2 \times \frac{F_{2i} - F_{2\min}}{F_{2\max} - F_{2\min}} + \ldots + \mathrm{w_j} \times \frac{F_{ji} - F_{j\min}}{F_{j\max} - F_{j\min}} \tag{16}$$

$Where,\ w_j - weight\ of feature$ j;

$F_{ji} - value\ of\ feature\ "j"\ for\ solution\ variant\ "i"$

After obtaining one or more indicators (performance indices) for crisp features, the information about the architecture can be used along with these indicators to make informed decisions. In chapter 5, we provide an example of how this can be done for the case study of commercial aircraft.

➢ Architectural distance and the application of fuzzy logic for architecture variants evaluations

The designer often has to account for the relative features of solution variants. In such situations, the common practice is to take the decisions that align with previous experience and knowledge. This approach is subjective and prone to biases. Eliminating the subjectivity and bias from judgment about the relative aspects of architecture alternatives is a topic that literature addressed, and we discussed in the literature review and in chapter 2. One of the most popular and effective methods that deal with fuzziness is the application of fuzzy logic. The fuzzy logic takes the fuzzy features in the input and gives a crisp value at the output. There can be multiple input variables and one or more output variables. Please refer to section 2.4.2 for a detailed explanation of the application of fuzzy logic for evaluation of relative features

3.3 Applications of the proposed framework

As we have seen, architectural distance in a new concept and it can be used in different ways to assess the evaluation process, explore the design space and so on. Below we list a few applications that we believe are important.

- Reduce the size of design space

By clustering architectures that have low architectural distance, the designer will be able to reduce the design space and the number of architectures they evaluate.

- Support the decision-making process

Design involves a lot of decision making. With the huge amount of information presented and the high levels of uncertainty, it is very useful to have indicators that reflect the major aspects of the system. In our framework, the designer will know the distance between architectures. In addition, the can calculate one or more indicators for crisp and relative features to evaluate the solution alternatives. Such amount of information is easier to comprehend than considering each feature individually.

- Study history of innovation for a certain product

Given the architectural distance between a specific product's architectures through the years can help to understand the shifts and milestones in the company's production history.

- Cataloging design concepts

The idea of cataloging engineered systems has been around for a long time. Cataloging all the engineered systems will help us understand the process of engineering design and evolution of systems.

e.g., Architectural distance can improve the international patent classification (IPC) by adding additional rigorous criteria for classification.

- Estimate the architectural costs of new products

This is specifically very useful if a company is looking to introduce a new product and would like to know the architectural cost of moving from the old product to the new one. The architectural cost in this case can be associated with other parameters such as cost or schedule.

On one hand, this might lead to suppressing innovation as big companies avoid radical changes (e.g., aircraft companies). On the other hand, this application can support the case for radical changes if architects show that the performance will radically be enhanced.

3.4 Limitations and recommendations

In this chapter, we presented the proposed framework as an approach to evaluating solution variant in the design space by comparing their architectures. The first part of the framework (Steps 1-3) returns the calculated architectural distance. In step 4, the designer combines information about the architectural distance with the known information about crisp and relative features to evaluate solution variants. Consequently, the designer can make better-informed decisions during conceptual design.

The framework has some limitation. below we discuss these limitations and how possibly we can overcome them:

1- The method for calculating architectural distance follows clearly defined steps and was constructed with the account to similar applications from scientific fields. However, there is a need for further validation.
The validation of the framework was improved with a survey study and a case study of a complex system. In chapter 4, we present the survey study and discuss its conclusions and outcomes. In chapter 5, we presented the results of the case study we conducted on commercial aircraft.

2- It does not account for the relationship between architectural decisions. This appears when two architectural decisions are coupled. For example, for a suborbital vehicle, if the designer chooses to use two models, then he/she must decide the takeoff/landing modes for each model individually. This is not the case if the architecture of the suborbital vehicle includes only one model.

To mitigate this, we can introduce a new variable to the framework that will account for the connectivity of the architectural decisions.

3- The architectural distance is affected by the weighting values. While the subjectivity element, in this case, can make the results unreproducible, it also can be thought as additional flexibility the designer could use to account for his/her priorities.

One possible way to improve this limitation is to introduce a more rigorous guideline for assigning weights. This can be done by providing detailed criteria that the function/feature should be screened by before assigning the weight.

CHAPTER 4:

SIMILARITY IN DESIGN – SURVEY STUDY

Previously, we have established the notion of similarity in design. We distinguished three aspects of a system in the conceptual design phase – architecture, crisp features, and relative features. In chapter 3, we presented the proposed framework for solution variants evaluation. The evaluation of solution variants by comparing their architectures is a new idea that requires further inspection. At the end of chapter two, we discussed some ways to improve the framework and increase its validity. One of the ways we suggested was conducting a survey study.

The survey study came in two parts: interviews and a questionnaire. The interviews took the form of informal and formal discussions with researchers from the systems engineering field. We took the interviews part of the survey as feedback that helped in shaping the framework in general and the questionnaire study in part. The questionnaire surveyed students and professionals from two universities on questions about similarity in design.

This chapter will mainly cover the questionnaire survey study. We start the chapter with an executive summary in section 4.1. Next, in 4.2, we discuss the background and objectives before we elaborate on the methodology in section 4.3. Then, we presented the results and concluded in sections 4.4 and 4.5. The questionnaire itself can be found in (C: The survey

Note: We use the words system and product interchangeably to refer to the objects presented in the questionnaire.

4.1 Executive summary

The survey's aim was to understand how young designers think about similarity in design. This was done to validate and refine the proposed framework in two ways. First, clarifying the importance of different aspects of a system. Second, finding correlations between the similarity values from the survey on one side and from the framework on the other side. The survey targeted over 250 students, and professionals from The Massachusetts Institute of Technology Systems Design and Management Program and the Skoltech Space and Systems Engineering Program from March 15th to April 30th. This section provides the key analytical points of the survey.

Participants answered questions about the importance of system aspects during the evaluation process. In addition, we also asked the participant to consider pairs of architectures from 4 groups of products and assign a similarity value in a 5-degree scale (extremely similar – Not at all similar). The four groups of products are chairs, wine openers, musical instruments, and means of transportation. As it is difficult to capture all aspects of a product (system), we focused on the architecture - the main aspect addressed in our framework. Nevertheless, the questionnaire included questions about crisp features (shape, size, color), and relative features (complexity).

Our analysis of the survey data identifies key points that should be considered when studying the concept of similarity for architecture alternatives.

- Functions that are directly to the delivered value are more important than those that fulfill a supporting rule. Supporting functions are more important than additional functions (i.e., functions that do not contribute to the delivered value). In other words, **value-related functions>supporting functions>additional functions**. In parallel, the importance of an element of form is directly correlated with how much it contributes to the delivered value. In other words, **value-related elements of form > supporting elements of form > additional elements of form**.
- The judgment about similarity is affected by:
    - Similarity in form and function in the first place.

      The more similar the form and function two systems are the more similarity values assigned to them.
    - Crisp features in the second place

      Participants listed crisp features in the following order of importance to the judgment about similarity: shape, performance, color.
    - Relative features

      We asked participants about the importance of complexity in the judgment about similarity and there was a major agreement that it comes last in the list we provided.
- Although we could not map the similarity values from the survey and those calculated by the framework, the ordinal similarity was partially matching for three of the groups (the exception was musical instruments). This can be explained by two reasons. First, the fact that participants have different levels of knowledge about the presented products. Second, participants considered aspects of the products that were not considered during calculations.

<u>Conclusions</u>: In general, participants listed architecture that achieves the same functions using similar tools (form) as similar/close. In contrast, they listed architectures that use different tools (elements of form) to deliver the function as dissimilar/far. The order of importance for similarity judgment starts with form and function then attributes of form and function followed by the fuzzy features.

The survey study helped in understanding how young professionals conceive similarity in design. In addition, it emphasized the division between the three categories of features that we use in the framework.

Recommendations for further inspection:

- Widen the population of the study to include people from different industries.
- Include collecting information for more complex and specialized systems (e.g., airplane wings, antibiotics, building construction techniques, … etc.) and surveying experts in these systems.
- Further efforts can include studying intangible systems such as services and algorithms.

4.2 Background and objectives

The survey study is part of the research we did to develop a framework for solution variants evaluation by comparing their architectures. The survey study was conducted in two parts. The early phase included interviews and discussion about the possible approaches to address the question in hand. The second part was a questionnaire.

We surveyed participants to evaluate the similarity between systems. In addition, participants answered questions about the importance of different aspects of a system to the estimated similarity. We specifically measured the importance of elements of form, functions and their relation to the delivered value from the system(object).

The information from the survey was used for two purposes. First, to distinguish the different aspects of a system and understand the importance of each aspect. Second, to find correlations between similarity values with the calculated similarity. Ultimately, these findings contributed to validating the framework and understanding how designers perceive similarity in design.

4.3 Methodology

The questionnaire targeted 250 students from two groups. The first group was the System Design and Management (SDM) Program students 2018 and 2019, Massachusetts Institute of Technology. The second group was students from the Space and Engineering Systems Program, Skoltech. Both groups have a background in systems engineering and system architecture. In addition, many students from the SDM group have worked in engineering projects in academia, industry or government. Although the targeted population belonged to different age and experience groups, the majority of the population have at least one degree in an engineering-related field. In total, we received 79 responses, from which 19 were excluded due to incomplete answers.

The questionnaire contained 18 questions, 16 numerical and 2 text questions. In numerical questions, we used standard Likert scales. The outline of the questionnaire consisted of 3 main sections. Section 1 was a brief introduction to the topic of the study. Next, in section 2, we introduced four groups of products - chairs, wine openers, musical instruments and means of transportation. For each group of products, we presented four different architectures. We asked the participants to evaluate the similarity between the architectures in each group. In addition, we asked them to state their level of expertise on each group of architectures and the importance of different elements of form. Finally, the last section of the survey asked participants about how important different aspects of architecture in the evaluation process between architecture variants are. Randomization was used in all the MCQs to eliminate the bias that can be created by the preorder of choices in questions. Below we describe the three sections in detail.

As mentioned above, section 1 introduced the participant to the study goal, the task at hand and briefly explained the main terms and research background.

For each group of architectures in section 2, we asked three questions. In the first question, participants answered the multiple-choice question "**How familiar are you with [name of product group]?**" by choosing one answer between 1-"expert knowledge (I study/develop them)" and 4- " No knowledge at all. (You can proceed to the next set of questions)". The second question was also a multiple-choice question. The question was " **When comparing [name of product group], how important are each of the following elements of form for delivering the overall function: [main function]?"**. We listed one element of form that contributes directly to the delivered value, one element of form which plays a supporting role but does not contribute directly to the delivered value, and finally an element of form that achieves an additional function that is not related to the delivered value. The third question was a multiple-choice question matrix with a list of 6 pairs of architectures in the rows. In this question, we asked participants to "**Please indicate how similar you think the designs are for each pair of transportation means**." The similarity scale ranged from 1- " Extremely similar " to 5-" Not at all similar ".

