Skolkovo Institute of Science and Technology

MODEL-BASED FRAMEWORK FOR SYSTEM CONCEPT

Doctoral Thesis

by

YAROSLAV MENSHENIN

DOCTORAL PROGRAM IN ENGINEERING SYSTEMS

Supervisor
Professor Edward Crawley

Moscow – 2020

I hereby declare that the work presented in this thesis was carried out by myself at Skolkovo Institute of Science and Technology, Moscow, except where due acknowledgement is made, and has not been submitted for any other degree.

Candidate (Yaroslav Menshenin)
Supervisor (Prof. Edward Crawley)

"Continuous effort – not strength or intelligence – is the key to unlocking our potential" –

Sir Winston Churchill

Abstract

The development of a concept for a system is a key step towards creating the system's architecture. Most previous concept development approaches focus on the procedures for conceptual design activity – the sequence of activities and tasks. This dissertation is motivated by the desire to elaborate in details the notional content of a system concept and to provide the means of encoding and analyzing space systems concepts in a digital environment. In this Thesis a model-based *system concept representation framework* is developed and presented. Such a framework answers both questions – "what" should be specified in concept during conceptual design phase, and "how" this should be done. Such framework utilizes such conceptual modeling languages as Object-Process Methodology (OPM) and Systems Modeling Language (SysML).

Throughout this dissertation the Design Research Methodology is used. The validation of the proposed framework is performed through the application of framework to the analytical surveys: patents, urban architectural patterns, and software patterns. We argue that the proposed framework could digitally represent the core essence of the analytical surveys. We also demonstrate that by using modeling syntax and ontology conveying a strictly defined meaning instead of texts in natural language, we can represent the same conceptual information in much less data volume.

The utility of the proposed framework is demonstrated on two space-related case studies: commercial suborbital human spaceflight systems (Virgin Galactic, Blue Origin, and XCOR projects) and space communication missions (TDRSS, EDRS, and NEN projects).

For every project of each case study the model-based system concept representation framework is built on different levels of granularity. One of the forms of utility of the proposed framework is that it provides the means to *encode* the unstructured data contained in a publicly available descriptions of space projects into a structured set of both texts and models, which are consistent with each other. This would allow systems engineer to have a concept classification scheme and searchable database, documenting core information about a concept. This approach could be used within the INCOSE's Model-Based Conceptual Design initiative.

Another form of utility is that the proposed framework supports concept *generation* at early phases of the design process in a model-based environment. For example, it allows systematic combination of entries to produce new concepts. The proposed framework contains 28 entries and a modeling ontology. By extending and re-combining them, the systems engineer can develop novel concept, revealing alternative and previously unexplored concepts in a concise way.

Additional utility of framework is that it enables a formal analysis; such as it provides a methodology for system engineer to *measure* the conceptual similarities between alternative concepts.

Finally, another additional form of utility is that the concept knowledge is *reused* in later stages of the design process – during the architecture development.

Publications

1. Menshenin, Y. and Crawley, E., 2020. A System Concept Representation Framework and its Testing on Patents, Urban Architectural Patterns, and Software Patterns. *In Systems Engineering Journal* https://doi.org/10.1002/sys.21547

2. Menshenin, Y., Knoll, D., Brovar, Y. and Fortin, C., 2020. Analysis of MBSE/PLM Integration: From Conceptual Design to Detailed Design. *In Proceedings of IFIP 17th International Conference on Product Lifecycle*, Rapperswil, Switzerland (Virtual event)

3. Menshenin, Y., Brovar, Y., Crawley, E., and Fortin, C., 2020. Analysis of Systems Structural Relationships Through a DSM-Based Approach. *In Proceedings of the 22nd International DSM Conference*, Cambridge, MA, USA (Online conference)

4. Menshenin, Y., Knoll, D., Brovar, Y., Crawley, E., and Fortin, C., 2020. Digital Engineering Information Exchange Model for Space Missions Architecture: a Case Study of a CubeSat Mission. *In Proceedings of 71st International Astronautical Congress*, Dubai, UAE (CyberSpace Edition) (accepted)

5. Menshenin, Y., Podladchikova, T., Fortin, C. and Crawley, E., 2020. A Holistic Approach to Space Sector Education: a Case Study of a Space Sector Course. *In Proceedings of 71st International Astronautical Congress*, Dubai, UAE (CyberSpace Edition) (accepted)

6. Menshenin, Y. and Crawley, E., 2019. Applying the Model-Based Concept Framework to Capturing the Safety Aspects of the Suborbital Human Spaceflight Missions. *In Journal of Space Safety Engineering*, 6, pp. 235-247 https://doi.org/10.1016/j.jsse.2019.10.008

7. Menshenin, Y., Latyshev, K., Mahfouz, A., Pritykin, D., 2019. Applying the Model-Based Conceptual Design Approach to ADCS Development: a Case Study of a CubeSat Swarm Mission. *In Proceedings of the 70th International Astronautical Congress*

8. Menshenin, Y. and Crawley, E., 2018. DSM-Based Methods to Represent Specialization Relationships in a Concept Framework. *In Proceedings of the 20th International DSM Conference*, pp. 151-157

9. Menshenin, Y. and Crawley, E., 2018. Model-based Concept Framework for Suborbital Human Spaceflight Missions. *In Proceedings of the 69th International Astronautical Congress*

10. Menshenin, Y. and Crawley, E., 2018. A Framework for Concept and its Testing on Patents. *In INCOSE International Symposium*, Vol. 28, No. 1, pp. 1564-1577. https://doi.org/10.1002/j.2334-5837.2018.00568.x

Conference Presentations

1. Menshenin, Y., 2019. Model-Based Systems Engineering – Ways of Thinking and Approaches. *Keynote Presentation at the Doctoral Workshop within 16th International Product Lifecycle Management Conference*, Moscow, Russia

2. Menshenin, Y. and Crawley, E., 2017. Concept Framework for Suborbital Human Spaceflight Systems in Concurrent Engineering Design Environment. *In 1st Airbus CTO Concurrent Design Facility Workshop (Poster Session)*, Toulouse, France

3. Menshenin, Y., 2017. Development of a Concept Framework for Complex Systems in Systems Engineering Paradigm. *In 5th International Spring School on Systems Engineering*, Enschede, Netherlands

4. Menshenin, Y. and Crawley, E., 2017. A Framework for the Genome of a Concept and its Testing on Space Communications Systems. *In Skoltech & MIT Conference "Shaping the Future: Big Data, Biomedicine and Frontier Technologies"*, Moscow, Russia

5. Menshenin, Y. and Crawley, E., 2016. How do We Create the Abstraction of the Concept and its Representation in the Framework of System Architecture Methodology in Concurrent Engineering Environment. *In 14th International DESIGN Conference (Poster Session at PhD Forum)*, Dubrovnik, Croatia

6. Menshenin, Y. and Crawley, E., 2015. Representation of Concept in Concurrent Engineering Facility. *In SkoltechOn Conference*, Moscow, Russia

Acknowledgements

The years of PhD studies were very fruitful, challenging, and wonderful. I jumped into the boat that had never swum where I swam on it with the only one intention: going all the way through it – from the beginning to the end of PhD, experiencing it by myself, doing it with full dedication and passion, and paving the way for the next generation of PhDs with whom we would collaborate in future – one way or another. Now I can say “it works”, because it is already happening.

There is a person who co-founded an entity called Skoltech in which I could start building my own way. This person is the Skoltech Founding President and my Supervisor Prof. Edward Crawley. Ed showed me how interesting the scientific life is, how many things can be done if you truly believe in what you are doing, and how important it is – to respect what you do. Ed’s students are a big family in a small world – we know each other and work with each other, and this is truly a result of Ed’s dedication, passion, and way of life.

I should say many thanks to the colleagues from Skoltech Space Center. I am grateful to Prof. Clement Fortin, with whom I've been working on a research project. I am truly thankful to Prof. Rupert Gerzer, with whom I had a number of important discussions. Prof. Alessandro Golkar was always nearby being ready to support. Prof. Anton Ivanov invited me to join him in delivering the number of lectures for Space Center’s partners. In general, the discussions with all of you and the joint research work made my PhD life a bit less stressful.

Special thanks should be sent to my colleagues and friends at MIT System Architecture Group – Markus Guerster, Iñigo del Portillo, Marc Sanchez Net, Bruce Cameron.

None of these would have been possible without a constant support, faith, and love of my wife Kate. When I started my PhD, I could not have imagined that by the date of graduation I will have already be married. And now I can add: that I will have already be a father, which has happened on July 7. I really hope that one day my daughter Alice would be interested at least opening this Thesis to get a sense of what her dad was working on at around this time.

I am grateful for the opportunity to dive deep into the world of science to make my own contribution to it. I am confident that this path has just started.

Table of Contents

List of Symbols, Abbreviations

CDF – Concurrent Design Facility

CEDL – Concurrent Engineering Design Laboratory

ConOps – Concept of Operations

DLR – German Aerospace Center

DMM – Domain Mapping Matrices

DNA – Deoxyribonucleic acid

DPCC – Devolved Payload Control Center

DSM – Design Structure Matrix

DRM – Design Research Methodology

EDRS – European data relay satellite

EM signal – electromagnetic signal

ESA – European Space Agency

ESTEC – European Space Research and Technology Centre

GF – Generic Form

HTHL – Horizontal Takeoff Horizontal Landing

IEoF – Internal element of Form

INCOSE – International Council on Systems Engineering

IO – Internal operand

IP – Internal process

ISL terminal –Inter-Satellite Link terminal

LCT – Laser communication terminal

MBCD – Model-based conceptual design

MBSE – Model-based systems engineering

MOC – Mission Operations Center

NASA – National Aeronautics and Space Administration

NEN – Near Earth Network

OISL terminal – Optical Inter-Satellite Link terminal

OMG – Object Management Group

OPD – Object Process Diagram

OPL – Object Process Language

OPM – Object Process Methodology

OV-1 – Operational Viewpoint-1

P2P transportation – Point-to-point transportation

SCaN program – Space Communications and Navigation program

SF – Specific Form

SNO – Solution-neutral operand

SNP – Solution-neutral process

Spacecom – Space Communications

SSO – Solution-specific operand

SSP – Solution-specific process

SysML – Systems Modeling Language

TDRS – Tracking and data relay satellite

UML – Unified Modeling Language

USPTO – United States Patent and Trademark Office

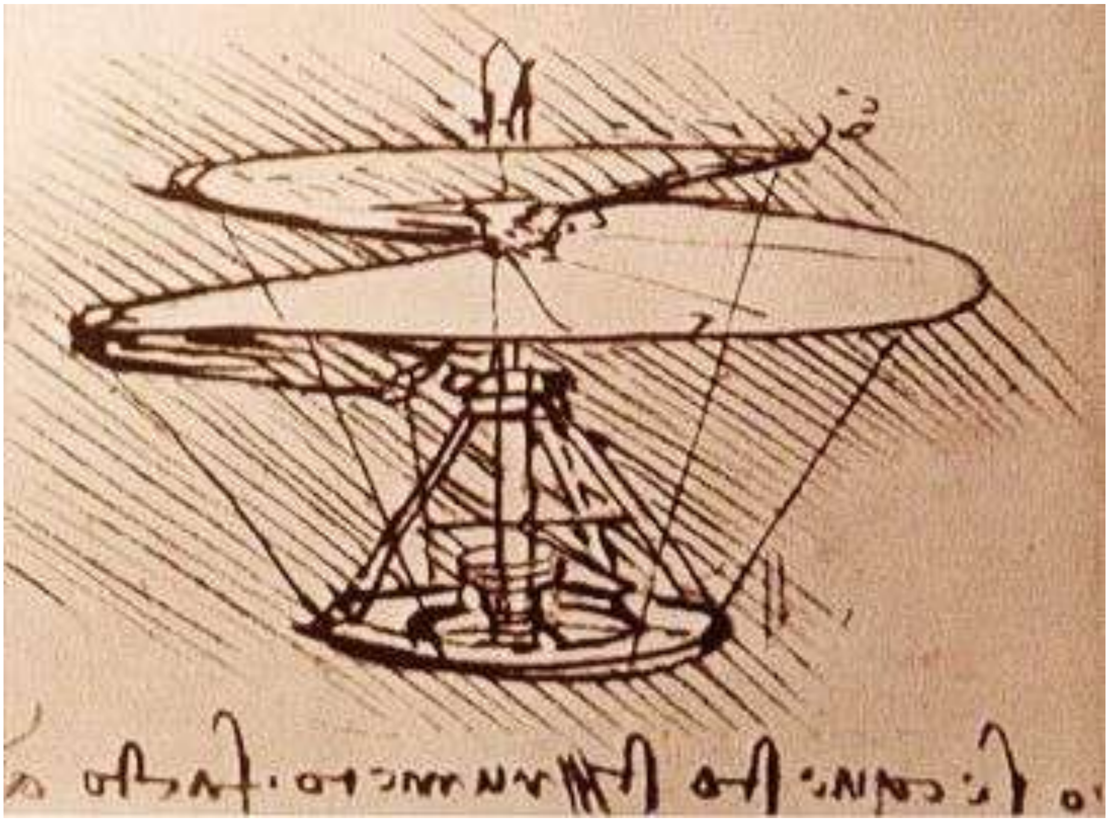
VTVL – Vertical takeoff, vertical landing

WIPO – World Intellectual Property Organization

List of Figures

List of Tables

Chapter 1. Introduction



1.1 Motivation for Digital Support of System Concept Representation Framework Development

Throughout human history, the design process was not only scientific adventure, but also an art. The design has the features that unite all disciplines. This can be perfectly illustrated by one of the symbols of Renaissance – Leonardo da Vinci. The scientific explorations of Leonardo da Vinci were extremely wide ranging as they included anatomy, astronomy, botany, chemistry, hydrodynamics, optics, physics, and zoology. Of particular relevance are da Vinci's contributions to geology, with specific regard to his studies on body fossils and sedimentary geology. Baucon sheds light on the obscure aspect of Leonardo's geological knowledge: his ichnological studies (Baucon 2010). This unique combination of interests and knowledge led to the appearance of the concepts that were almost impossible to imagine at the age of da Vinci. For example, that is how he explained the concept of helicopter: "I have discovered that a screw-shaped device such as this, if it is well-made from starched linen, will rise in the air if turned quickly".

Many inventions have been created since that time – from steam engine to iPhone. What unites these inventions is that they were *designed*. The times were different; the inventors could have different sex, age, nationality, convictions, and religious. But each one of these inventors put the efforts in the same process called *designing*. Each of them could follow their own understanding of the design process, or even not thinking about it at all.

The twentieth century was – in many dimensions – a dramatic century. These hundred years – a very short period of time – brought us two world wars, as well as the invention of potentially catastrophic and planet Earth-threatening weapons. On the other side, humanity had never prospered so fast as it was during the twentieth century. This led to the end of the second millennium, when computers, laptops, smartphones came to each house. Moreover, the same devices with advanced capabilities became available to large corporations, as well as universities, startups, and other organizations.

This created huge opportunities for the *design* itself. In the past the inventor's instruments were a piece of paper, a pencil and a library. The designer could put this on the paper and engage the imagination to develop the innovative solution, or to analyze the existing solution. The new era brought us computational and digital capabilities that can be used to facilitate the *design* process. This opens up an opportunity to engage a systematic approach on early phase of the design process facilitating a concept development at the beginning of product development process.

The development of a concept for a system is a key step towards creating the system's architecture. Most previous concept development approaches focus on the procedures for the conceptual design activity – the sequence of activities and tasks, such as in the generic product development process – see Ulrich and Eppinger (2007) and concept model of Pahl and Beitz (2013). In this dissertation a model-based *system concept representation framework* is developed and presented. Such a framework answers both questions – "what" should be specified in concept during conceptual design phase, and "how" this should be done.

Model-Based Conceptual Design (MBCD) is the "application of Model-Based Systems Engineering (MBSE) to the exploratory research and concept stages of the generic lifecycle" (MBCD WG). In this dissertation we develop and present a model-based *system concept representation framework* that *encodes* the core information about concept at conceptual design phase. This would allow systems engineer to have a concept classification scheme and searchable database, documenting core information about a concept. Thus, this approach could be used within the INCOSE's Model-Based Conceptual Design initiative. We validated the proposed framework through a variety of analytical surveys (patents, urban architectural patterns, and software patterns), and applied it to two space-related case studies: suborbital human spaceflight systems (Virgin Galactic, Blue Origin, and XCOR) and space communications systems (Tracking and

Data Relay Satellite Systems, European Data Relay Satellite System, and Near Earth Network System).

Figure 1.1: Airbus Pop.Up concept

Another utility of framework is that it supports concept *generation* at early phases of the design process in a model-based environment. For example, it allows systematic combination of entries to produce new concepts. The proposed framework contains 28 entries and a modeling ontology. By extending and re-combining them, the systems engineer can develop novel concept, revealing alternative and previously unexplored concepts in a concise way. The new innovative concepts often combine multiple functions as it is shown on example of Airbus Pop.Up concept demonstrated in Figure 1.1. This concept unites a number of functions, which were previously intrinsic to different systems: a car and a helicopter. However, thanks to a newly developed approach we see that both functions – flying and driving – could be put into one vehicle. Our dissertation proposes the systematic methods of how these possibilities could be predicted, explored, and analyzed.

Additional utility of framework is that it enables a formal analysis; such as it provides a methodology for the system engineer to *measure* the conceptual similarities between alternative concepts. Such assessment could be considered as a proxy of cost of change from one concept to another; or one sub-system to another one.

Finally, another additional form of utility is that the concept knowledge is *reused* in later stages of the design process – during the architecture development.

1.2 General Objective

There are few general objectives of this work.

The first general objective is to develop a *system concept representation framework* that can systematically represent the concept's constituents, their definitions and interconnections. Such a framework would support the design process during the conceptual design phase and would contribute to the INCOSE's Model-Based Conceptual Design Initiative.

The second objective of this work is to develop and present the ontology and semantics for the model-based framework. This would contain the core principles that should be followed at the conceptual design phase, as well as the definitions of concept framework's entries and criteria to include these entries into the framework. Having this as a tool, the multinational and multidisciplinary group of systems engineers, systems architects, designers, researchers, and managers could work together on the specific topic operating the same language – both natural and modeling.

The third general objective of this work is to demonstrate the utility of the proposed framework. To do so we have chosen the set of socio-technical systems and societal challenges (disclosed in set of analytical surveys – patents, urban architectural patterns, and software patterns); and purely technical systems (disclosed in two case studies – commercial suborbital human spaceflight systems and space communication missions).

1.3 Background and Literature Review

This dissertation integrates four pillars, which represent the respective field of knowledge and are demonstrated in Figure 1.2.

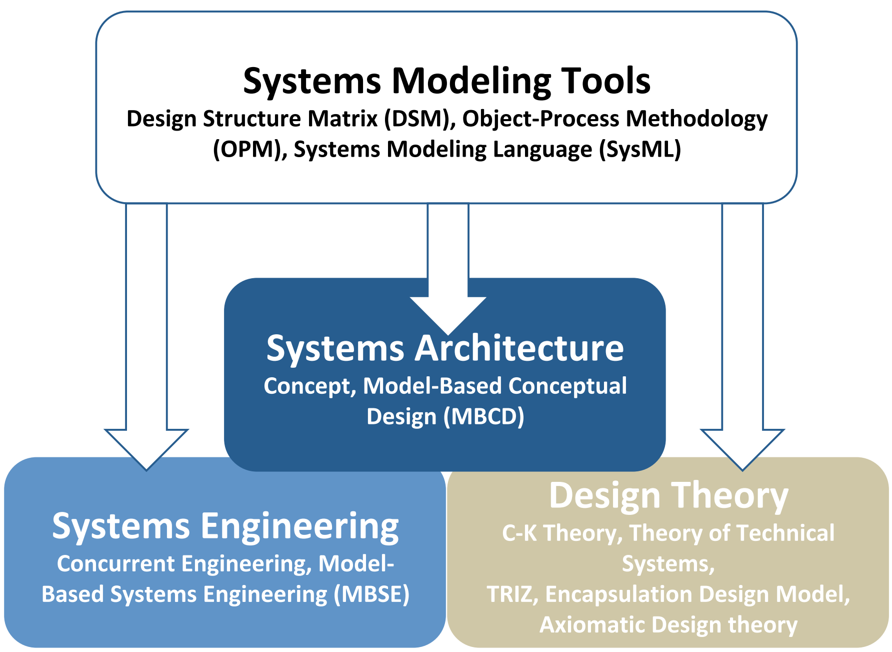


Figure 1.2: Four pillars of the dissertation

The first pillar is the Systems Engineering in which we are focusing on the Model-Based Systems Engineering (INCOSE 2015). We also explore the concurrent engineering (CE) approach, which allows reducing time spent on the complex systems design process with enhanced communications links between the design session team members (Prasad 1996; Bandecchi et al. 1999). This element is important, because the proposed in this dissertation methods and tools could facilitate the concept development in a Concurrent Engineering facility by *encoding* and *generating* the concept or the set of alternative concepts.

The second pillar, which we highlighted as the separate one is the Systems Architecture, in which the emphasis is made on concept itself and Model-Based Conceptual Design (INCOSE MBCD). In this part we are mainly focusing on the models and theories proposed in the works of Crawley et al. (2015), Ulrich and Eppinger (2016), Pahl and Bietz (2013), Andreasen et al. (2015). This pillar is essential for our work, because it explores deeply the notions of concept, conceptual design and their placement in the overall product development process. It also integrates the model-based representation (INCOSE MBCD) of conceptual design stage within the entire lifecycle (INCOSE 2015).

The third pillar is the Design Theory. Our Thesis is not trying to substitute the existing knowledge and approaches, rather to advance our understanding of conceptual design built upon them. Among design theories are the C-K theory (Hatchuel and Weil 2009), Theory of Technical Systems (Hubka 1973), TRIZ (Altshuller 1999), Encapsulation Design Model (Andreasen et al. 2015), Axiomatic Design Theory (Suh

1990; 1998). The design theories are important for our work, because they have the previously developed and proposed approaches, methods, and tools. Thus, by exploring them we are able to identify what scientific knowledge already exists, and which new frontiers are opening up thanks to new technologies and digital tools.

The fourth pillar in the dissertation is the Systems Modeling tools, represented by the Design Structure Matrix (Eppinger and Browning 2012) as the method of managing complex relationships among systems elements; the Object-Process Methodology (OPM) that allows to represent the systems using the Object-Process Diagrams (OPDs) and Object-Process Language (OPL) (Dori 2002); and the Systems Modeling Language (SysML) representing the systems using nine types of diagrams (Friedenthal et al. 2014). This part of the research is essential, as it provides the modeling tools that allow putting the document-based information into the digital representation. The conceptual modeling languages also propose the means for ontology, semantics, and syntax.

As it is shown in Figure 1.2 and as it is discussed above, these four pillars – being integrated to solve a common problem – form the research gap that could be fulfilled by the development of the model-based *system concept representation framework*. Such framework would specify how to *encode* the existing systems concepts, or to *generate* the new ones; enabling the formal analysis, such as *measuring* a similarity between alternative concepts; and *reusing* a concept knowledge in later stages of the design process.

1.4 Specific Objectives

One of the specific objectives of our work is to develop and present a model-based *system concept representation framework*, which supports conceptual design phase by *encoding* the conceptual information about multiple alternatives. We propose a framework that is based on five propositions rooted in systems engineering and design theory that lead to 28 entries in the framework. Thus, if the system engineer adopts the proposed approach, he or she will have a tool that digitally supports the design process at the conceptual design phase. This includes the detailed information about "what" should be specified for the conceptual design stage, and "how" this information can be encoded in a model-based manner. Such approach could potentially support the concurrent engineering design sessions.

The next specific objective is to demonstrate the utility of the proposed approach. For this we have conducted three studies for small-N analysis: mapped eight selected US patents, nine selected urban architectural patterns, and three selected software patterns to the framework; and three studies for the large-N analysis: mapped twenty five selected US patents, twenty seven selected urban architectural patterns, and twelve selected software patterns to the framework. Patents, urban architectural patterns, and software patterns represent a rich body of knowledge contained in them, and therefore they must logically contain a description of the concepts underlying them.