Section 3 starts with a text question" For each of the four sets of products you were asked to compare […] **what was your approach or thought process in determining the extent of similarity between designs?".** This open-end question allowed the participant to describe the criteria they used to evaluate the similarity between products. The second question in this section was a multiple-choice question: "Please choose all the features that you think are important when comparing any designs ". The list of choices included: form, function, performance, complexity, shape, and color. Next, we asked participants about their overall understanding of the questions they answered, " To what extent do you agree or disagree with the following statements: **"I understood the tasks I was supposed to do in the survey**." The answer choices ranged from 1- "strongly agree" to 5- "strongly disagree". The last two questions asked the participants about their occupation and place of work/study.

4.4 Results

79 responses were recorded, 19 of which were incomplete. Out of the 60 responses left, 52 responses came from MIT SDM affiliates and 8 responses were collected from participants affiliated with Skoltech and other organizations.

To process the data, we started by excluding the incomplete answers. Next, we grouped the data by the respondent's affiliation. The first group consisted of MIT affiliates. The second group included Skoltech and other affiliates. Then, we proceeded to analyze the data from the different groups in two stages. In the first stage, we processed the data from different groups separately. In the second stage, we processed and analyzed the data from all groups in one pool.

The results of data analysis from the first stage showed that participant from different backgrounds showed different answer patterns. In general, participants from the SDM group showed consistency in their answers throughout the questionnaire and the population. This can be explained as they have formal education on system engineering and system architecture. In addition, more than 60% of participants have experience in engineering projects. In contrast, participants in the second group showed inconsistency in answers through the questionnaire and the population. This might be caused by many factors that include the different educational backgrounds of participants and the lack of formal education on system architecture. The analysis below is presented for the results received from the first group unless stated otherwise.

4.4.1 Importance of form and function

The first important outcome we received was about the importance of different aspects of a system. 87% of participant stated that form is important for the evaluation of solution variants. Although the relationship between form and function is not always a one-to-one mapping, the same percentage of participants also stated that the function is as important as the form for the evaluation process. 58% of participants listed performance as important for the evaluation process followed by 33% and 24% for two form properties (namely, color and shape). Surprisingly, none of the participants thinks that the complexity of an architecture is important in the evaluation process. This was not clear with the second group as none of the architecture aspects was voted by more than 40% of the participants. Responses of the second group showed that shape and color are slightly more important than the form and function when evaluating architecture variants (see Figure 15).

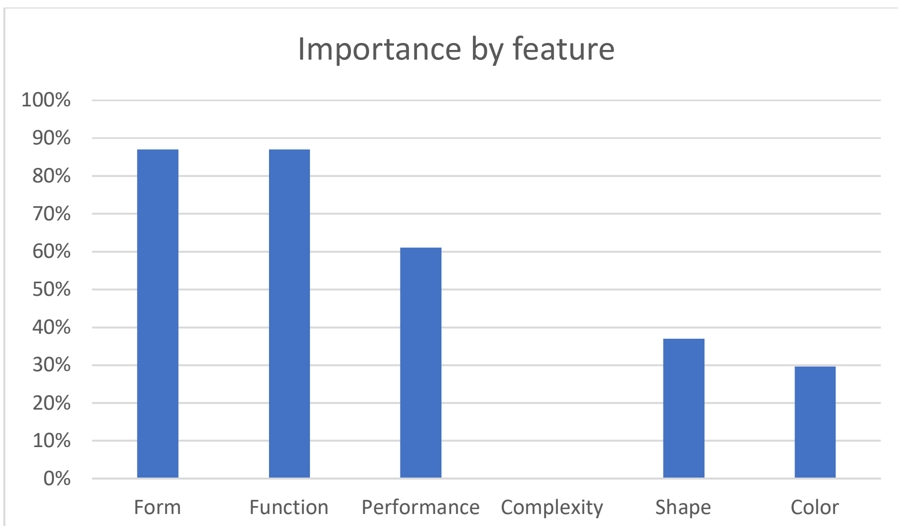


Figure 15. Importance of different features for comparing two products

4.4.2 Value and form and function

One of the assumptions we wanted to verify with the questionnaire was about the importance of different elements of form. The assumption was that, when comparing systems, functions (and their corresponding elements of form) shall receive different weighting values according to how much they contribute to the delivered value. In other words, functions (and elements of form that directly deliver the value shall have a higher weighting value than functions (and elements of form that have a supporting role or are not contributing to the delivered value.

The survey responses were in sync with this hypothesis as participants ordered the importance of elements of form for the 4 architecture groups as follows:

1- Main elements of form (directly deliver the value)
2- Supporting elements of form (supporting role)
3- Additional elements of form (deliver additional (extra) value)

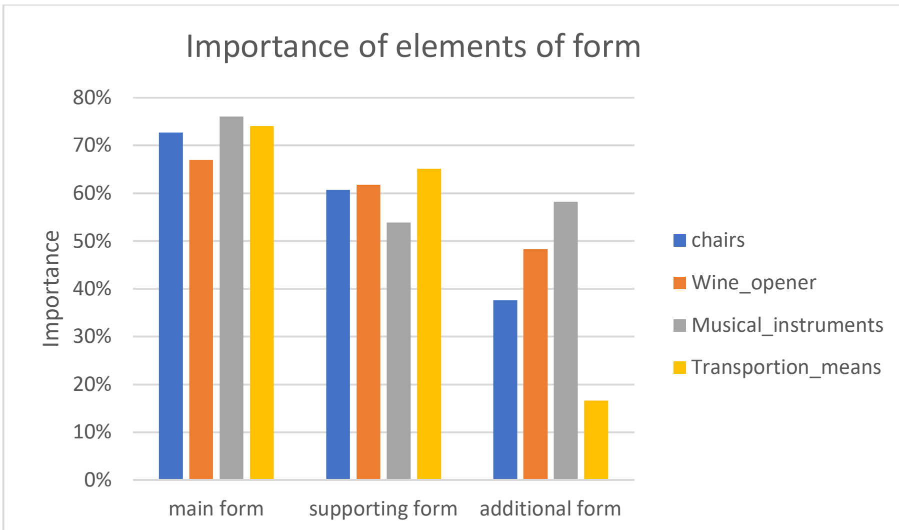


Figure 16. Importance of three types of form

4.4.3 The concept of similarity and architecture alternatives evaluation

The groups of architectures in the survey were deliberately selected to have a common main function for each group. We chose simple products to minimize the elements that would be considered when evaluating similarity between architectures. We also asked participants not to consider the costs as this aspect is often not known during the conceptual design phase and it can add an unneeded level of complexity to the evaluation process.

Overall, pairs that have more similar main form were considered more similar than pairs with different main form. This was less significant for pairs that have a different supporting form or additional form. Figure 17 shows the DISTANCE (not similarity) between pairs from 4 products groups. In the previous chapters, we established the connection between similarity and distance.

In short, $\boldsymbol{distance} = \mathbf{1} - \boldsymbol{similarity}$ in case of normalized similarity value. For example, Scandinavian chair has larger distance to all the other alternative in the group of chairs. This resonates with our assumption about the importance of types of form because the Scandinavian chair have a different main form in comparison to the other alternatives in the group.

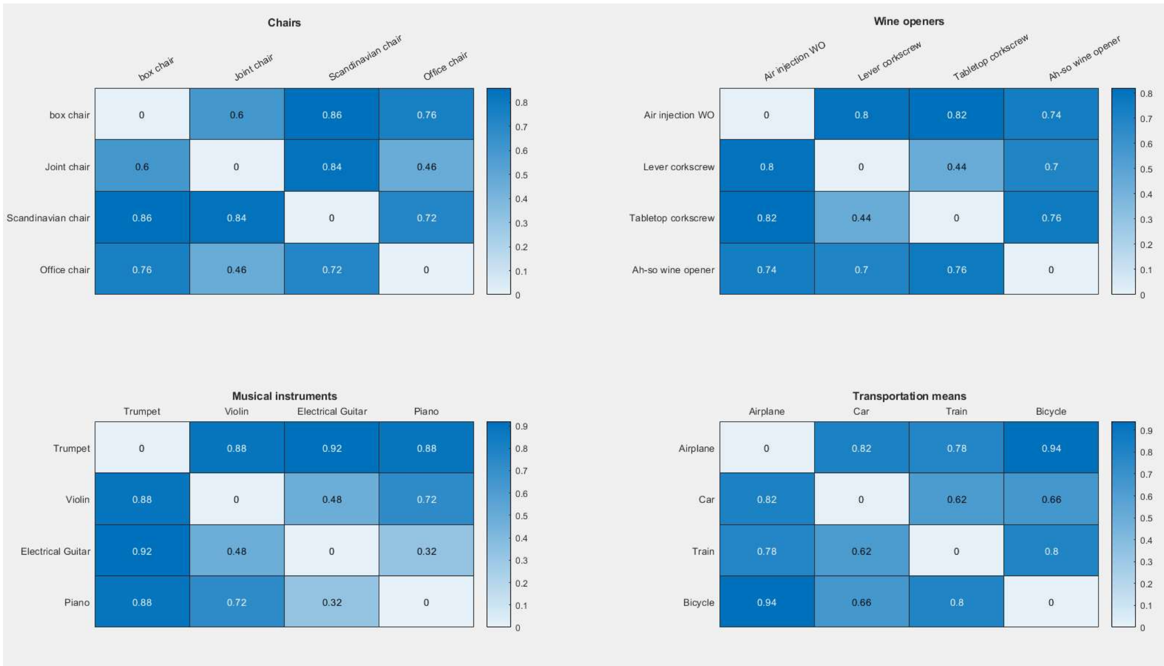


Figure 17. Distance values between pairs from the four groups of products

4.4.4 Matching values of similarity from the questionnaire with those calculated by the framework

Although we made a great effort to minimize the aspects participants shall consider in the comparison, we could not map the similarity values from the survey and those calculated by the framework. We chose simple systems and eliminated the cost from the comparison. However, the ordinal similarity was partially matching for three of the groups with the exception being the group of musical instruments. This can be explained by two reasons. First, the fact that participants have different levels of knowledge about the presented products. Second, participants considered aspects of the products that were not considered in our calculations. Figure 18 shows the distance values acquired from the survey compared with those we calculated with the framework.