To demonstrate the utility of the proposed approach we applied it to two space-related case studies: commercial suborbital human spaceflight systems and space communication systems. For each project of each case study, we built a detailed concept model.

Another specific objective of this work is to demonstrate how this information can be used for quantitative assessment of formal conceptual similarity between alternative concepts. This includes the novel method, developed in this dissertation, that enhances the Design Structure Matrix (DSM) approach by means of keep tracking of both types of relationships – specialization and decomposition – in one matrix. In turn, this creates an opportunity for quantitative assessment of the conceptual similarity between alternative concepts.

1.5 Research questions and hypotheses

Our work is aiming at addressing the following research questions:

1) What information about system concept is required in order to have a representation of a concept?

2) How the information required for system concept representation can be encoded in a model-based manner to support system concepts and their alternatives development?

3) How the information encoded in *system concept representation framework* can support the quantitative assessment of formal conceptual similarity between alternative concepts?

Therefore, the hypothesis of our Thesis is that the proposed *system concept representation framework* contains a necessary information to describe the system concept. Another hypothesis of our work is that such information can be encoded in a model-based manner to represent system concepts and their alternatives in a digital environment. We also have a hypothesis that having such a framework supports design studies in terms of quantitative assessment of formal conceptual similarity between alternative concepts.

1.6 Overview of Thesis

In this work we use the Design Research Methodology (DRM) (Blessing and Chakrabarti 2009), presented in Figure 1.3.

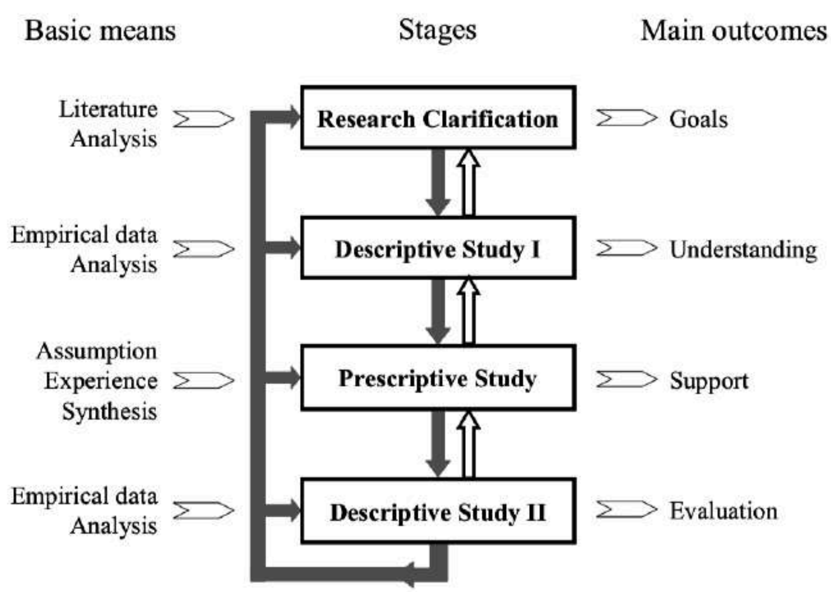
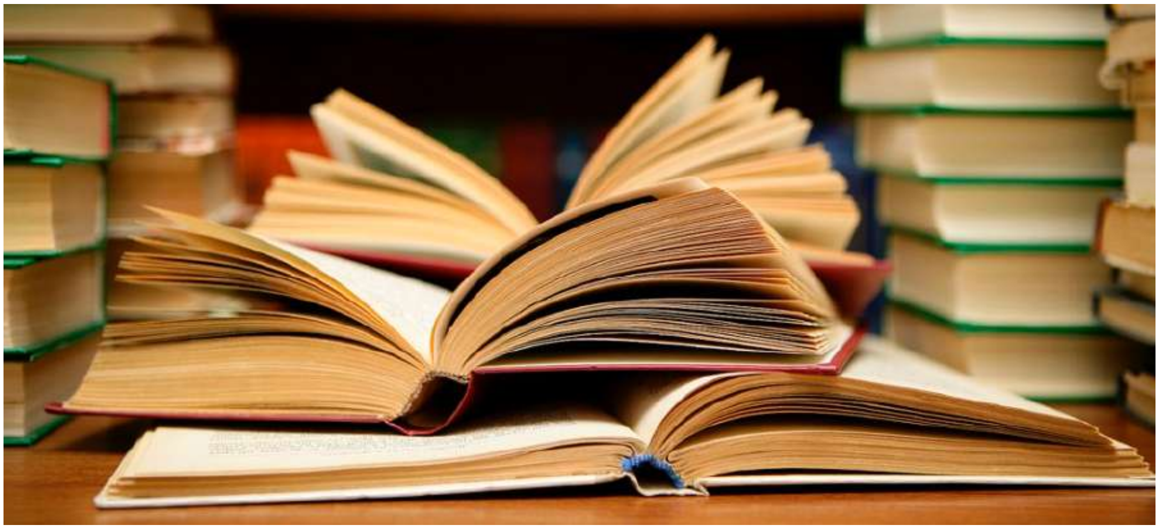


Figure 1.3: DRM framework (Blessing and Chakrabarti 2009)

The Thesis is organized as follows. Chapters 1 and 2 are dedicated to the Research Clarification stage, according to DRM (see Figure 1.3). In Chapter 2 we provide the overview of the concept in literature. We discuss the related theories that were developed in the past and which we can enhance by our work. This is the Descriptive Study I in the terminology of DRM. Chapter 3 has some elements of the

Descriptive Study I in terms of the literature review that supports the propositions development. However, the main purpose of Chapter 3 is to present the proposed model-based *system concept representation framework*. There we explain each one of the five propositions of the proposed framework, as well as all 28 entries. In Chapter 3, in parallel with providing the theoretical rationale for our proposal we demonstrate how this can be applied to the development of two alternatives of the aircraft concept: tube and wing aircraft concept, and blended wing body aircraft concept. We also demonstrate the applicability of the proposed approach for a small case, such as a coffee maker. Thus, the main role of Chapter 3, in DRM terminology, is to facilitate the Prescriptive Study. In the following Chapter 4 we validate the proposed methodology by means of analytical surveys, such as patents, urban architectural patterns, and software patterns. We conduct small-N analysis and large-N analysis for each of the analytical surveys to map the selected samples to the framework. Chapter 5 presents the first case study, aiming at demonstration of the utility of the proposed approach. In Chapter 5 we apply the model-based concept framework to the development of suborbital human spaceflight systems, such as Virgin Galactic, Blue Origin, and XCOR. For each one of these projects we build the models on different levels of granularity: at the first level of decomposition, and at the second level of decomposition. After that we demonstrate how this information, with some benefit provided by usage of the Design Structure Matrix approach, enables the quantitative measure of conceptual difference between competing alternative concepts. In Chapter 6 we demonstrate the utility of the proposed approach on example of space communications systems (TDRS system, EDRS system, and NEN system). We build the models for each concept on both levels of decomposition, after which we present a quantitative approach to measure conceptual similarity between alternative concepts. These three chapters (Chapter 4, Chapter 5, and Chapter 6) form the basis for the Descriptive Study II, according to the DRM framework (see Figure 1.3). We provide the conclusions in Chapter 7 summarizing the main outcomes of the work.

Chapter 2. Literature Review



2.1 Concept in Literature

To address research questions indicated in section 1.5, in Chapter 2 we focus on the systems engineering and design theories that touch the idea of concept (Le Masson et al. 2013). In particular, in Chapter 2 we explore these theories making an emphasis on the role of concept in them and how the specific theory proposes to capture and to represent a concept. This literature review paves the way to the discussion of the research questions of section 1.5 to be further and deeper explored in Chapter 3. After that we specify the research gaps that we aim at fulfilling in current Thesis, and explain how the proposed model-based *system concept representation framework* serves this purpose.

2.1.1 Theory of Technical Systems and its Successors

Hubka formulated the Theory of Technical Systems (Hubka 1973) that has been transformed by Andreasen into the Domain Theory (Andreasen 1992). This theory attracts our attention, as Andreasen proposed strict domains that are required for the design process. Each of these domains has its own "language", allowing the designer to "spell a product in different ways". Packing this theory into the Chromosome model (Ferreirinha et al. 1990) reveals the interconnectivity between entities of different domains, as it is shown in Figure 2.1. Note that the model has gradually been changed (Jensen 1999; Mortensen 1999).

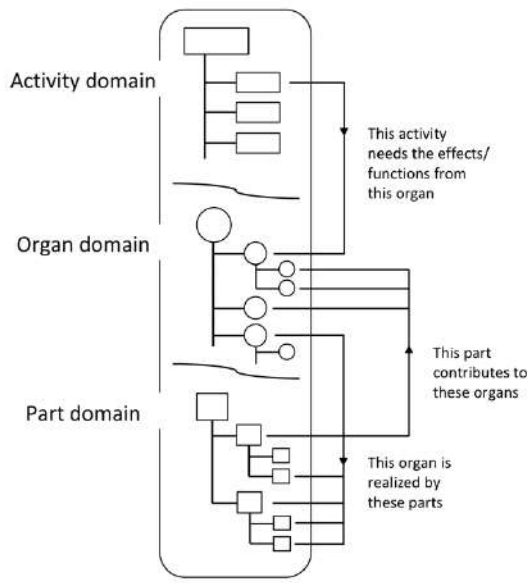


Figure 2.1: Three domains of domain theory

These theories and models have some common ground with our work, because in them the authors – directly or indirectly – have represented a concept. For example, reading a Figure 2.1 reveals that the specific activity needs the functions from the specific organ, or that the specific organ is realized by the specific parts. Such approach might represent some parts of a concept. In our model we develop and present a structured approach to concept *encoding* by assigning the instruments (objects) to processes to operands (objects). These fundamental constructs (objects and processes) form a starting point for concept ontology. We also argue that the same principles are applied to any level of system decomposition. Another difference of our work is that we make use of the state-of-the-art model-based approach to *encode* the concept, its formal and functional relationships, as well as structure, context and the concept of operations.

2.1.2 Axiomatic Design Theory

Introducing the Axiomatic Design theory, Suh (1990; 1998) proposed the scientific approach to design process, which is based on mathematical representation of design process and on design axioms and its corollaries.

In Suh's theory the design process is considered as a constant interplay between what we want to achieve (Functional Requirements in Functional domain) and how we choose to achieve it (Design Parameters in Physical Domain). This process is demonstrated in Figure 2.2, in which the move from FR to DP to conceptualize a design is demonstrated.

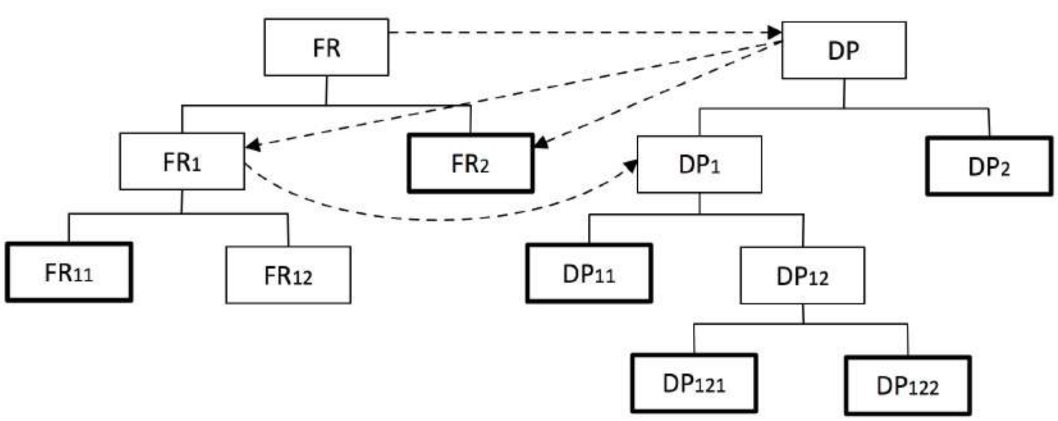


Figure 2.2: Zigzagging process

Suh proposed to view a design process through a mathematical representation. The nature of mapping between a given **FR** and a **DP** vector having a design matrix **[A]** is given by the design equation as:

$$\{\mathbf{FR}\} = [\mathrm{A}]\{\mathbf{DP}\} \tag{1}$$

where {FR} is the functional requirement vector

{DP} is the design parameter vector

[A] is the design matrix.