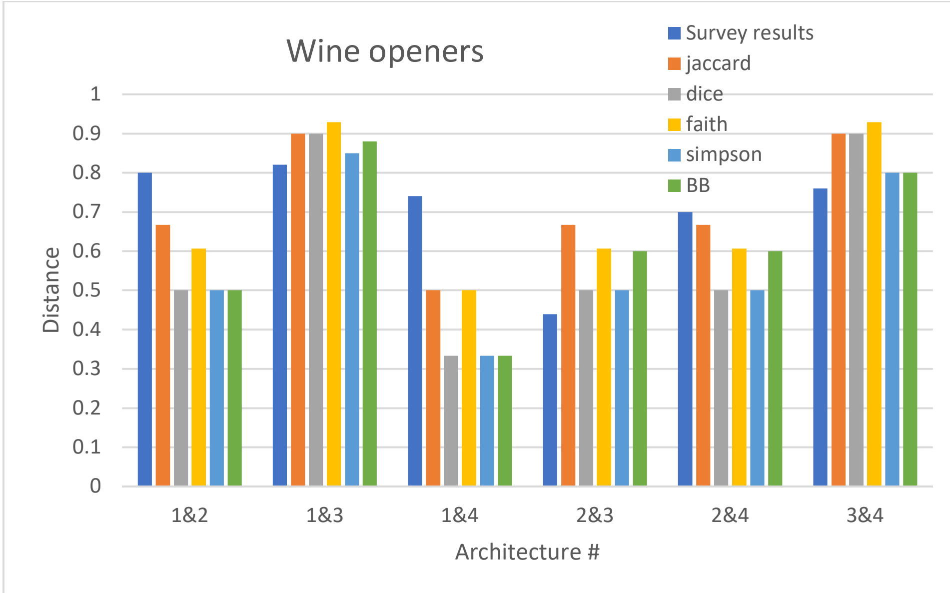


Figure 18. Distance between wine openers pairs from survey data and from framework using 4 different similarity measures

4.5 Conclusion and recommendations

To conclude, the questionnaire was of great value in understanding how young professionals understand the notion of similarity in design. We were able to verify two assumptions. The first assumption about the importance of the different system aspects and the second about the importance of different types of form (main, supporting and additional). In addition, we were able to establish a partial correlation between similarity values from the questionnaire with those calculated using the framework

The questionnaire findings can be summarized as follows:

- Function and form are of most importance when comparing pairs of systems
- Participants indirectly distinguished three categories of feature (by assigning importance value to different features)
- The importance of functions and their corresponding elements of form is directly related to how the value they deliver is related to the overall function
- Ordinal correlation between the survey values for similarity and those calculated using the framework was found for three product groups.

Recommendations for further inspection:

- Broaden the population of the study to include people from different industries.
- Collecting information for more complex and specialized systems (e.g., airplane wings, antibiotics, building construction techniques, … etc.) and surveying experts in these systems.
- Include intangible systems such as services and algorithms in the study

CHAPTER 5:

CASE STUDY – COMMERCIAL AIRCRAFT

Earlier, we showed that in the past century, many researchers have developed evaluation methods for specific systems to navigate through design space and make tradeoffs. In previous chapters, three aspects of a system were distinguished during conceptual design. The first being the architecture. The second and third are the crisp features and relative features. This clear division for system aspects allowed for the development of our framework. We developed a method for comparing the architectures of solution variants using similarity measures. In addition, we provided a general formulation for evaluating the crisp and relative features.

In chapter 4, we discussed the results of the survey study we carried out to verify the assumptions of our framework. The chapter discussed the implications of the results. One specifically important implication was that form and function are of most importance for comparing solution variants during the conceptual design phase. In chapter 3, a detailed description was provided for the proposed framework for solution variant evaluation by comparing their architectures. We also provided a general formulation for evaluation for crisp features and relative features. Although we provided some illustrative examples in chapters 2 and 3, in this chapter we will provide a case study that will demonstrate the work of the framework. We will use the case study as an example to show some practical applications designers can benefit from.

This chapter will start by introducing the database in section 5.1. Next, we will apply the framework step by step to calculate the architectural distance between the aircraft architectures. Then, we will provide an example of performance index calculation. We will discuss how this indicator can be used along with the architectural distance to provide important insights for the designer. Finally, we conclude with a short summary of the chapter and results.

5.1 Introduction to the database

The example, we are going to apply the framework to, is a database of aircrafts designs in the period between 1913 to 2016. The database was collected and refined by Demetrios[28] from various resources. The database includes airliner commercial aircraft and excludes military aircraft, cargo aircraft, and rotorcraft. This was done to maintain consistency. In addition, another criterion was added to maintain a consistent set of needs. This criterion excluded aircraft with a capacity of less than 30 passengers. This was done because aircraft with a passenger capacity less than 30 (usually business and light aircraft) have different design drivers from commercial passenger aircraft. To exclude experimental aircraft architectures, an additional constraint was introduced. The constraint is imposed on the set of aircraft is that the minimum number manufactured is 10. "This is done to exclude experimental aircraft, which may exhibit aircraft architectures that are not commercially viable, which is a proxy for commercially feasible production and 'value' in the market.[56]"

Designers make important decisions related to the architecture of the aircraft during the conceptual design phase. As mentioned in chapter 3, the architecture is "*the allocation of function to elements of form and definition of relationships among the elements and with the surrounding context.*[10]" Demetrious argues that the choice between twin-aisle or single-aisle is a design parameter related to the performance of the aircraft, rather than an architectural decision that enables a new function to be fulfilled. This can be understood from the definition we adapted for architecture early in this work. In the database, Demetrios chose the set of architectural decisions that enable the most important functions to be fulfilled. In our case study, we will address only four of these functions. These functions are lifting payload, propelling payload, providing control (directions in flight), and taxiing payload. We will not go deeper into discussing the database. More information about the database can be found in Demetrios thesis[28].

5.2 Applying the framework

In this section, two aircraft architectures are examined by the proposed framework.

Step 1: Understand the design space

1. Identify the value delivered by the system

We previously mentioned that "value is benefit at cost". For our set of aircraft, the benefit can be thought of as changing the location of the passenger from A to B.

2. Identify the solution neutral function and solution-specific function

The solution-neutral function of aircraft is "transporting passengers"[3]. The value is changing the location of the passenger. The solution specific function specializes the process "transporting" to "flying". The solution specific function, therefore, is "flying passengers" (See Figure 19).

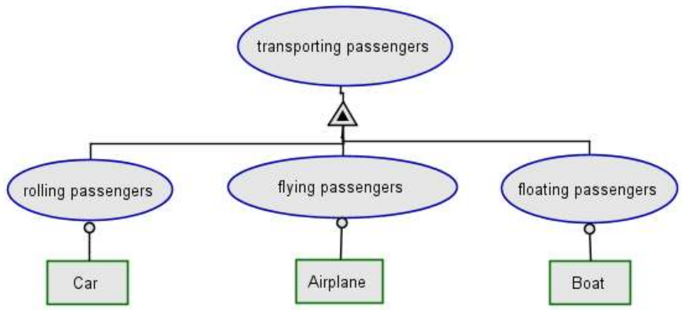


Figure 19. Solution-neutral and solution specific functions

3. Identify the high-level functional decomposition

As mentioned in the introduction of this chapter, we will focus only on four functions for simplicity (see Figure 20).

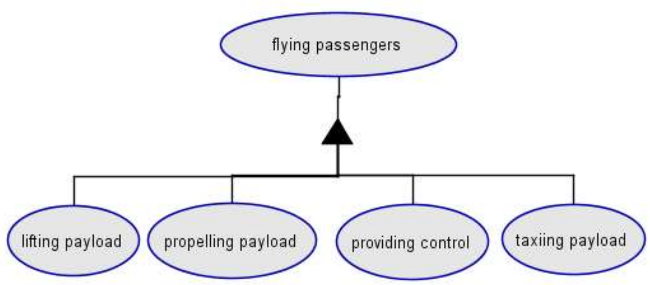


Figure 20. High-level functional decomposition

1. Identify the design space

[3] Other functions such as carrying weapons and cargo were excluded as we only focus on airliner commercial aircraft.

For our case study, the design space is pre-defined. It consists of 156 aircraft.

Step 2: Prepare the data

Step 2.a. Write the details

1- For each high-level function, identify the main architectural decisions

Previously, a list of four functions was established for the set of aircraft. Each of these functions corresponds to one (or more) form. To keep things simple, we will assume a one-to-one mapping between function and form. The functions correspond to wing configuration, propulsion system configuration, control system configuration and landing gear configuration respectively. Below we list some[4] of the architectural decisions for each function.

F1. lifting payload
=> wing configuration: Wing Vertical Location, Wing Shape, Passive Control Shape

F2. propelling payload
=> propulsion system configuration: Engine Type, Number of Engines, Engine Location

F3. providing control
=> Pitch control configuration: Location (along height), Angle, Shape
=> Yaw control configuration: Configuration

F4. taxiing payload
=> landing gear configuration: Mechanism, Landing Gear Arrangement, Location of body Gear

[4] This is not intended to be an exhaustive list.

2- For each architectural decision, identify the main options

Table 18. Functions, architectural decisions and options for aircraft architecture

	Option 1	Option 2	Option 3	Option 4	Option 5	Option 6	Option 7	Option 8	Option 9
Wing	**1. Wing Configuration (Lifting Payload)**								
Wing Vertical Location	High Wing	Mid Wing	Low Wing	Parasol Wing	Shoulder Wing				
Wing Shape	Rectangular	Tapered	Delta	Swept Back	Swept Forward	Elliptical			
Passive Control Shape	Dihedral	Anhedral	Straight		Polyhedral				
Propulsion	**2. Propulsion System Configuration (Propelling Payload)**								
Engine Type	Piston Prop	Turboprop	Turbofan	Turbojet					
Number of Engines	1	2	3	4					
Engine Location	Inside Vertical Tail & underwing	Side of Fuselage Aft of wing before tail	Side of Fuselage & in Vertical Tail	Under Wing	In Wing	Above Wing			
Control	**3. Control system configuration (Provide Control)**								
PITCH	**PITCH**								
Location (along height)	Fuselage (low - inverted T)	Vertical Tail (T-Tail)	Vertical Tail (cruciform)	N/A					
Angle	Dihedral	Anhedral	Straight		N/A				
Shape	Tapered	Rectangular	Swept Back	Elliptical		N/A			
YAW	**YAW**								
Configuration	1 Vertical Tail	2 Vertical Tails	3 Vertical Tails						
Landing Gear	**4. Landing Gear Configuration (Taxiing payload)**								
Mechanism	Fixed	Retractable Enclosed	Retractable not enclosed						
Landing Gear Arrangement	Single Main	Tail Dragger	Tricycle	Nose gear, triple bogie body gear	Nose gear, 4 bogie body gear	Nose gear, multiple Body gear sets	N/A		
Location of body Gear	In the Wing	Wing Podded	In the Fuselage	Fuselage Podded	Wing-Fuselage	In Nacelle	Wing-Fuselage & Fuselage (3+ body gears)	Fuselage Side-Podded	Fixed to Wing