For example, if the design has three FRs and three DPs, the design matrix has the following view:

$$[A] = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \tag{2}$$

Our approach complements the Suh's theory in rationalizing a conceptualization process, meaning the movement from Functional domain to Physical domain. We call this transition *conceptual design* that facilitates concept *generation*. Such approach also facilitates a *formal analysis*, such as conceptual similarity assessment.

2.1.3 Encapsulation Model

Andreasen et al. proposed the Encapsulation Design Model (Andreasen et al. 2015) that has 5 constituent elements, of which we will discuss the Concept Synthesis in details, as it mostly, concerns a problem we explore in this Thesis.

The Concept Synthesis, or conceptualization model is demonstrated in Figure 2.3. This model consists of three steps: goal formulation, ideation, and evaluation and choice. The authors carefully explain the role of each step and its placement in conceptualization process. In particular, they emphasize the importance of "use scenarios, functionality, appearance, similarity to existing products, features, sub systems, and properties, etc."

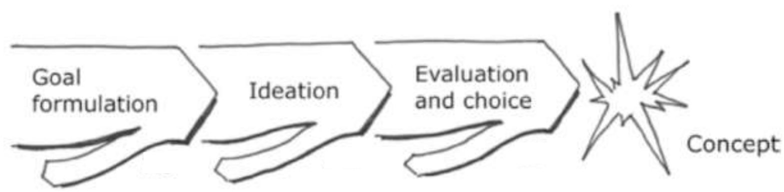


Figure 2.3: Three steps of Conceptualization Model (Andreasen et al. 2015)

This model has a procedural nature, specifying what and in which sequence should be done during the conceptualization. In our work we are aiming at exploring in details the notion of concept itself and its constituents, as well as to propose ontology to *encode* existing concepts and to *generate* the new ones by means of model-based approach.

2.1.4 C-K Theory

C-K theory was first introduced by Hatchuel and Weil (Hatchuel and Weil 2003; Hatchuel and Weil 2009) and has gained the interest in the design community. For instance, Kroll (2013) has compared the C-K theory with parameter analysis. Shai et al. (2013) used C-K theory as an analytical tool to address one of the key challenges of design theory: modeling scientific discovery.

A central proposition of C-K theory is that design can be modeled as the constant interplay between the space of concepts (C) and the space of knowledge (K) (in our terminology, the means to *encode* a concept). Another proposition of the theory is an attempt to improve our understanding of innovative design (in our terminology, the means to *generate* a concept).

Authors explain that the design can only partition an initial concept in the hope that this expansion of attributes will create useful new concepts and new knowledge. The partitioning attributes in C must be extracted from K. C is always tree-structured, whereas K is expanded by new propositions. K grows by the adjunction of new objects (new islands) or by new properties linking these objects (changing the form of the islands). This is shown in Figure 2.4.

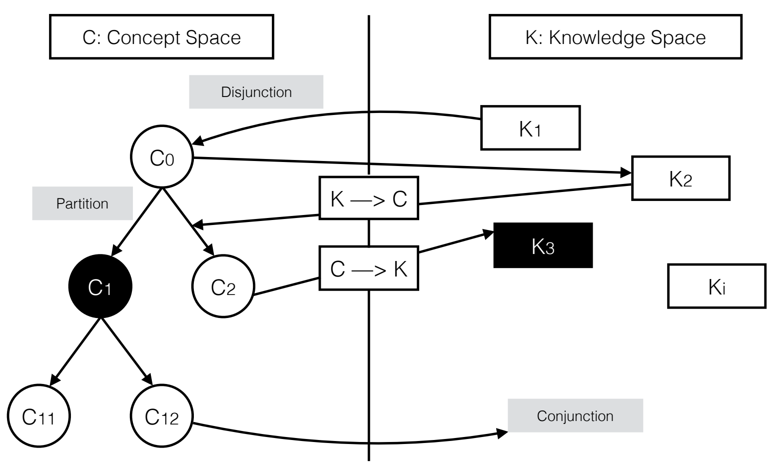


Figure 2.4: C-K dynamics

The authors of the C-K theory made a brief reference to the Axiomatic Design theory developed by Suh (1990). Hatchuel et al. proposed that Suh's theory does not contain a concept or knowledge, described by the theory. They state that design not only generates "solutions" but also, by the same procedures, *new concepts* and *new propositions* in K. Thus, according to them, the C-K theory captures the birth of new objects. This understanding is very important in our theory as well, as the *generating* of new concepts is one of the forms of utility enabled by a successful implementation of the theory proposed in this Thesis.

2.2 What is a Concept?

Concept is not a precisely defined idea. According to Crawley et al., concept is "a product or system vision, idea, notion, or mental image that maps function to form. It is a scheme for the system and how it works. It embodies a sense of how the system will function and an abstraction of the system form. It is a simplification of the system architecture that allows for high-level reasoning" (Crawley et al. 2015). Another definition, proposed by Ulrich and Eppinger, states that a concept is "an approximate description of the technology, working principles, and form of the product. It is a concise description of how the product will satisfy the customer needs" (Ulrich and Eppinger 2015). Andreasen et al. state that "a concept is a design proposal that is detailed enough to justify if it is a good answer to the task and intention, and show a high probability of realization and success" (Andreasen et al. 2015). These three definitions have a number of features in common. We can conclude that the concept has a form, function, and initial hints about how the system works – the concept of operations. The form is the physical embodiment of the system, while the function outlines what processes the system is performing. This definition is a key starting point for this research, as they create an appropriate level of abstraction for the concept framework developed in this Thesis.

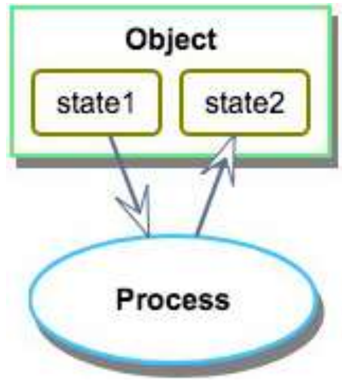
2.3 Representation of Concept

In our dissertation we make use of the conceptual modeling languages, such as Object-Process Methodology (see sub-section 2.3.1) and SysML (see sub-section 2.3.2) in order to represent a concept.

2.3.1 Object-Process Methodology (OPM)

In our Thesis we use the Object-Process Methodology (OPM) (Dori 2002) as the primary conceptual modeling language. OPM represents the models in a graphical notation, which is called Object-Process Diagrams (OPD), and in a textual notation that is entitled Object-Process Language (OPL). Both of these representations follow strict rules to be explained in this sub-section. Both of them can be constructed in a Cloud-based environment called OPCloud (Dori et al. 2019).

OPM has the only two building blocks that are shown in Figure 2.5(a). These blocks are object (shown in rectangle) and process (shown in oval). According to Crawley, the object is “that which has the potential for stable unconditional existence for some period of time” (Crawley et al. 2015). Each object has some state, which is changed by the process (indicated inside the rectangle in Figure 2.5(a)). For example, it can change the state of the object called “spaceship” from state “not built” to state “built”. The corresponding OPL, extracted from OPCloud, is shown in Figure 2.5(b).



(a)

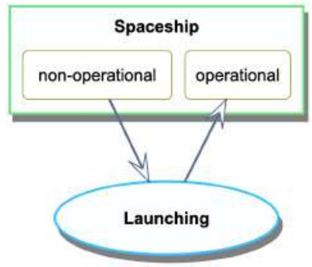
Object can be state1 or **state2 .**
Process changes **Object** from **state1** to **state2 .**

(b)

Figure 2.5: OPM building blocks in OPD (a) and corresponding OPL (b)

Consider an example of a high-level abstraction of the spaceship launching. For this system we imply that the functional intent is to facilitate a safe launch of a spaceship. In other words, the object of this system is "spaceship", while the process is "launching". Together the process and the operand (an object that is changed by the process) constitute the function of the system. Figure 2.6 shows the OPD and OPL for the above-mentioned system. From this figure the one can see that the process "launching" changes the state of the object "spaceship" from "non-operational" to "operational". Note that at this level of abstraction we have not yet found the instrument of execution of the function. The core idea is that our approach together with OPM ontology forms the foundation for finding the alternative concepts based on prescribed process, rather than from unexpected eureka.

As such, the notion presented in Figure 2.6 abstract enough to keep the wide range of the alternative solutions, or concepts.



(a)

Spaceship can be **non-operational** or **operational**.
Launching changes **Spaceship** from **non-operational** to **operational**.

Spaceship can be **non-operational** or **operational**.
Launching changes **Spaceship** from **non-operational** to **operational**.

(b)

Figure 2.6: OPM building blocks in OPD (a) and corresponding OPL (b) for the function "launching a spaceship"

2.3.2 Systems Modeling Language (SysML)

The Systems Modeling Language (SysML) appeared in the early 2000s as the joint effort of the Object Management Group (OMG) and the International Council on Systems Engineering (INCOSE).

According to Fridenthal et. al, SysML has a grammar of nine types of diagrams (Friedenthal 2014), some of which were taken from the Unified Modeling Language (UML). These diagrams cover the entire spectrum of designer needs to represent systems and systems-of-systems. In Figure 2.7 all 9 diagrams – structural and behavioral – are summarized.

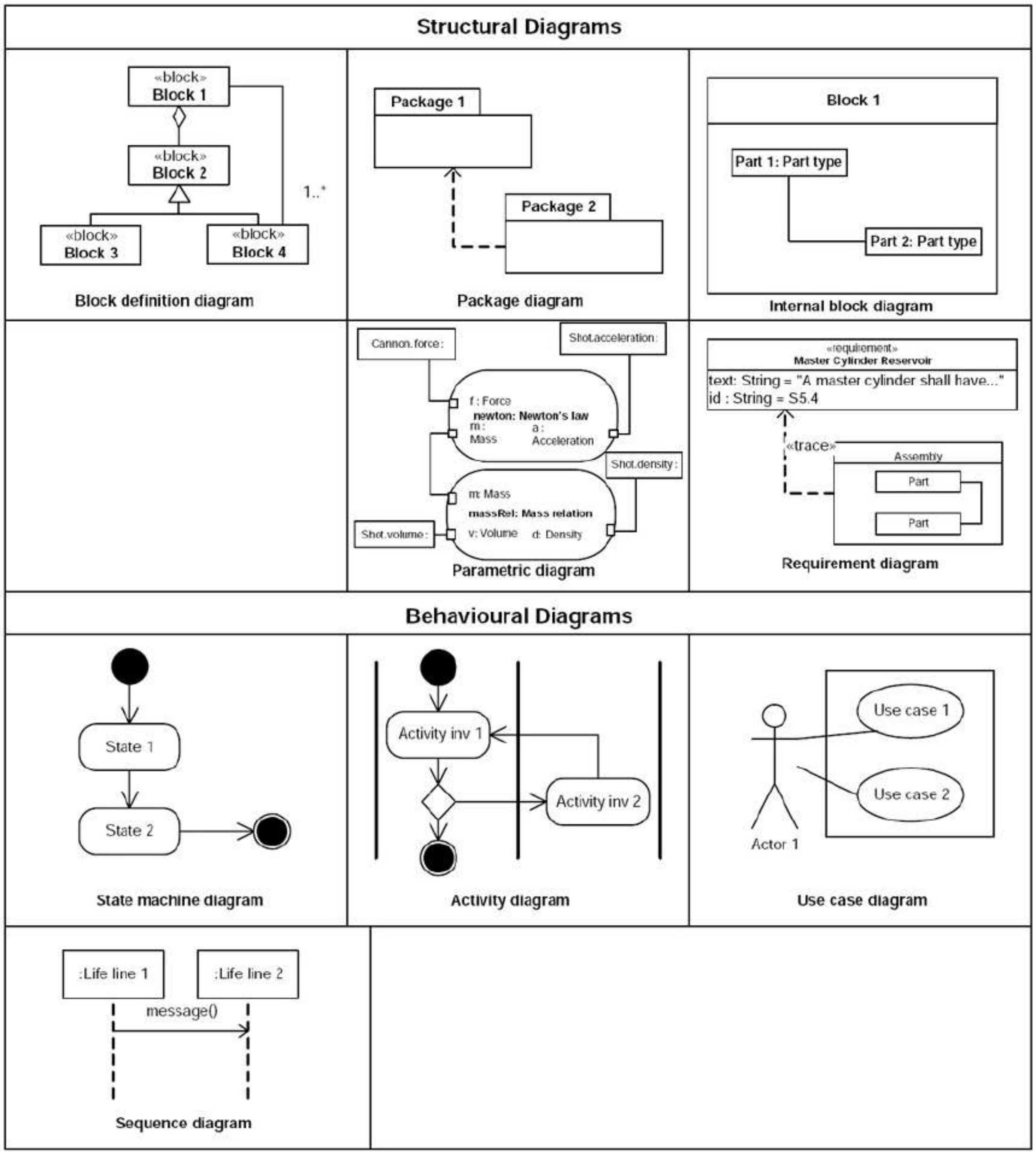


Figure 2.7: SysML diagrams (Holt and Perry 2008)

We considered SysML as the candidate for primary conceptual modeling language in our research work, but instead chose Object-Process Methodology, because it represents the system graphically in a significantly smaller number of diagrams. OPM allows strict definition of building blocks: operands, processes, and attributes. It also allows breaking down of a concept's elements into internal building blocks, and the representation of formal and functional relationships among these internal building blocks. Another advantage of OPM is that it has both linguistic and graphical representations, one of which can be generated from another in OPCAT (Dori et al. 2003). OPM is now specified in ISO 19450 (ISO 2015). However, we will also present the example of the aircraft concept in SysML to demonstrate that this conceptual modeling language can also be used for concept modeling purposes.

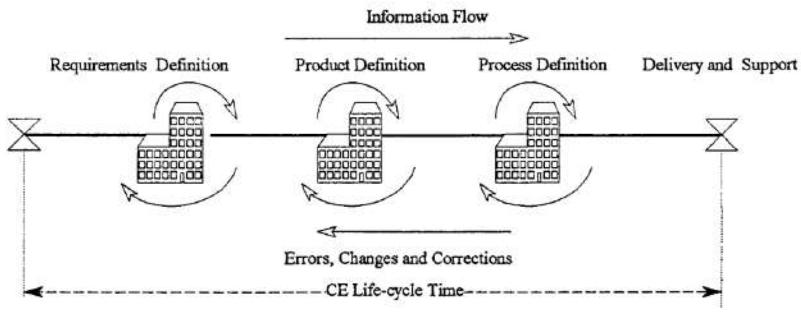
2.4 Concurrent Engineering Environment

2.4.1 Overview of Concurrent Engineering Approach

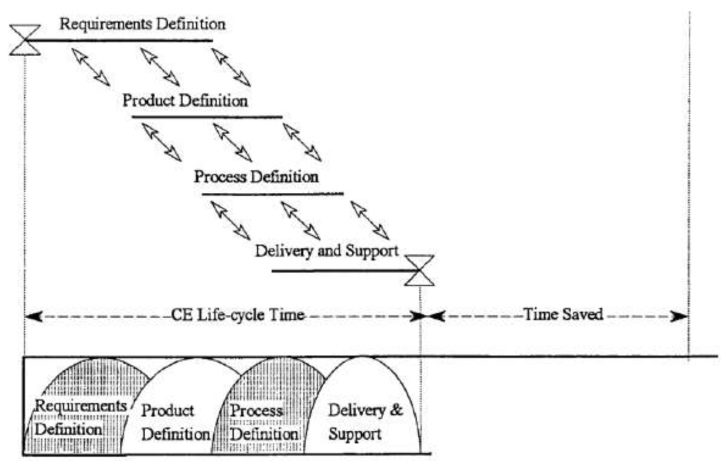
Concurrent engineering is a methodology that allows managing the multidisciplinary design sessions for complex systems. According to ESA, concurrent engineering is "a systematic approach to integrated product development that emphasizes the response to customer expectations. It embodies team values of co-operation, trust and sharing in such a manner that decision making is by consensus, involving all perspectives in parallel, from the beginning of the product life-cycle".

The concurrent engineering approach is used in a variety of industries and business sectors, but a pioneering sector that introduced this methodology was a space sector (Bandecchi et al. 1999).

The key difference of concurrent engineering from the traditional approach is demonstrated in Figure 2.8. From this Figure extracted from the paper of Prasad (1995) the one can notice that the concurrent engineering approach might lead to time saving and parallelization of such steps as requirements definition, product definition, process definition, and delivery and support.



(a)



(b)

Figure 2.8: Sequential engineering (a) versus Concurrent Engineering (b) (Prasad 1995)

Uhlig et al. in ESA mentioned that the adaptation of concurrent engineering methodology into the ESA’s practice (see next sub-section) reduced the time required for the design sessions from 6-9 months to 3-6 weeks (Uhlig et al. 2015).

We provide a brief introduction of Concurrent Engineering approach, because the model-based *system concept representation framework* that is proposed in this Thesis could potentially be used in a Concurrent Engineering environment to facilitate the design process on early stage. Traditionally, the concept development starts with the exploitation of unstructured approaches, such as brainstorming, and a data is gathered from stakeholders without some structured means. Thus, there is a research opportunity to engage the proposed framework at a very beginning of design process, and to use a methodology for searching for the alternative concepts starting from the stakeholders and their needs going down to the solution-neutral environment to the integrated concept and to the concept of operations information.

The framework proposed in our work could be implemented in concurrent engineering environment. Thus, discussion of concurrent engineering is relevant, because it rationalizes the second research question of section 1.5: “How the information required for system concept representation can be encoded in a model-based manner to support system concepts and their alternatives development?”

2.4.2 Concurrent Engineering Centers in the World

Over the decades, dozens of concurrent engineering centers appeared on the map of engineering practice in the world.

The NASA Jet Propulsion Laboratory’s Team X is a “cross-functional multidisciplinary team of engineers that utilizes concurrent engineering methodologies to complete rapid design, analysis and evaluation of mission concept designs" (NASA 1998). It was founded in 1995 and is placed in Caltech, California.

The European Space Agency has its own Concurrent Design Facility (CDF) at the European Space Research and Technology Center (ESTEC) located in Noordwijk, the Netherlands. CDF has been operational starting from 2000 and since then hundreds of design sessions aiming at feasibility studies were conducted there.

In recent years the number of new Concurrent Design Facilities were established and opened for the design studies in business, academia, and industry. The concurrent engineering methodology has been applied to a wide range of problems – for example, it was used for technology roadmapping in Airbus. In 2017 Airbus opened its own Concurrent Design Facility that "allows teams of experts from across several disciplines to work on design studies using concurrent engineering methodology for complex engineering systems" (Airbus 2017); in 2015 the Concurrent Engineering Design Laboratory was established at Skoltech (Golkar 2016); in 2008 the German Aerospace Center (DLR) opened a new Concurrent Design Facility in Bremen that is used mainly for design studies at the new DLR Institute of Space Systems (Schumann 2008). This confirms that concurrent engineering approach found its place in variety of organizations and is used as an effective tool to manage the multidisciplinary design sessions for complex systems in engineering context.

2.5 Research Opportunities and Forms of Utility for Concept Framework Development

As we see from the literature review that aims to support the research questions of section 1.5, concept takes an important place in systems engineering and design practice. It opens up a research opportunity to develop the model-based instruments that would allow *encoding* the existing concepts, and supporting a concepts *generation* process in early phase of systems engineering. Such instrument aimed at supporting systems engineers during the conceptual design is proposed in our Thesis.

The proposed framework has several forms of utility. First it lends structure. Often the information in documents specifying system concept is presented in an unstructured way, as a set of textual and graphical information. The proposed framework provides the means to *encode* this unstructured data into a structured set of both texts and models, which are consistent with each other. This would allow systems engineer to have a concept classification scheme and searchable database, documenting core information about a concept. This approach could be used within the INCOSE's Model-Based Conceptual Design initiative (INCOSE MBCD).

Another utility is that the proposed framework supports concept *generation* at early phases of the design process in a model-based environment. For example, it allows systematic combination of entries to produce new concepts. The proposed framework contains 28 entries and a modeling ontology. By extending and re-combining them, the systems engineer can develop novel concept, revealing alternative and previously unexplored concepts in a concise way. This utility could also be utilized within the INCOSE's Model-Based Conceptual Design initiative (INCOSE MBCD).

In addition to the two above-mentioned forms of utility, there is additional utility in the framework. The framework provides a methodology for the system engineer to *measure* the conceptual similarities among alternative concepts.

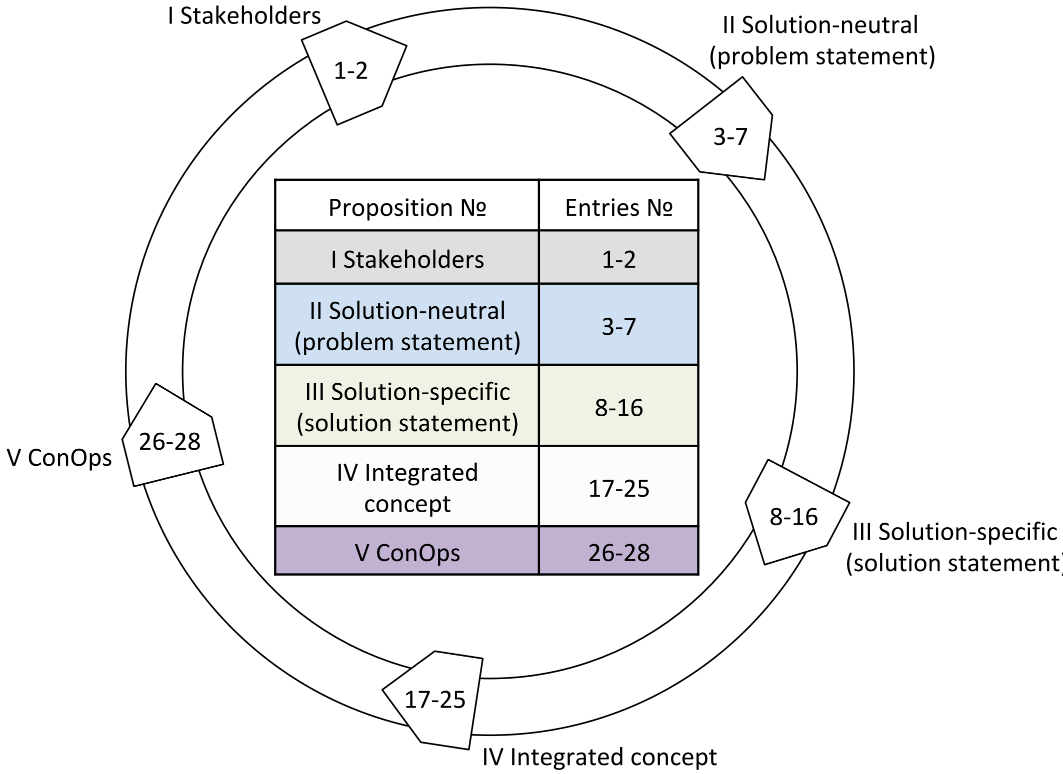
Finally, another additional form of utility is that using the framework leads seamlessly to architecture development. The approach leads to the definition of a concept as a subset of the information that will eventually be used to describe the architecture. Thus, the utility of the framework is that the concept knowledge is *reused* in later stages of the design process – during the architecture development.

2.6 Summary

In summary, in this Chapter we briefly reviewed the systems engineering and design science literature discussing a concept; we discussed the research gaps and research opportunities for *system concept framework development*. All of these aimed at supporting the research questions and hypotheses indicated in section 1.5. We have shown a clear need to dive into the essence of the concept and its constituents and to tie these constructs with the modeling capabilities. We have argued that the proposed model-based *system concept representation framework* serves as a toolbox for the system engineer allowing him or her to keep track of the concept development following a strict ontology and semantics of the conceptual modeling language.