3- Assign weights to functions

In chapter 3, the tabulated judgment method was described for assigning weight to functions. An example was provided as well. In this chapter, we will consider the functions as equally important. For simplicity, We assign weights of 0.25 to the four functions we are considering. The weights are shown in Table 19:

Table 19. Assigned weights for functions of aircraft

Function	**weight**
lifting payload	0.25
propelling payload	0.25
providing control	0.25
taxiing payload	0.25

Step 2.b. Define architecture #1

1- Define the set of architectural decisions for architecture #1

a) Wing configuration = {low wing, swept back, Dihedral}
b) Propulsion system configuration = {turbofan, 2, under wing}
c) Control system configuration = {Fuselage (low - inverted T), Dihedral, Swept back, Vertical tail}
d) Landing gear configuration = {retractable, Tricycle, Wing-Fuselage & Fuselage }

2- Create a binary vector from architectural decisions for each function

a) Wing configuration

$$W_b = [0,0,1,0,0,0,1,0,0,0,0,1,0,0,0]$$

b) Propulsion system configuration

$$prop_b = [0,0,1,0,0,0,0,1,0,0,0,1,0,0]$$

c) Control system configuration

$$ctrl_b = [1,0,0,0,1,0,0,0,0,0,1,0,0,1,0,0]$$

d) Landing gear configuration

$$\mathrm{lg}_b = [0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0]$$

Step 2.c. Architecture #2

1- Define the set of architectural decisions for architecture #2

a) Wing configuration = {low wing, swept back, Dihedral}

b) Propulsion system configuration = {turbofan, 4, under wing}

c) Control system configuration = {Fuselage (low - inverted T), Dihedral, Swept back, Vertical tail}

d) Landing gear configuration = {retractable, (Nose gear or 4 bogie body gear), Wing-Fuselage & Fuselage}

3- Create a binary vector from architectural decisions for each function

a) Wing configuration

$$W_b = [0,0,1,0,0,0,0,0,1,0,0,1,0,0,0]$$

b) Propulsion system configuration

$$prop_b = [0,0,1,0,0,0,0,1,0,0,0,1,0,0]$$

c) Control system configuration

$$ctrl_b = [1,0,0,0,1,0,0,0,0,0,1,0,0,1,0,0]$$

d) Landing gear configuration

$$\mathrm{lg}_b = [0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0]$$

	Option 1	Option 2	Option 3	Option 4	Option 5	Option 6	Option 7	Option 8	Option 9
Wing	**1. Wing Configuration (Lifting Payload)**								
Wing Vertical Location	High Wing	Mid Wing	Low Wing	Parasol Wing	Shoulder Wing				
Wing Shape	Rectangular	Tapered	Delta	Swept Back	Swept Forward	Elliptical			
Passive Control Shape	Dihedral	Anhedral	Straight		Polyhedral				
Propulsion	**2. Propulsion System Configuration (Propelling Payload)**								
Engine Type	Piston Prop	Turboprop	Turbofan	Turbojet					
Number of Engines	1	2	3	4					
Engine Location	Inside Vertical Tail & underwing	Side of Fuselage Aft of wing before tail	Side of Fuselage & in Vertical Tail	under Wing	In Wing	Above Wing			
Control	**3. Control system configuration (Provide Control)**								
PITCH	PITCH								
Location (along height)	Fuselage (low - inverted T)	Vertical Tail (T-Tail)	Vertical Tail (cruciform)	N/A					
Angle	Dihedral	Anhedral	Straight		N/A				
Shape	Tapered	Rectangular	Swept Back	Elliptical		N/A			
YAW	YAW								
Configuration	1 Vertical Tail	2 Vertical Tails	3 Vertical Tails						
Landing Gear	**4. Landing Gear Configuration (Taxiing payload)**								
Mechanism	Fixed	Retractable Enclosed	retractable not enclosed						
Landing Gear Arrangement	Single Main	Tail Dragger	Tricycle	Nose gear, triple bogie body gear	Nose gear, 4 bogie body gear	Nose gear, multiple Body gear sets	N/A		
Location of body Gear	In the Wing	Wing Podded	In the Fuselage	Fuselage Podded	Wing-Fuselage	In Nacelle	Wing-Fuselage & Fuselage (3+ body gears)	Fuselage Side-Podded	Fixed to Wing

Boeing 747 -200

Boeing 757 -200

Figure 21. Architectural decisions for Boeing 747 and Boeing 757

Step 3: Calculate the architectural distance between the two architectures

1- Create the operational taxonomical unites table for each set of architectural decisions

Table 20 shows the operational taxonomic units for the four functions considered for the two architectures B747 and B757. For a detailed description of how to create the OTU table refer to chapters 2 and 3.

Table 20. OTU for four sets of architectural decisions of two aircraft architectures(B747&B757)

Name	**a**	**b**	**c**	**d**
Wing configuration	3	0	0	12
Engine configuration	2	1	1	10
Control system configuration	4	0	0	14
Landing gear configuration	1	2	2	14

2- Calculate the binary distance between the binary vectors of the two architectures

Table 21 shows the architectural distances between four sets of architectural decisions for B727 and B757. For a detailed description of how to calculate binary distance from binary vectors refer to chapters 2 and 3.

Table 21. Binary distance values for four sets of architectural decisions of B747&B757

Name	Function	symbol	binary distance between architecture #1 and architecture #2
Wing configuration	lifting payload	d_1	0
Engine configuration	propelling payload	d_2	0.5
Control system configuration	providing control	d_3	0
Landing gear configuration	taxiing payload	d_4	0.8

3- Calculate the architectural distance between the two architectures

To calculate the architectural distance, we sum the binary distances from Table 21 multiplied by the weights (see Table 19).

$$D = w_1 * d_1 + w_2 * d_2 + w_3 * d_3 + w_4 * d_4$$

$$D = 0.25 \times 0 + 0.25 \times 0.5 + 0.25 \times 0 + 0.25 \times 0{,}8 = 0.325$$

A MATLAB transcript was created to automate the calculation of architectural distance. Architectural distance was calculated pairwise for 157 architectures, generating 12246 pairs. Figure B1 is the branching tree for these architectures. The branching tree uses the neighbor-joining algorithm to collect items in a tree according to the distance between these architectures. Branching tree is famously used for representing biological species in what is called phylogenetic tree (phylotree). For such a diagram, the node in the common set of architectural decisions (nearest ancestor in case of phylotree) and the path length is the architectural distance (genetic distance in case of phylotree).

Step 4: Make sense of the results

Now that we have calculated the architectural distance between aircraft architectures, we can proceed to make sense of the results. Here we explore some ways of using the architectural distance.

- Clustering similar architectures

  If we consider our database as a design space, an important way to reduce its size is to cluster the similar architectures. We use MATLAB to cluster 157 aircraft into 54 unique architectures with zero architectural distance between pairs (see Figure 22). The designer can reduce the design space even further by, for example, clustering architectures with architectural distance $D < 0.1$. Figure 22 shows a heat map of aircraft architectures with a dendrogram. Aircrafts can be seen on the X-axis and Y-axis. The darkest the degree of the red color, the larger the architectural distance.

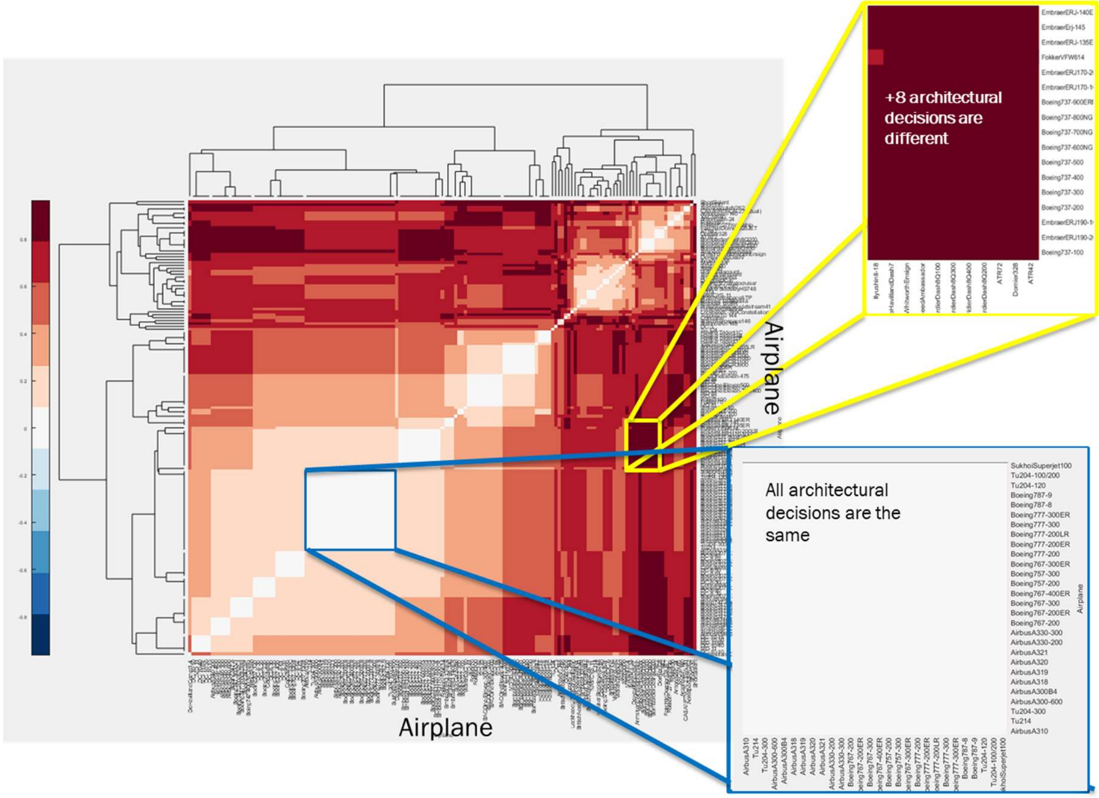


Figure 22. Clustering similar aircraft architectures

➢ Calculate performance index for

Performance index accounts for a selection of features that the designer chooses to fit with his/her preferences. For our example we use the formula below to calculate the performance index[28]:

$$Performanc\ Index = w_1.\ \left[\frac{PCE_i - PCE_{min}}{PCE_{max} - PCE_{min}}\right] + w_2.\ \left[\frac{T_{wi} - T_{wmin}}{T_{wmax} - T_{wmin}}\right] + w_3.\ \left[\frac{V_i - V_{min}}{V_{min} - V_{max}}\right] + w_4.\ \left[\frac{p_i - p_{min}}{p_{min} - p_{max}}\right] \quad (17)$$

Where: $$PCE_i = \frac{N_{pi}\ .\ R_i}{(W_{0i}\ - W_{ei}\ - W_{pi}\ . N_{pi})}\ is\ Passenger\ carrying\ efficiency \quad (18)$$