We concluded the Chapter with the forms of utility of the proposed framework.

Chapter 3. Creating a System Concept Representation Framework



Proposition №	Entries №
I Stakeholders	1-2
II Solution-neutral (problem statement)	3-7
III Solution-specific (solution statement)	8-16
IV Integrated concept	17-25
V ConOps	26-28

3.1 Introduction of System Concept Representation Framework

The objective of this Chapter is to develop and present a *system concept representation framework*, which is built upon 5 propositions. These propositions include (I) the stakeholders, (II) the solution-neutral problem statement, (III) the solution-specific solution statement, (IV) the integrated concept, and (V) the concept of operations (ConOps). The rationale for including each of these propositions into the framework is provided in corresponding section of Chapter 3. In this capacity this Chapter supports three research questions formulated in section 1.5. These propositions contain 28 entries of the framework, which will also be explained in this Chapter.

In order to explain the research motivation for the development of model-based *system concept representation framework* methodology, we propose the following analogy with molecular biology. The genetic information is encoded in DNA, made up of four nucleotides: adenine (A), guanine (G), thymine (T), and cytosine (C). Thus, the commonly used letters AGTC represents the building blocks of DNA. As we will show later the "genome" of a concept contains its own "AGTC", which are the operands, processes and attributes. Any system in the world can be represented by means of these three building blocks or their combination.

The building blocks have less meaning without some rules behind explaining how these building blocks are combined and how do they create value. The molecular structure of DNA, for the first time called "double helix", was firstly described by Crick and Watson in 1953 (Crick and Watson 1953). The double helix describes the appearance of double-stranded DNA. Each DNA strand is a long, linear molecule made of AGTC that form a chain. The two strands are connected through interactions between pairs of nucleotides, also called base pairs. One of the principles is that there are two types of base pairing occur: adenine (A) pairs with thymine (T), and cytosine (C) pairs with guanine (G). The "genome" of a concept that is proposed in this Chapter is built upon some rules, explained throughout the Chapter. These rules explain how the framework is developed and represented in a model-based environment.

There is one more important question that should be addressed to understand a genome. This question is related to a chromosome – a DNA molecule with part or all of the genetic material of an organism. The total number of chromosomes is defining a specific organism. The unified concept framework, made up of above-mentioned 5 propositions, forms a “genome” of a concept that is proposed in this Thesis.

The concept framework, presented in this Chapter, is validated through the analytical surveys (patents, urban architectural patterns and software patterns) presented in Chapter 4. Its utility is shown on two space-related case studies: suborbital human spaceflight systems presented in Chapter 5; and space communications systems presented in Chapter 6.

As we discussed in section 1.6, in this dissertation the Design Research Methodology (DRM) is used. According to it, the main goal of Chapter 3 is to provide the prescriptive study. Throughout this Chapter we demonstrate the proposed methodology and its representation in a model-based environment using a running example, the development of the aircraft concept. Each proposition and entry of the concept framework is explained on the example of aircraft concept both in table and digital representations that create the opportunity to *encode* a concept in a model-based environment. At the level of integrated concept (the 4th proposition of the framework) the aircraft concept is specialized into a tube and wing aircraft concept and the blended wing body aircraft concept. One of the forms of utility of the proposed framework is that it enables a formal analysis – such as *measuring* conceptual differences between alternative solutions. This example will be demonstrated for the aircraft concept.

The remainder of this Chapter is organized as follows. In Sect. 3.2 we present the *system concept representation framework*, comprised of 5 propositions spread among 28 entries rooted in systems engineering and design literature. The rationale for including these entries into the framework and their model-based representations are presented in subsequent sections. As such, in Sect. 3.3, the rationale for including first proposition (stakeholders and stakeholders' needs) is provided. There is also a literature review on the subject and the discussion on the allocation of this information in the framework. In Sect. 3.4, we explain the importance and allocation of the second proposition (solution-neutral problem statement), which includes the solution-neutral operand, solution-neutral process, and corresponding attributes. Sect. 3.5 discusses the solution-specific solution statement and its placement in the concept framework. At this step the conceptual design is executed and the possible solution (a concept) is defined. In Sect. 3.6 we discuss the integrated concept, which includes the decomposed elements of form, processes and operands with corresponding attributes, and the information on formal and functional interactions (structure and interactions, correspondingly). Sect. 3.7 explains the rationale for including the Concept of Operations (ConOps) into the framework. Sect. 3.8 demonstrates that the concept framework can be successfully represented in SysML modeling language. Sect. 3.9 demonstrates the utility of the proposed framework for a small-case example, such as a coffee maker. In Sect. 3.10 we demonstrate the unified table with 28 questions, each one of which is related to the corresponding entry of concept framework. Such a table can serve as an appropriate tool for system engineer to check whether all entries of framework are taken into account, or some are missed and should be fixed. The summary of Chapter 3 is presented in section 3.11.

3.2 Propositions and Entries of System Concept Representation Framework

In this section we are presenting the *system concept representation framework* (Menshenin and Crawley 2020), which is built upon five propositions, namely: (I) the stakeholders, (II) the solution-neutral problem statement, (III) the solution-specific solution statement, (IV) the integrated concept, and (V) the concept of operations (ConOps). These propositions are presented in Figure 3.1. The unified framework in table format is presented in Table 3.1, in which the entries of the framework are outlined. These entries from 1 to 28 are spread among the above-mentioned 5 propositions and marked by different colors in order to facilitate the identification of which entry is related to which proposition.

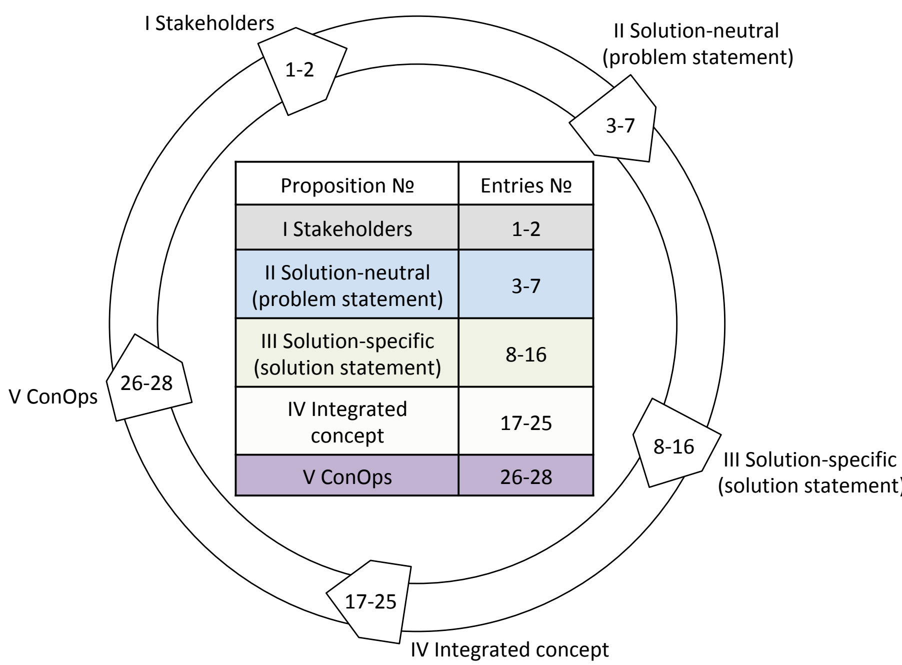


Figure 3.1: Five propositions and twenty eight entries of concept framework

For example, we can see that the first proposition (on stakeholders) contains two entries in the concept framework: the stakeholders and the stakeholders' need; the second proposition (on solution-neutral problem statement) has five entries, numbered from 3 to 7; and so on.

Table 3.1: System concept representation framework

Conceptual design Decomposition

	Proposition I: Stakeholders Proposition II: Solution-neutral Environment (Problem statement)		Proposition III: Solution-specific Environment (Solution statement)		Proposition IV: Integrated concept Proposition V: Concept of Operations
1	**Stakeholders**				
2	Need				
3	**Solution-neutral operand (SNO)**	**8**	**Solution-specific operand (SSO)**	**17**	**Internal Operands (IO)**
4	SNO value attribute	9	SSO value attribute	18	IO value attribute
5	SNO other attribute	10	SSO other attribute	19	IO other attribute
6	**Solution-neutral process (SNP)**	**11**	**Solution-specific process (SSP)**	**20**	**Internal Processes (IP)**
7	SNP attribute	12	SSP attribute	21	IP attribute
		13	**Generic Form**	**22**	**Internal Elements of Form (IEoF)**
		14	Generic Form attribute	23	IEoF attribute
		15	**Specific Form**	24	Structure
		16	Specific Form attribute	25	Interactions
				26	Concept of Operations
				27	Operator
				28	Context

In the following sections we discuss each of these propositions in details, explaining the rationale for including them into the framework and presenting them in a model-based manner. Thus, we are answering the research questions that are formulated in section 1.5. We are also demonstrating the applicability of proposed framework to the running example of this Chapter, the aircraft's alternative concepts development.

3.3 Proposition I. Stakeholders and Stakeholders' Needs

In sub-section 3.3.1 we discuss the literature on the stakeholders and their needs, explaining the rationale for including the information about stakeholders into the concept framework. Sub-section 3.3.2 is dedicated to the representation of how this information is *encoded* in the model-based concept framework. The stakeholders and stakeholders' need information for the running example of this Chapter, the aircraft concept, is presented in sub-section 3.3.3.

3.3.1 Stakeholders and Their Needs: Literature and Rationale

Stakeholders are viewed as an important factor in several fields, including strategic management, political science, engineering, and design science. In each of these fields the stakeholders are considered through different lenses, but what unifies all of them is a shared understanding of the importance of stakeholders' identification and clear definition of their needs.

The first appearance of the word "stakeholders" took place in internal memorandum at the Stanford Research Institute in 1963. This term was defined as "those groups without whose support organization would cease to exist" (Freeman 1984). The different perspectives exist in the literature on who should be considered as stakeholders. One group of authors believes that stakeholders are "people or small groups with power to respond to, negotiate with, and change the strategic future of the organization" (Eden and Ackermann 1998). To them stakeholders are only individuals or groups who can directly affect the organization, or – if we say it more broadly – a problem. According to

Eden and Ackermann, if the individual or group of people doesn't have such power, they are not stakeholders.

Freeman refined his definition, stating that stakeholders are "any group or individual who can affect or is affected by the achievement of the organization's objectives" (Freeman 1984). Nutt and Backoff defined stakeholders as "all parties who will be affected by or will affect [the organization's] strategy" (Nutt and Backoff 1992). Other researchers have also mentioned the importance of including both groups and individuals in the stakeholder definition: those who are affected by the organization or project and those who affect the organization or project (Bryson et al. 2002; Freeman and McVea 2001). These definitions extend the understanding of who can be considered as stakeholders. According to these authors, such individual or groups do not necessarily have power. In this dissertation, we share this understanding and imply this view of the problem while developing a framework.

The stakeholders have needs, which are often explained in a fuzzy way. NASA Systems Engineering Handbook of 2016 states that the needs "are defined in the answer to the question 'What problem are we trying to solve?'" It is important to note that the need "should relate to the problem that the system is supposed to solve but not to the solution". Schrieverhoff and Lindemann mention that "all systems and products are designed to fulfill the needs and requirements of their stakeholders". As such, they proposed a systematic approach to elaborate on evolving stakeholder needs and requirements on a system architecture level (Schrieverhoff and Lindemann 2012).

The literature indicates that stakeholders and their needs play an important role in characterizing a system concept. Their involvement in the design process is crucial to clearly stating the goal – setting up the problem to be solved by the concept. Therefore, we include "stakeholders" and "need" into the concept framework as entries 1 and 2 (see Table 3.1). Although the second entry is stated in the singular, it should be noted that there might be multiple stakeholders and multiple needs (Crawley et al. 2015). In the worked aircraft example, the need is shown as an attribute of the stakeholders on the OPM diagram.

The needs are the attributes of the stakeholders. The better the needs are stated, the more accurate the specialization process will be and more effectively the principle solutions for a stated problem can be found.

3.3.2 Model-Based Representation of Stakeholders and Their Needs

The stakeholders and the stakeholders' need are the first two entries of the framework, captured in Table 3.2. There is also a model-based representation of the same information, the generic structure of which is shown in Figure 3.2 (triangle in Figure 3.2 denotes that the "need" is an attribute of "stakeholders"). This information is allocated upstream, since prior to the concept development the system engineer should clearly identify who are the individuals, or group of people, or organizations, whose needs should be satisfied by the developed concept.

Table 3.2: Entries of proposition I (stakeholders)

Stakeholders	
1	**Stakeholders**
2	Need

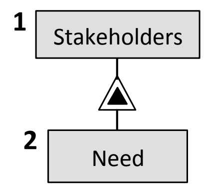


Figure 3.2: Model-based representation of proposition I (stakeholders)

The digital representation of this information, as well as other entries of concept framework, follows the same rules to be explained throughout this Chapter. This notation allows capturing the concept's entries and principle solution at the early design stages. It also creates the opportunity to use the proposed framework within a model-based conceptual design (MBCD) initiative (INCOSE).

3.3.3 Applying the Methodology to the Running Example of Aircraft Concept

In the running example of this Chapter we demonstrate the development of a framework for the aircraft concept. For this concept we assume that the stakeholder is traveler (see entry 1 in Table 3.3 and Figure 3.3), whose need is to get somewhere (see entry 2 in Table 3.3 and Figure 3.3). "Get somewhere" is the fuzzy stated need. This formulation is intentionally presented in such way, because sometimes stakeholders do not have an explicitly formulated need, so this is a task of system architect to clarify the needs of stakeholders. Table 3.3 and Figure 3.3 are the inputs to the concept framework and the digital representation of the same information, correspondingly.

Note that at this level of conceptualization we have not yet stated a problem we are trying to solve. This is a task of the following sub-sections.

Table 3.3: Stakeholders and their needs for aircraft concept

Stakeholders		
1	**Stakeholders**	**Traveler**
2	Need	Get somewhere

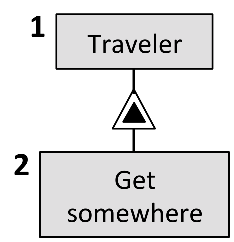


Figure 3.3: Model-based representation of stakeholders proposition for aircraft concept

Note that we put the numbers next to each block in order to demonstrate the interconnection between the representation in a grid format and in an OPM notation.

3.4 Proposition II. Solution-Neutral Information (Problem Statement)

Sub-section 3.4.1 discusses a literature on the *solution-neutral environment*, explaining the rationale for including this information into the concept framework. In sub-section 3.4.2 we demonstrate how the problem statement is encoded in the model-based concept framework. The problem statement for the running example of this Chapter is presented in sub-section 3.4.3.

3.4.1 Solution-Neutral Information: Literature and Rationale

In his book *The Principles of Design* Nam Suh introduced the term *solution-neutral environment*, the purpose of which is to formulate the functional requirements (Suh 1990). Nordlund et al. noted that these functional requirements "shall be stated purely as a requirement and be free of any bias from prospective solution approach such as specific technical discipline or implementation strategy" (Nordlund et al. 2015).

Andreasen et al. have also discussed the term *solution-neutral*, proposing that this is a place for goal formulation (Andreasen et al. 2015). According to the authors, the major aspect of goal formulation in *solution-neutral* is that the solutions are not known at that point. Wallace and Andreasen stated that "A goal statement may never be correct, can never be complete, can never be final, can never prevent incorrect and conflicting interpretations, can never resolve creative conflict, and it should not!" (Andreasen and Wallace 2011).

These ideas rationalize the inclusion of the *solution-neutral* problem statement to the concept framework. The second proposition, dedicated to *solution-neutral* and presented in entries 3 to 7 in Table 3.4 and in Figure 3.4, deals with the solution-neutral operand and the solution-neutral process, plus their attributes. The main goal of this part of framework is to formulate the problem statement based on stakeholders' need. Such a functional intent should be represented in the most abstracted way. The abstraction is important in the formulation of operand and process, because at this level of conceptualization the system architect should be open to any possible ideas and should not – intentionally or unintentionally – focus his or her attention on any specialization of either process or operand. Abstraction keeps open the opportunity to *generate* the novel and previously unexplored concepts. With this background, it becomes clear why potential solutions are not present in the second proposition of framework.

Since this is the first appearance in the framework of such words as operand, process, and attributes, it is important to provide their definitions and features. The two intrinsic parts of any function are its operand and process. The process illustrates a dynamic nature of the function: it reflects the action and is usually represented by verb. The operand is an object that is changed by the process. It is usually represented by noun. Thus, any *function* consists of *operand* and *process* (Dori 2002).

An object and process might have the attributes, which describe their qualities, states or characteristics. According to Hubka and Eder (1988) the attributes are “all those features which belong substantially to the object”. We distinguish the value-related attributes and other attributes (Crawley et al. 2015). If the process changes the attribute of an object, we call such attribute a value-related one. Other attributes are those, which are important to be aware of, but are not changed by the process.

3.4.2 Model-Based Representation of Solution-Neutral Problem Statement

The solution-neutral problem statement consists of the solution-neutral operand, its value-related and other attributes; and the solution-neutral process, with its attributes. This data is related to the second proposition and is included into the entries 3 to 7 of the framework. This part of the framework is presented in Table 3.4.

The purpose of filling in the solution-neutral information in concept framework is to formulate a problem we are aiming to solve. This problem is usually stated in the abstract way: the only generic process and generic operand are presented (with corresponding attributes). Table 3.4 summarizes this information in the framework.

Table 3.4: Entries of proposition II (solution-neutral environment)

Solution-neutral environment (Problem statement)	
3	**Solution-neutral operand (SNO)**
4	SNO value attribute
5	SNO other attribute
6	**Solution-neutral process (SNP)**
7	SNP attribute

Figure 3.4 is a model-based representation of the second proposition built in OPM notation. This information is equivalent to the one presented in Table 3.4. Both versions have exactly the same meaning and can be used to explain the solution-neutral proposition of the framework. Note that the number of value-related attributes, as well as other attributes can vary from concept to concept.

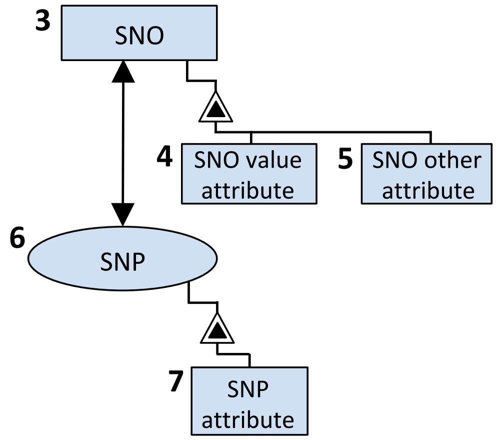


Figure 3.4: Model-based representation of solution-neutral environment

3.4.3 Applying the Methodology to the Running Example of Aircraft Concept

The solution-neutral function of the aircraft concept is “transporting a traveler” (see Table 3.5 and Figure 3.5). As was discussed in previous section, the function consists of process and operand, so we clearly see the solution-neutral process “transporting” (see entry 6 in Table 3.5 and Figure 3.5), and the solution-neutral operand “traveler” (see entry 3 in Table 3.5 and Figure 3.5). Note that in this example the solution-neutral operand is the same as stakeholder.

The value-related attribute of the traveler is “location” (see entry 4 in Table 3.5 and Figure 3.5). We marked “location” as the value-related attribute, because the process “transporting” changes the location of traveler from one place to another. Thus, changing the location of the traveler creates value. Transporting does not change the number of travelers, but it is important to know, thus we included “number” as the solution-neutral’s other attribute (see entry 5 in Table 3.5 and Figure 3.5).

The solution-neutral process “transporting” has an attribute “safely” (see entry 7 in Table 3.5 and Figure 3.5), since regardless the chosen concept, transporting should be performed in a safe manner. The full solution-neutral problem statement is to “transport a traveler from one location to another location, safely”. This information is presented in Table 3.5 and Figure 3.5.

Table 3.5: Solution-neutral information for aircraft concept

Solution-neutral environment (Problem statement)		
3	**Solution-neutral operand (SNO)**	**Traveler**
4	SNO value attribute	Location
5	SNO other attribute	Number
6	**Solution-neutral process (SNP)**	**Transporting**
7	SNP attribute	Safely

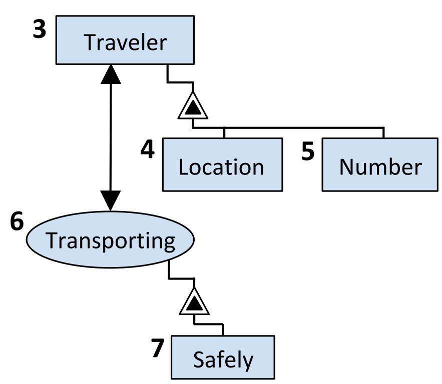


Figure 3.5: Model-based representation of solution-neutral information for aircraft concept

3.5 Proposition III. Solution-Specific Information (Solution Statement)

In this section we provide a rationale for including the solution-specific information into the framework. In sub-section 3.5.1 we discuss the notion of concept and conceptual design on the basis of systematic approach (Pahl and Beitz 2007), C-K design theory (Hatchuel and Weil 2003), generic product development process (Ulrich and Eppinger 2007). In this sub-section we also present the generic model of the conceptual design process. In sub-section 3.5.2 we demonstrate the model-based mechanisms to *encode* the solution-specific information into the concept framework. The solution-specific statement for the running example of this Chapter is presented in sub-section 3.5.3.

3.5.1 Solution-Specific Information: Literature and Rationale

Solution-specific information appears as a result of specialization of the solution-neutral's operand and process and their attributes. In this section we provide a historical perspective on the solution-neutral to solution-specific step, which is commonly known as conceptual design.

In their book, Pahl and Beitz proposed the steps in the planning and design process. According to them, conceptual design is performed after the planning and task clarification step. Pahl and Beitz noted that the role of the conceptual design phase is to define the principle solution. They also presented the steps of conceptual design, which reveal the procedural nature of the conceptual design process: "...by identifying the essential problems through abstraction, establishing function structures, searching for appropriate working principles and combining these into a working structure – the basic solution path is laid down through the elaboration of a solution principle" (Pahl and Beitz 2007).

Ulrich and Eppinger have also focused on the presentation of sequence of activities and tasks in their generic product development process (Ulrich and Eppinger 2007). In our dissertation, we focus on making more exact and standard the description of concept by providing a strict explanation of what is inside the concept, what are its elements and intrinsic parts, what are their formal and functional relationships, and how all this information can be encoded in a model-based manner. In addition to "what" we explain "how" this can be done to better support systems architects and systems engineers.

C-K theory (discussed in details in sub-section 2.1.4) includes a notion of concept (Hatchuel and Weil 2003). In this theory Hatchuel and Weil distinguish concept (Concept Space) and knowledge (Knowledge Space). According to them, the concept is a proposition that has no logical status in the knowledge space. Thus, the authors define a design as "the process by which a concept generates other concepts or is transformed into knowledge, i.e. propositions in the knowledge space" (See Figure 2.4).

The important idea that C-K theory provides to the designer is the inheritance of the previous knowledge in the newly generated knowledge that emerged by means of concept. This logical construct has some common features with our framework, as the movement from solution-neutral to solution-specific becomes possible when the previous knowledge, or generalized function in solution-neutral environment, is mapped to the new knowledge, or the specialized function in solution-specific environment. This is executed by means of conceptual design process that is shown in Figure 3.6.