$N_{pi} - Passenger\ capacity\ of\ aircraft$;

$R_i\ - range\ with\ maximum\ payload$;

$W_{0i}\ - maximum\ takeoff\ weight$;

$W_{ei} - weight\ of\ the\ aircraft\ (exluding\ th\ fuel\ and\ payload\ );$

$W_{pi} -\ weight\ assigned\ to\ a\ passenger$

$$T_{wi}\ = \frac{Ti \cdot NE}{W_{0i}}\ - thrust\ to\ weight\ ratio$$

$V_i - maximum\ cruise\ velocity,$

$p_i - listed\ price$

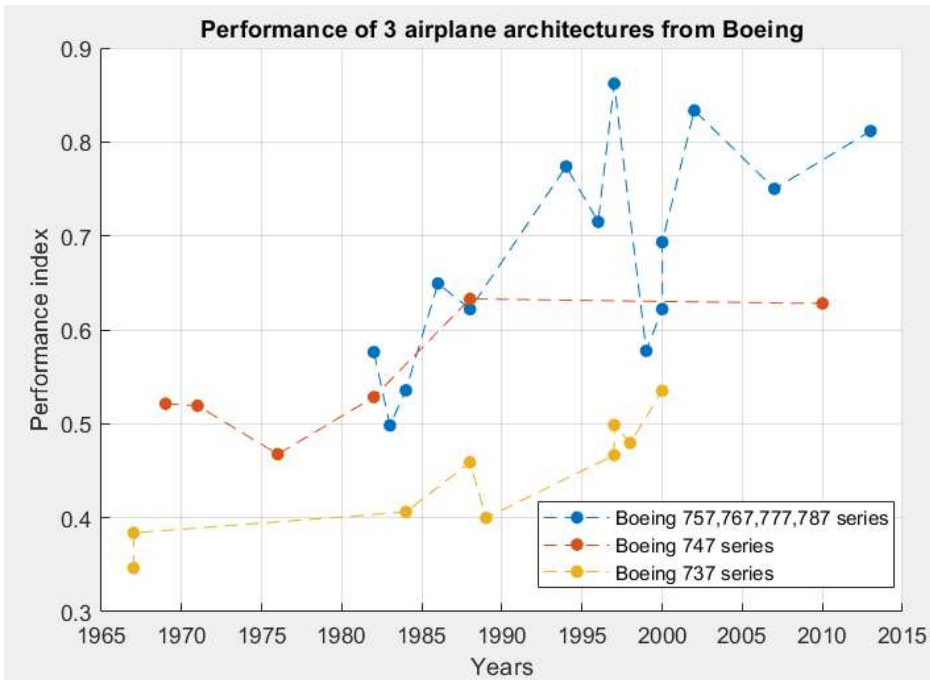


Figure 23. Performance of three unique Boeing architectures

Information about the performance can be used along with information about the architectural distance to get important insights. For example, Figure 23 shows the performance "P" through years for Boeing 737, Boeing 747 and Boeing 757-787. It is clear that $P_{757-787} > P_{B747} > P_{B737}$. The architectural distance (see Figure 24) provides additional insight. From the figure, it can be seen the architectural distance between B737 and B757-787 is shorter than that between B747 and B757-787. This piece of information can be very useful if the designer is to decide to move from one architecture to another (for example from B747 or B737 to B757-787). The architectural distance (cost) will be less in the latter case but the performance increase will be higher.

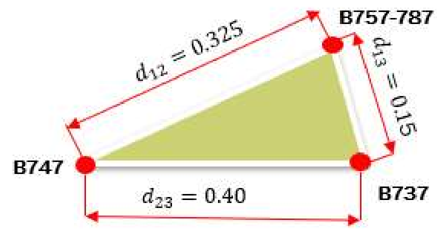


Figure 24. Architectural distance between three unique Boeing architectures

➢ Understanding the architectural history of commercial aircraft architectures

Figure 25 illustrates the number of distinct aircraft architectures (green line) and the average architectural distance between architectures. With time, architectures become more similar. This piece of information is in alignment by the well-established idea of dominant architectures. Throughout time some architectural decisions (can be generalized to architectures) become dominant.

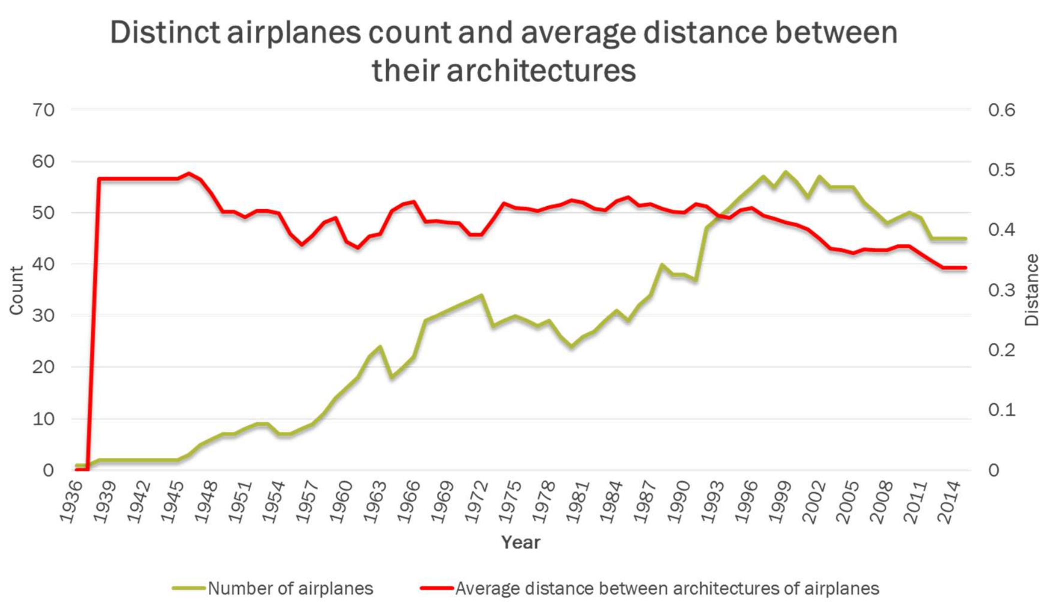


Figure 25. Average architectural distance throughout time

5.3 Conclusion

Chapter 5 demonstrated the application of the proposed framework to a set of airliner commercial aircraft. The database was used from a previous work of Kellari[28]. For this set of aircraft, architectural distance was calculated. In addition, performance index was calculated for three distinguished Boeing architectures.

To show the value of the results, few possible applications of the calculated architectural distance were demonstrated. First, 156 airplanes were clustered into 57 unique architectures. This allowed for decreasing the design space and consequently can reduce the effort the designer need to evaluate solution variants. Next, another important application was demonstrated by combining the indicator of architectural distance and that of performance index of three distinguished Boing aircraft architectures. Finally, we showed how the history of architectural changes for a specific product can be studied given the architectural distance.

CHAPTER 6:
CONCLUSION

In this thesis, the notion of architectural distance was introduced. In addition, a framework for solution variants evaluation by comparing their architectures was proposed. In this chapter, we conclude the thesis by restating the research question (6.1), providing a short summary (6.2), listing the main findings (6.3) and discussing the practical implications (6.4) and limitations (6.5). Finally, we consider the scope of future research. (6.6)

6.1 The problem

The conceptual design phase is characterized by the lack of information, a high level of uncertainty and fuzziness. Designers rely on their intuition, experience and design guidelines for solution variants evaluation. In the literature review, we identified a gap in the research on evaluation methods. We noted that most evaluation methods are case based, lack a general formulation and do not account for the architecture. Given that, this research addressed the need for a rigorous method for solution variant evaluation by comparing their architectures.

6.2 Summary

In this work, we started by reviewing the literature on evaluation methods and similarity. Later, three main aspects of engineered systems were distinguished. The first is the architecture. The second and third categories are concerned with system's features - crisp features and relative features. After identifying the three aspects of a system, the issue of evaluating solution variants was tackled. While there are many methods that address crisp and relative features, the focus of this work was comparing architectures. For that purpose, a framework for solution variant evaluation was proposed.

Solution variants evaluation by comparing architectures was the basis of the proposed framework. The framework utilizes binary similarity measure to calculate a value of distance between architectures. In this method, architecture is presented as a binary vector of architectural decisions. We introduced the term "architectural distance" to refer to the value obtained by the binary similarity measure. Architectural distance takes a value between 0 and 1. The higher the architectural distance between two architectures, the less similar they are. Next, we introduced a general formulation of for calculating the performance index for crisp features. And showed the application of fuzzy logic for evaluation of relative features. This was done to account for crisp and relative features in the evaluation process. The outcome from the proposed framework is one or more indicators that represent the three aspects of solution variants under comparison. These indicators are architectural distance, performance index (indices) and one or more indicators for relative features. Such information can provide important insight for designers to make informed decisions during conceptual design.

In this thesis, we carried out a survey study and a case study. The survey study aimed at understanding how young designers perceive similarity in design. This helped verifying important assumptions about the importance of system aspects and assigning weights for different functions. Although correlation was not ideal between participants estimation of similarity and those we calculated, we were able to find ordinal correlation for ¾ of the cases.

In the case study we demonstrated the application of the framework to a set of 156 aircraft architectures. We calculated the architectural distance between 12246 pairs of aircraft architecture. Later we showed how such information can be used to reduce the size of design space, help to make informed decisions and understanding history of innovation for a certain product.

6.3 Main findings

1- There is a distinction between three aspects of a system – architecture, crisp features, and relative features
2- The system architecture is of most importance for the evaluation process during conceptual design. Second in importance are the crisp features related to performance characteristics and form properties.
3- A system architecture can be presented as a binary vector of architectural decisions. This was the basis for using binary similarity measures to evaluate architectural distance.
4- A framework was developed for calculating architectural distance using binary similarity measures. Our framework can capture the main architectural decisions and return the value of architectural distance between solution variants.
5- The importance of functions, elements of form and architectural decisions is directly related to how much they contribute to the value delivered by the system

6.4 Practical implications of the findings

The framework provides important information about the architecture of solution variants. Such information can be utilized in many ways. While most practical implementations are positive, it can be used to suppress innovation. Below is a list of some of the possible applications:

- Reduce the size of design space
- Support the decision-making process
- Study the history of innovation for a certain product
- Cataloging and classification of engineered systems
- Estimate the architectural costs of new products

6.5 Limitations and recommendations

1) In the real world, architectural decisions are often coupled. This means that the change of one architectural decision can affect another architectural decision. The proposed framework does not directly account for this property. However, coupled architectural decisions can be assigned more weight during evaluation. This can mitigate the effect of coupled architectural decisions. Another way is to introduce a new variable to the framework that will account for the connectivity of the architectural decisions.
2) The architectural distance is affected by the weighting values. While the subjectivity element, in this case, can make the results unreproducible, it also can be thought as additional flexibility the designer could use to account for his/her priorities.

One possible way to improve this limitation is to introduce a more rigorous guideline for assigning weights. This can be done by providing detailed criteria that the function/feature should be screened by before assigning the weight

6.6 Scope for future research

1) Architectural distance and category theory

Exploring the possibility of representing the system architecture as a functor and the architectural distance as a natural transformation.

2) The transitive relationship between architectures.

Consider three architectures A, B and C. An interesting direction for research would be studying the transitive relationship between architectures. If the architectural distance between A and B is D_{AB} and the distance between B and C is D_{BC}, what can be interfered about the distance between A and C D_{AC}?

3) Architectural distance as an analogy for genetic distance.

In biology, the concept of similarity has many useful applications. One of which is categorizing species (phylogenetic tree). Is it possible to draw a phylogenetic tree for all engineered systems?! A genome map for engineered systems?

To sum up, the research distinguished three aspects of a system and proposed a framework for evaluation. We demonstrated the work and applications of the framework with a case study. In addition, we carried out a survey study and discussed its results. The framework provides powerful insights about the architecture for both novice and seasoned designers. The concept of architectural distance has a wide variety of application that includes, but is not limited to, solution variants evaluation, reducing the size of design space, design documentation and cataloging, and studying the history of innovation for products.

REFERENCES

[1] M. M. Andreasen, C. T. Hansen, and P. Cash, *Conceptual Design*. Cham: Springer International Publishing, 2015.

[2] V. Hubka and W. E. Eder, *Theory of Technical Systems: A Total Concept Theory for Engineering Design*, vol. 53, no. 9. 1988.

[3] S. L. Konda, I. Monarch, P. Sargent, and E. Subrahmanian, "Shared Memory: A Unifying Theme for Design Research and Practice," *Res. Eng. Des.*, vol. 4, no. 1, pp. 23–42, 1992.

[4] H. Yoshikawa and K. Uehara, "Design Theory for CAD/CAM Integration," *CIRP Ann.*, vol. 34, no. 1, pp. 173–178, 1985.

[5] N. P. Suh, "Axiomatic Design Theory for Systems," *Res. Eng. Des.*, vol. 10, no. 4, pp. 189–209, 1998.

[6] D. Braha and Y. Reich, "Topological structures for modeling engineering design processes."

[7] O. Shai and Y. Reich, "Infused design. I. Theory," *Res. Eng. Des.*, vol. 15, no. 2, pp. 93–107, 2004.

[8] A. Hatchuel and B. Weil, "C-K design theory: An advanced formulation," *Res. Eng. Des.*, vol. 19, no. 4, pp. 181–192, 2009.

[9] The Open Group, *The Open Group Architecture Framework (TOGAF) version 9.1*. 2009.

[10] E. Crawley, B. Cameron, and D. Selva, *System architecture: strategy and product development for complex systems*. 2015.

[11] O. de Weck, J. Agte, J. Sobieszczanski-Sobieski, P. Arendsen, A. Morris, and M. Spieck, "State-of-the-Art and Future Trends in Multidisciplinary Design Optimization," 2012.

[12] G. Pahl, W. Beitz, J. Feldhusen, and K. H. Grote, *Engineering Design, a Systematic Approach*. 2007.

[13] M. R. Vladu, G. Dobre, and R. F. Mirica, *On Evaluation Methods in Product Development. Application on Mechanical Transmissions*, vol. 13. Dordrecht: Springer Netherlands, 2013.

[14] M. R. Vladu, G. Dobre, and R. F. Mirică, "Application of the fuzzy logic in the evaluation of solution variants in the product development process," *UPB Sci. Bull. Ser. C Electr. Eng.*, vol. 75, no. 1, pp. 207–222, 2013.

[15] L. A. Zadeh, "Fuzzy Sets-Information and Control-1965," *Inf. Control*, 1965.

[16] F. Ashby and D. Ennis, "Similarity measures," *Scholarpedia*, vol. 2, no. 12, p. 4116,

2007.

[17] R. Henkel, R. Hoehndorf, T. Kacprowski, C. Knu, W. Liebermeister, and D. Waltemath, “Notions of similarity for systems biology models,” vol. 19, no. May 2016, pp. 77–88, 2018.

[18] S. A. Elavarasi, J. Akilandeswari, and K. Menaga, “A Survey on Semantic Similarity Measure,” vol. 2, no. 3, pp. 389–398, 2014.

[19] S. Choi, S. Cha, and C. C. Tappert, “A Survey of Binary Similarity and Distance Measures,” vol. 0.

[20] K. Tamura, G. Stecher, D. Peterson, A. Filipski, and S. Kumar, “MEGA6: Molecular evolutionary genetics analysis version 6.0,” *Mol. Biol. Evol.*, 2013.

[21] Cavalli-Sforza L. L. and Edwards A. W. F., “Phylogenetic Analysis: Models and Estimation Procedures,” *Evolution (N. Y).*, 1967.

[22] P. O. Lewis, S. Kumar, K. Tamura, and M. Nei, “MEGA: Molecular Evolutionary Genetics Analysis, Version 1.02.,” *Syst. Biol.*, 2006.

[23] J. Reynolds, B. S. Weir, and C. C. Cockerham, “Estimation of the coancestry coefficient: Basis for a short-term genetic distance,” *Genetics*, 1983.

[24] D. J. States, W. Gish, and S. F. Altschul, “Improved sensitivity of nucleic acid database searches using application-specific scoring matrices,” *Methods*, 1991.

[25] D. Jurafsky and J. H. Martin, “Speech and Language Processing 18 BT - An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition,” in *An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2009.

[26] K. U. Schulz and S. Mihov, “Fast string correction with Levenshtein automata,” *Int. J. Doc. Anal. Recognit.*, 2003.

[27] G. Navarro, “A guided tour to approximate string matching,” *ACM Comput. Surv.*, 2002.

[28] D. Kellari, “What ’ s next for the airliner ? Historical Analysis and Future Predictions of Aircraft Architecture and Performance,” MIT, 2016.

[29] R. Henkel, R. Hoehndorf, T. Kacprowski, C. Knüpfer, W. Liebermeister, and D. Waltemath, “Notions of similarity for systems biology models,” *Brief. Bioinform.*, vol. 19, no. 1, pp. 77–88, 2018.

[30] P. Jaccard, “Étude comparative de la distribution florale dans une portion des Alpes et des Jura,” *Bull. Soc. Vaudoise Sci. Nat*, 1901.

[31] S. . Forbes, “On the local distribution of certain Illinois fishes. An essay in statistical ecology,” *Bull. Illinois State Lab. Nat. Hist.*, 1907.

[32] S. . Forbes, “Method of determining and measuring the associative relations of species,” *Science (80-.).*, no. 61, p. 524, 1925.

[33] P. Willett, J. M. Barnard, and G. M. Downs, “Chemical similarity searching,” *J. Chem. Inf. Comput. Sci.*, 1998.

[34] J. R. Smith and Shih-Fu Chang, “Automated binary texture feature sets for image retrieval,” 2002.

[35] H. . Driver and A. L. Kroeber, “Quantitative Expression of Cultural Relationships,” *Univ. Calif. Press*, 1932.

[36] R. R. Sokal and P. H. A. Sneath, *Principles of Numerical Taxonomy*. 1963.

[37] Z. Hubalek, “Coefficients of association and similarity, based on binary (presence-absence) data: an evaluation.,” *Biol. Rev.*, 1982.

[38] E. L. Michael, “Marine Ecology and the Coefficient of Association: A Plea in Behalf of Quantitative Biology,” *J. Ecol.*, 2006.

[39] P. Willett, “Similarity-based virtual screening using 2D fingerprints,” *Drug Discovery Today*. 2006.

[40] S.-H. Cha, C. Tappert, and S. Yoon, “Enhancing Binary Feature Vector Similarity Measures,” *J. Pattern Recognit. Res.*, 2013.

[41] S. H. Cha and C. C. Tappert, “Optimizing binary feature vector similarity measure using genetic algorithm and handwritten character recognition,” in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2003.

[42] R. M. Cormack, “A Review of Classification,” *J. R. Stat. Soc. Ser. A*, 2006.

[43] L. A. Goodman and W. H. Kruskal, “Measures of Association for Cross Classifications*,” *J. Am. Stat. Assoc.*, 1954.

[44] L. A. Goodman and W. H. Kruskal, “Measures of Association for Cross Classifications III: Approximate Sampling Theory,” *J. Am. Stat. Assoc.*, 1963.

[45] L. A. Goodman and W. H. Kruskal, “Measures of Association for Cross Classifications. II: Further Discussion and References,” *J. Am. Stat. Assoc.*, 1959.

[46] K. M. Shaw, G. Dunn, and B. S. Everitt, “An Introduction to Mathematical Taxonomy.,” *J. R. Stat. Soc. Ser. A*, 2006.

[47] M. Murguia and J. L. Villasenor, “Estimating the effect of the similarity coefficient and the cluster algorithm on biogeographic classifications,” *Ann. Bot. Fenn.*, 2003.

[48] R. E. Tulloss, “Assessement of similarity indices for undesirable properties and a new tripartite similarity index based on cost functions,” *North*, 1997.

[49] F. Saïs and R. Thomopoulos, "A reconciliation-driven approach of case-based prediction : state of the art , method overview and application in food science To cite this version : HAL Id : hal-01605378 CASE - BASED PREDICTION : S TATE OF THE ART ," 2016.

[50] GOM Space, "NanoCom ANT-6F UHF - Data sheet." .

[51] M. Hallikainen *et al.*, "AALTO-1 – AN EXPERIMENTAL NANOSATELLITE FOR HYPERSPECTRAL REMOTE SENSING Aalto University , School of Electrical Engineering , Department of Radio Science and Engineering VTT Technical Research Centre of Finland , P . O . Box 1000 , FI-02044 VTT , Finland," pp. 4367–4370, 2011.

[52] ISIS, "ISIS Antenna systems - Technical Datasheet." 2010.

[53] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets Syst.*, 1978.

[54] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man. Mach. Stud.*, 1975.

[55] D. Selva, B. Cameron, and E. Crawley, "Patterns in System Architecture Decisions," *Syst. Eng.*, vol. 19, no. 6, pp. 477–497, Nov. 2016.

[56] D. Kellari, E. F. Crawley, B. G. Cameron, S. Researcher, and M. Avenue, "Architectural decisions in commercial aircraft from the DC3 to the 787."

APPENDICES

A: Similarity measures

Table A1 - Normalized similarity and correlation measures

Name	Similarity Formula
Jaccard	$\frac{a}{n-d} = \frac{a}{a+b+c}$
Russel & Rao	$\frac{a}{n}$
Rogers & Tanimoto	$\frac{a+d}{a+2(b+c)+d}$
Kulczynski #1	$\frac{a}{b+c}$
Kulczynski #2	$0.5 \times (\frac{a}{a+b} + \frac{a}{a+c})$
Dice	$\frac{2a}{2a+b+c}$
Pearson's Phi coefficient	$\frac{(a \times d) - (c \times b)}{\sqrt{((a+c)(c+d)(a+b)(b+d))}}$
Baroni-Urbani/Buser	$\frac{a+\sqrt{(a*d)}}{a+b+c+\sqrt{(a*d)}}$
Braun-Blanquet	$if\ (a+b) > (a+c)\ then\ S := \frac{a}{a+b};$ $else\ S: = \frac{a}{a+c}$
Simpson similarity coefficient	$if\ (a+b) < (a+c) then\ S := \frac{a}{a+b};$ $else\ S := \frac{a}{a+c}$
Michael	$4 \times \frac{a \times d - b \times c}{(a+d)(a+d)+(b+c)(b+c)}$
Sokal and Sneath #1	$\frac{a}{a+2(b+c)}$
Sokal and Sneath #2	$0.25 \times (\frac{a}{a+b} + \frac{a}{a+c} + \frac{d}{b+d} + \frac{d}{c+d})$
Sokal and Sneath #3	$\frac{a \times d}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$
Sokal and Sneath #4	$\frac{a+d}{b+c}$
Sokal and Sneath #5	$\frac{2(a+d)}{2(a+d)+b+c}$
Simple Matching	$\frac{a+d}{a+b+c+d} = \frac{a+d}{n}$
Mean Hamming	$1 - D$

Sneath & Sokal	$\frac{a+d}{a\ +0.5(b+c)+d}$
Kocher & Wong	$\frac{a\times n}{(a+b)(c+d)}$
Faith	$\frac{a+\frac{d}{2}}{n}$
Ochiaï #1	$\frac{a}{\sqrt{(a+b)*(a+c)}}$
Ochiaï #2	$\frac{a\times d}{\sqrt{(a+b)*(a+c)*(d+b)*(d+c)}}$
Q_0	$\frac{b*c}{a*d}$
Yule's Sigma	$\frac{\sqrt{(a*d)}-\sqrt{(b*c)}}{\sqrt{(a*d)}+\sqrt{(b*c)}}$
Yule's Q	$\frac{a\times d-b\times c}{a\times d+b\times c}$
Upholt	$F\ =\frac{2a}{2*a+b+c}$ $S\ =\ 0.5\left(-F+\sqrt{F*F+8*F}\right)^{\frac{1}{n}};$
Excoffier	$n*(1-(a/n))$
Hamann	$(a-(b+c)+d)/n$
Roux #1	$\frac{a+d}{min(b,c)+min(n-b,n-c)}$
Roux #2	$\frac{n-a\times d}{\sqrt{(a+b)(c+d)(a+c)(b+d)}}$
Michelet	$\frac{a^2}{b\times c}$
Fager & McGowan	$\frac{a}{\sqrt{(a+b)*(a+c)}}+\frac{1}{\sqrt{(a+b)}}$
Fager	$\frac{a}{\sqrt{(a+b)*(a+c)}}-\ max(b,c)$
Unigram subtuples	$Log\left(\frac{a\times d}{b/c}\right)-3.29\times\sqrt{\frac{1}{a}+\frac{1}{b}+\frac{1}{c}+\frac{1}{d}}$
U cost	$Log\left(1+\frac{min(b,c)+a}{max(b,c)+a}\right)$
S cost	$Log^{-0.5}\left(1+\frac{min(b,c)}{a+1}\right)$
R cost	$Log\left(1+\frac{a}{a+b}\right)\times log\left(1+\frac{a}{a+c}\right)$

T combined cost	$Sqrt(U * S * R)$ $U = Log\left(1 + \frac{min(b,c) + a}{max(b,c) + a}\right)$ $S = Log^{-0.5}\left(1 + \frac{min(b,c)}{a+1}\right)$ $R = Log\left(1 + \frac{a}{a+b}\right) \times log\left(1 + \frac{a}{a+c}\right)$
McConnoughy	$\frac{a \times a - b \times c}{\sqrt{(a+b)(a+c)}}$
Phi Square	$\frac{(a * d + b * c)^2}{(a+b)(a+c)(b+c)(b+d)}$
Forbes	$\frac{n \times a}{(a+b)(a+c)}$
Fossum	$\frac{n(a-0.5)(a-0.5)}{(a+b)(a+c)}$
Stiles	$\log_{10}\left(\frac{n\left(\lvert a \times d - b \times c\rvert - \frac{n}{2}\right)^2}{(a+b)(a+c)(b+d)(c+d)}\right)$
Dispersion	$\frac{a \times d - b \times c}{(a+b+c+d)^2}$
Dennis	$\frac{a \times d - b \times c}{\sqrt{n(a+b)(a+c)}}$
Pearson Chi Square	$\frac{n\left(abs(a * d - b * c) - \frac{n}{2}\right)^2}{(a+b)(c+d)(a+c)(b+d)}$
Mountford	$\frac{2a}{2b \times c + a \times b + a \times c}$
Mutual Information	$ln\left(\frac{a \times n}{(a+b)(a+c)}\right)$
Weighted Mutual Information #3	$ln\left(a^3 \times \frac{n}{(a+b)(a+c)}\right)$
Chi Square with correction of Yates n/2	$\frac{n * \left(abs(a * d - b * c) - \frac{n}{2}\right)^2}{(a+b)(c+d)(a+c)(b+d)}$
Normalized Collacation	$\frac{a}{b+c-a}$
Dunning	$2 * (a * log(a) + b * log(b) + c * log(c) + d * log(d))$ $- (a+b) * log(a+b) - (a+c) * log(a+c) - (b+d) * log(b+d) - (c+d) * log(c+d)$

	$+ (a + b + c + d) * log(a + b + c + d)$

Distance measures

Table A2 - distance measures

Name	Distance formula
Hamming distance	$b + c$
Euclidian distance	$\sqrt{b + c}$
Soergel distance	$\frac{b + c}{b + c + d}$
Mean Hamming distance	$(b + c)/n = \frac{b+c}{a+b+c+d}$
Mean Euclidian distance	$\sqrt{\frac{b + c}{n}} = \sqrt{\frac{b + c}{a + b + c + d}}$

B: Case Study

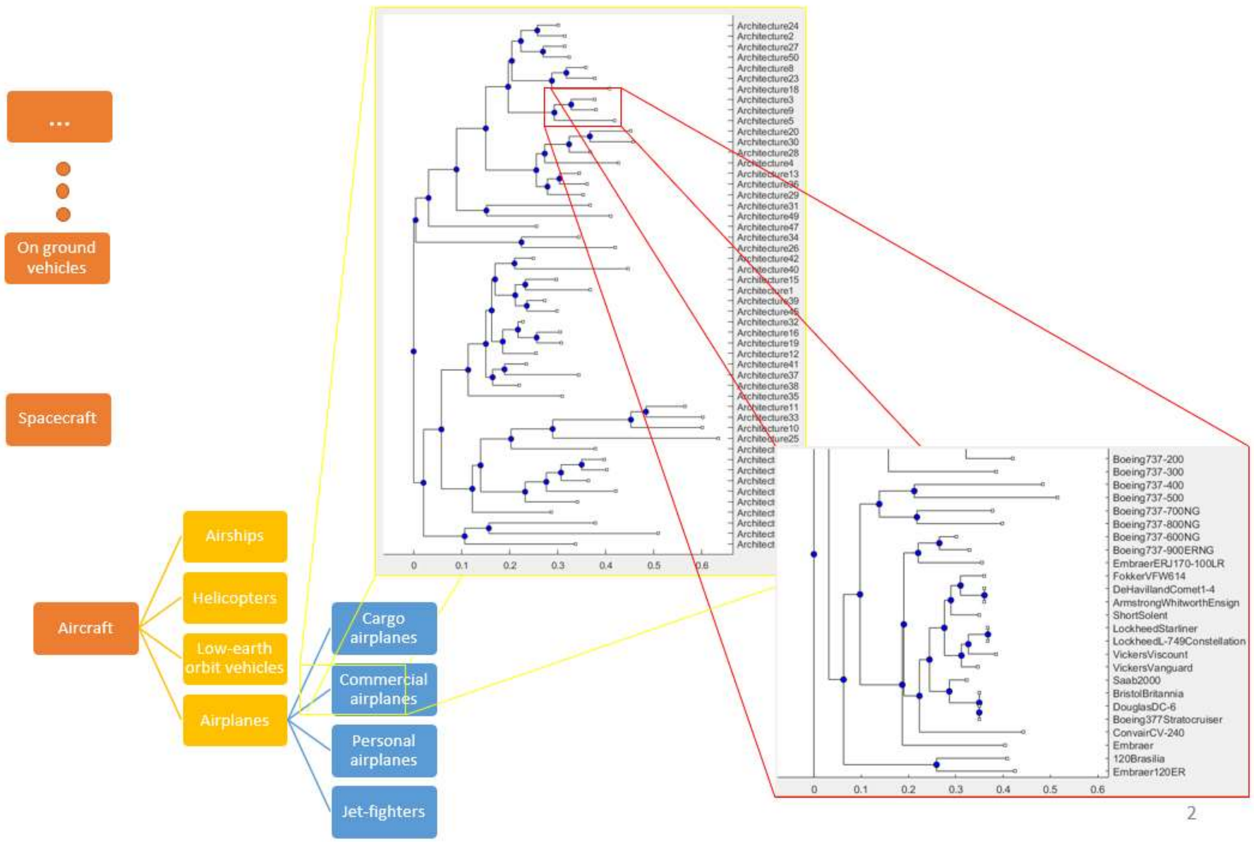


Figure B126. Cataloguing Design concepts (i.e., Architectural hierarchies)

C: The survey

Start of Block: Start

Q1.1 This survey is part of a research project that aims to understand the concept of similarity in design. Your answers are very important to the study, so take your time to answer the questions carefully. <u>All answers are anonymous</u> *and will only be used for this research purposes. The survey will take* ***<u>10-15 minutes</u>*** *of your time.*
The next section is a short introduction. Please read it carefully before moving to the questions.

End of Block: Start

Start of Block: Introduction

Q2.1 In this questionnaire, we are going to be asking you to assess some aspects of different designs of products and services. We will introduce designs that have the **SAME function** but have **DIFFERENT forms.** But what are form and function? **Read next!**
We know that different disciplines may use different terms to describe similar concepts, so before we get started, we want to define the key terms we use in this survey.
The **function** *is the activity, operation, or transformation that causes or contributes to performance*.
The **form** *is the physical or informational embodiment of a system that exists or has the potential for stable, unconditional, existing for some period of time and is instrumental in the execution of the function.*
The **concept/design** *is a product or system vision, idea, notion, or mental image that maps function to form.*
Your task: For each set, you will be asked to assess the importance of some design aspects and compare different designs. When doing these exercises, consider all the factors you think are important, but DO NOT CONSIDER THE COST.

End of Block: Introduction

Start of Block: Group I: Chairs

Q3.1 This set of questions asks you about **chair designs**. By chairs here, we refer to unpowered chairs for routine daily use (Does not include: Electrical wheelchairs, dental chairs, etc.)

Q3.2 **How familiar** are you with **chairs**?

○ Expert knowledge (I study and/or develop them) (1)

○ Average knowledge (I use them) (2)

○ Limited knowledge (not a user or a developer) (3)

○ No knowledge at all. [You can proceed to the next set of questions] (4)

Q3.3 When comparing chair designs, **how important** are each of the following elements of form **for delivering the overall function: seating the user**?

	Extremely important (1)	Very important (2)	Moderately important (3)	Slightly important (4)	Not at all important (5)
Bottom support (i.e., chair seat) (1)	○	○	○	○	○
Back support (i.e., chair back) (2)	○	○	○	○	○
Chair cover (i.e., chair padding) (3)	○	○	○	○	○

Q3.4 Now we are going to ask you some questions around comparing the following chair designs:

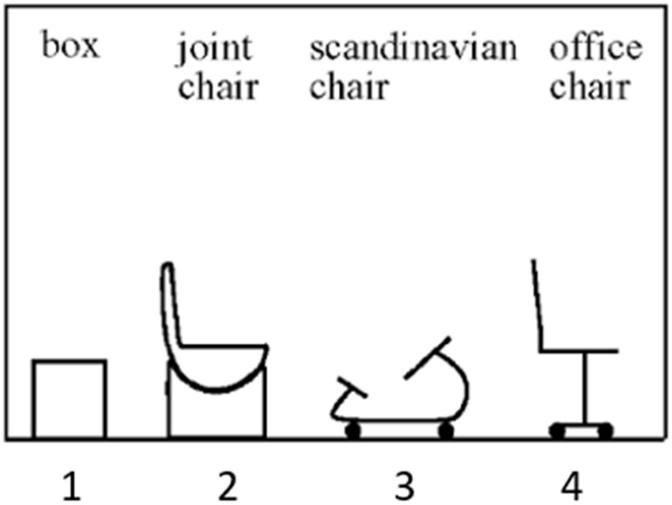
1- A box chair,

2- A joint chair,

3- A Scandinavian chair,

4- An office chair,

See the figure below for illustrative sketches



Q3.5

Q3.6 Please indicate how **similar** you think the designs are for each pair of **chairs**.

	Extremel y similar (1)	Ver y similar (2)	Moderatel y similar (3)	Slightl y similar (4)	No t at all similar (5)
Box and joint chair (1)					
Box and Scandinavian chair (2)					
Box and office chair (3)					
Joint chair and Scandinavian chair (4)					
Joint chair and office chair (5)					
Scandinavia n chair and office chair (6)					

End of Block: Group I: Chairs

Start of Block: Product groups

Q4.1 In this set of questions, we are going to ask you about **wine openers** – devices that you would use to open a bottle of wine that is sealed with a cork.

Q4.2 **How familiar** are you with **wine openers?**

○ Expert knowledge (I study and/or develop them) (1)

○ Average knowledge (I use them) (2)

○ Limited knowledge (not a user or a developer) (3)

○ No knowledge at all. [You can proceed to the next set of questions] (4)

Q4.3 When comparing wine opener designs, **how important** are each of the following elements of form **for delivering the overall function: removing the cork**?

	Extremely important (1)	Very important (2)	Moderately important (3)	Slightly important (4)	Not at all important (5)
The component that engages with the cork (e.g., screw) (1)	○	○	○	○	○
The component that moves the cork (e.g., lever) (2)	○	○	○	○	○
The handle (i.e., for hand grip) (3)	○	○	○	○	○

Q4.4 Now we are going to ask you some questions around comparing the following wine opener designs:1- An air injection wine opener

2- A lever corkscrew3- A tabletop corkscrew

4- An ah-so wine opener

See the figure below for illustrative pictures of these kinds of wine opener designs.

Q4

Q4.6 Please indicate **how similar** you think the designs are for each pair of **wine openers**.

	Extremely similar (1)	Very similar (2)	Moderately similar (3)	Slightly similar (4)	Not at all similar (5)
Air injection wine opener and lever corkscrew (1)	○	○	○	○	○
Air injection wine opener and tabletop wine opener (2)	○	○	○	○	○
Air injection wine opener and ah-so wine opener (3)	○	○	○	○	○
Lever corkscrew and tabletop wine opener (4)	○	○	○	○	○
Lever corkscrew and ah-so wine opener (5)	○	○	○	○	○
Tabletop wine opener and ah-so wine opener (6)	○	○	○	○	○

End of Block: Product groups

Start of Block: Musical_instruments

Q5.1 Now in this set of questions, we are going to ask you about **musical instruments.** In principle, any object that produces sound can be considered a musical instrument. However, by musical instruments here we mean musical instruments that are used for composing melodies. These can be chordophones, percussion instruments, electrophones or aerophones. Some examples are guitars, music boxes, electric guitars, flutes, violins, and drums.

Q5.2 **How familiar** are you with **musical instruments**?

- Expert knowledge (I study and/or develop them) (1)
- Average knowledge (I use them) (2)
- Limited knowledge (not a user or a developer) (3)
- No knowledge at all. [You can proceed to the next set of questions] (4)

Q5.3 When comparing musical instruments designs, **how important** are each of the following elements of form **for delivering the main function: composing melody**?

	Extremely important (1)	Very important (2)	Moderately important (3)	Slightly important (4)	Not at all important (5)
The component that creates the sound (e.g., guitar's strings) (1)					
The component that amplifies the sounds (e.g., guitar's body) (2)					
The component used for tuning the sound (e.g., guitar pegs) (3)					

Q5.4 Now we are going to ask you some questions around comparing the following musical instruments designs:1- A trumpet2- A violin3- An electric guitar4- A piano See the figure below for illustrative pictures.

Q5.5

Q5.6 Please indicate **how similar** you think the designs are for each pair of **musical instruments**.

	Extremel y similar (1)	Ver y similar (2)	Moderatel y similar (3)	Slightl y similar (4)	No t at all similar (5)
Trumpe t and violin (1)					
Trumpe t and electric guitar (2)					
Trumpe t and piano (3)					
Violin and electric guitar (4)					
Violin and piano (5)					
Electric guitar and piano (6)					

End of Block: Musical_instruments

Start of Block: Means of transportation

Q6.1 Now in this set of questions, we are going to ask you about means of transportation**. By transportation means here, we mean all manner of air, ground and water transport which are used for transporting passengers. Some examples are planes, helicopters, airships, gliders, cars, buses, trains, subways, ferries, bicycles, boats, ships, cruise ships, and submarines**

Q6.2 **How familiar** are you with **means of transportation**?

○ Expert knowledge (I study and/or develop them) (1)

○ Average knowledge (I use them) (2)

○ Limited knowledge (not a user or a developer) (3)

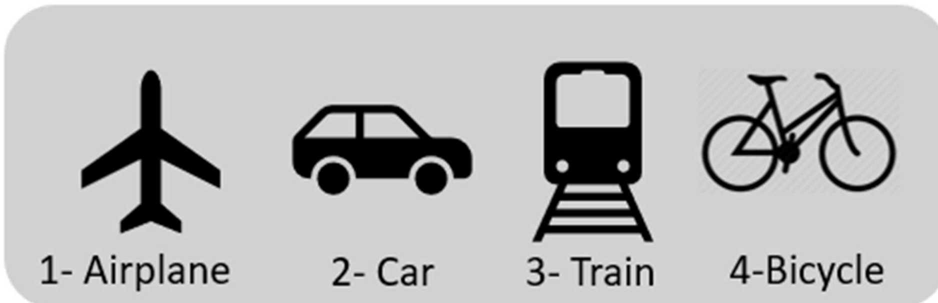
○ No knowledge at all. [You can proceed to the next set of questions] (4)

Q6.3 When comparing means of transportation designs, how important are each of the following elements of form for delivering the overall function: transporting passengers?

	Extreme ly important (1)	Ver y important (2)	Moderate ly important (3)	Slightl y important (4)	N ot at all important (5)
Moving systems (e.g., engine, propeller) (1)	○	○	○	○	○
Containing systems (e.g., body, fuselage) (2)	○	○	○	○	○
Entertainme nt systems (e.g., radio, television) (3)	○	○	○	○	○

Q6.4 Now we are going to ask you some questions around comparing several designs of transportation means that all have the same function of transporting passengers:1- An airplane2- A car3- A train4- A bicycle See the figure below for illustrative sketches.

Q6.5



Q6.6 Please indicate **how similar** you think the designs are for each pair of **transportation means**.

	Extremel y similar (1)	Ver y similar (2)	Moderatel y similar (3)	Slightl y similar (4)	No t at all similar (5)
Airplan e and car (1)	○	○	○	○	○
Airplan e and train (2)	○	○	○	○	○
Airplan e and bicycle (3)	○	○	○	○	○
Car and train (4)	○	○	○	○	○
Car and bicycle (5)	○	○	○	○	○
Train and bicycle (6)	○	○	○	○	○

End of Block: Means of transportation

Start of Block: Wrap-up

Q7.1 For each of the four sets of products you were asked to compare, there was a set of questions asking you how similar one product was to another – for example, how similar a box chair was to an office chair, how similar a lever corkscrew was to an ah-so wine opener, how similar a plastic bottle was to a glass cup, and how similar a car was to a train. **In general, what was your approach or thought process in determining the extent of similarity between designs?**

____

Q7.2 Please choose all the features that you think are important when comparing any designs.

- [ ] Elements of form (i.e., components and parts) (1)
- [ ] Functions (2)
- [ ] Performance (3)
- [ ] Complexity (5)
- [ ] Shape (6)
- [ ] Color (7)

Q7.3 To what extent do you agree or disagree with the following statements: **I understood the tasks I was supposed to do in the survey.**

○ Strongly agree (9)

○ Agree (8)

○ Neither agree nor disagree (7)

○ Disagree (6)

○ Strongly disagree (3)

Q7.4 What is your occupation? (select all that apply)

▢ Student (1)

▢ Researcher (2)

▢ Engineer (3)

▢ Designer (4)

▢ Other (please specify): (5) ____

Q7.5 Where do you study/work?

○ MIT (1)

○ Skoltech (2)

○ MIPT (3)

○ Other (please specify) (4) ____

Q7.6 We encourage you to write any **additional comments** you may have for us.

____

End of Block: Wrap-up