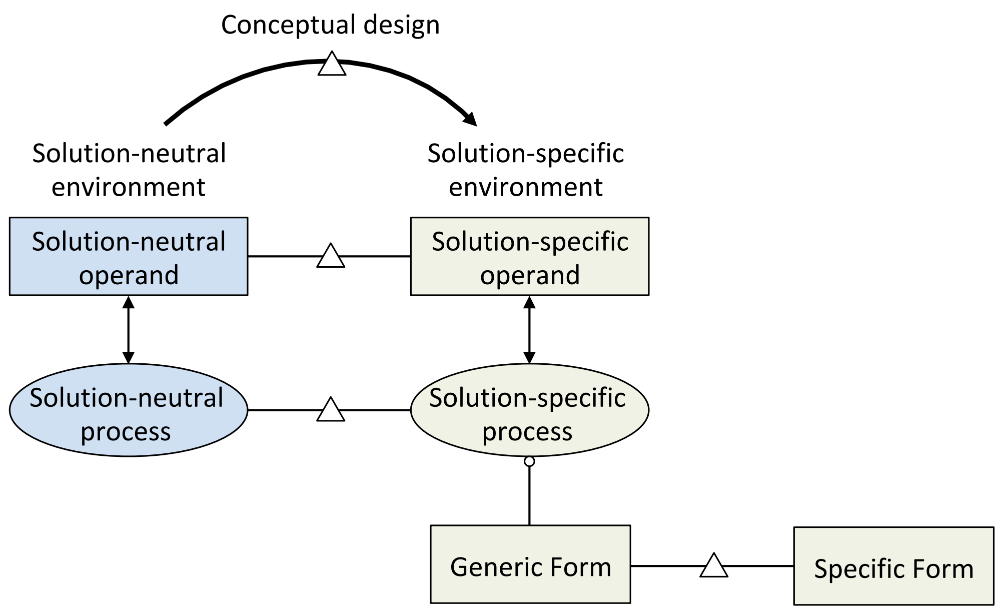


Figure 3.6: Simplified representation of conceptual design

In order to explain the rational for including the third proposition – *solution-specific* – into the concept framework of Table 3.1 as entries 8 to 16, we should discuss the simplified representation of Figure 3.6. From this Figure we can see that conceptual design is a movement from *solution-neutral* to *solution-specific*, which means that here the system architect specializes both the operand and process, and adds the instruments of form (the open triangle in OPM represents specialization, and the round headed arrow the instrument relationship). If the task of the second proposition was to stay in abstract environment, the task of this third proposition is, on the contrary, to specialize the information, to make it more detailed. Importantly, when we specialize the neutral operand and process to the specific operand and process, we should identify the form, which executes the specialized process. We first identify the generic form associated with the process. After that we can specialize the generic form to specific form.

As we will see later, some of the entries of the solution-specific environment are obligatory (operand, process, form), and some of them are additional (their attributes). This has a lot of common with the grammar – to construct a sentence the one needs noun – verb – noun (operand – process – form), while the adjectives (attributes) are optional.

There are different approaches on how to name an object that executes a function. Rephrasing the postulates of the Altstuller's TRIZ theory (Altshuller 1994) we can say that the ideal system is the one whose function is performed without an object performing that function. However, in real life we do not often meet the systems that execute their functions in ideal conditions, thus there should be a consensus on how to name the object that executes the function.

Andreasen et al. (2015) proposed to use the word *operator*, highlighting that the "operators drive the changes in the operands". However, we use the term *operator* differently, it is a part of ConOps proposition to be discussed in sub-section 3.7. Thus, we distinguish two words, and in our work, as it is mentioned above, we call the instrument that executes a function a *form* (See Table 3.1). We also distinguish different types of form – the generic form is an abstract version of an object that is usually associated closely with the solution-specific process. The specific form is a further specialization of the generic form. In general, several specific forms can be associated with each generic form.

In our aircraft example the solution-neutral operand “traveler” further specializes to “passenger”. The solution-neutral process of transporting specializes to flying. The generic form “aircraft” is closely associated with flying, but can be specialized to “tube and wing aircraft”, or to “blended wing body aircraft”. This will be discussed in details in sub-section 3.5.3. The generic form, the specific form, and their attributes are also part of the third proposition of the framework of Table 3.1.

3.5.2 Model-Based Representation of Solution-Specific Solution Statement

Similarly to the previous section, the solution-specific solution statement consists of the solution-specific operand, its value-related and other attributes; and the solution-specific process, with its attributes. However, the major difference between the solution-neutral and the solution-specific lies in the presence of form, shown in Figure 3.6. When we specialize the neutral operand and process to the specific operand and process, we should identify the form, which executes the specialized process. We first identify the generic form (see the entry 13 of the concept framework presented in Table 3.6 and Figure 3.7) associated with the process. After that we can specialize the generic form to specific form (see the entry 15 of the concept framework presented in Table 3.6 and Figure 3.7).

The solution-specific information occupies the entries 8 to 16 of framework presented in Table 3.1. The extracted information is presented in Table 3.6, and Figure 3.7 is a model-based representation of the same amount of information, as it is shown in Table 3.6. It should be noted that the generic model presented in Figure 3.7 has a clear connection to the diagram of Figure 3.4. In Figure 3.7 we omitted the representation of all details that has already been discussed in Figure 3.4. The neutral operand and neutral process are specialized to the specific operand and specific process, correspondingly.

Table 3.6: Entries of proposition III (solution-specific information)

8	**Solution-specific operand (SSO)**
9	SSO value attribute
10	SSO other attribute
11	**Solution-specific process (SSP)**
12	SSP attribute
13	**Generic Form**
14	Generic Form attribute
15	**Specific Form**
16	Specific Form attribute

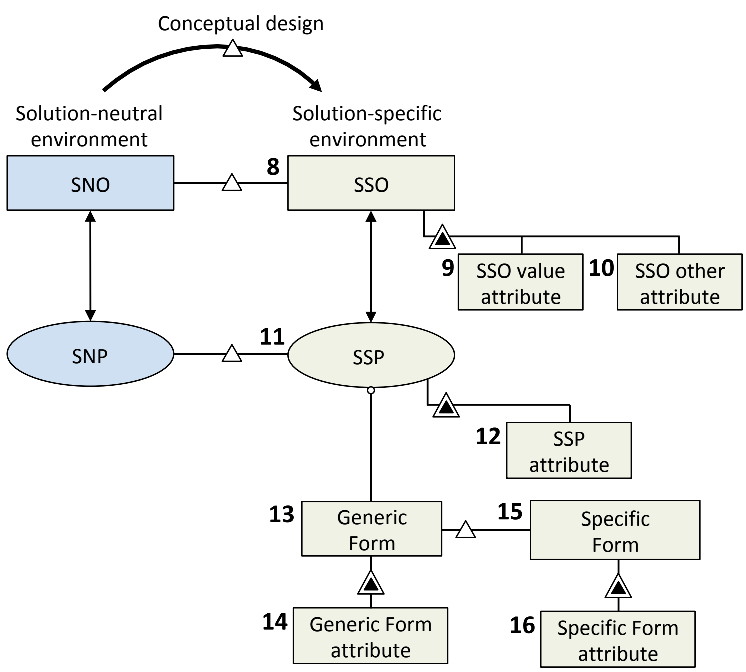


Figure 3.7: Model-based representation of conceptual design process

3.5.3 Applying the Methodology to the Running Example of Aircraft Concept

The core essences of this part of the concept framework are the execution of conceptual design and the definition of the specialized operand and specialized process. This information narrows down the set of possible solutions. As such, the generic form is defined and is associated with the solution-specific process. The specific form is a result of specialization of the generic form.

In this sub-section we demonstrate an illustrative application of the framework to the “tube and wing aircraft”, and “blended wing body aircraft” concepts (see Figure 3.8). Table 3.7 and Figure 3.9 represent the solution-specific information related to these two concepts.

Figure 3.8: Tube and wing (left) and Blended wing body (right) aircraft concepts

Table 3.7: Solution-specific information for aircraft concept

8	**Solution-specific operand (SSO)**	**Passenger**	
9	SSO value attribute	Location	
10	SSO other attribute	Number	
11	**Solution-specific process (SSP)**	**Flying**	
12	SSP attribute	Safely	
13	**Generic Form**	**Aircraft**	
14	Generic Form attribute	Cost	
15	**Specific Form**	**15A**	**15B**
		Tube and wing aircraft	Blended wing body aircraft
16	Specific Form attribute	Cost	

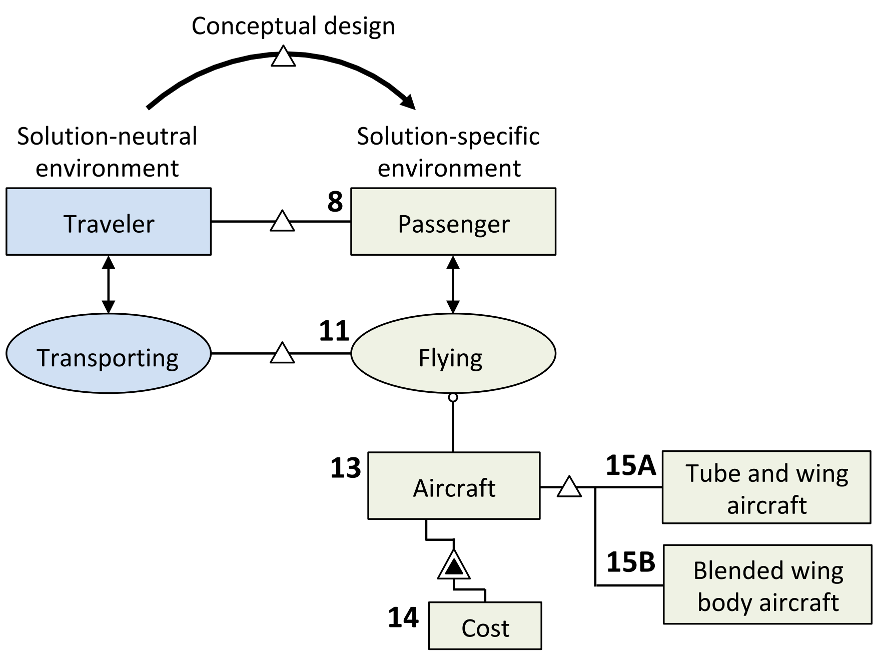


Figure 3.9: Model-based representation of conceptual design of aircraft concept

Figure 3.9 demonstrates the same information as indicated in Table 3.7. Note that we omitted the representation of the attributes for the reasons to be discussed later. Also at the level of specific form (entry 15) we marked the alternative concepts by labeling them with corresponding letter - A (tube and wing aircraft) and B (blended wing body aircraft). Note that the attributes "cost" are related to the production cost.

Figure 3.10 represents entries 1 to 16 of the framework presented in Table 3.1, applied to the tube and wing aircraft concept. As such, the propositions on stakeholders, the solution-neutral environment, and the solution-specific environment are all shown in this OPM-based diagram. Our notion of conceptual design is the increasing specialization conveyed by the movement from left to right, from the neutral to the specific environments. One can see from the Figure 3.10 that "transporting traveler" specializes to "flying passenger". Such a bridge from "transporting" (entry 6) to "flying" (entry 11) narrows down the set of possible solutions, allowing the system architect to assign the generic form "aircraft" (entry 13) to the specialized process "flying" (entry 11). Further specialization of "aircraft" leads to the specific form "tube and wing aircraft" concept (entry 15A). This is the level of abstraction that an aircraft conceptual designer at Airbus or Boeing actually considers, and it is important to note that it only appears at entry 15, because all of the earlier entries inform this specific form.

Figure 3.10 also contains important information on the decision that characterizes the conceptual design process. Propositions I (stakeholders) and II (solution-neutral environment) are responsible for the high-level information supporting the problem formulation, and are used to weight the decisions. Proposition III (solution-specific environment) highlights four key conceptual decisions. First, the operand must be specialized: the specialized operand could be "passenger" or "passenger with cargo". Second, the process of transporting must be specialized: it could be "flying" or "rolling", leading to completely different generic forms. Thus, the third conceptual decision is choosing the instruments of "flying" – it could be "aircraft", or "helicopter", for example (the instruments of "rolling" could be "car" or "train", for instance). The fourth conceptual decision is choosing between "tube and wing aircraft" and "blended wing body" aircraft concepts.

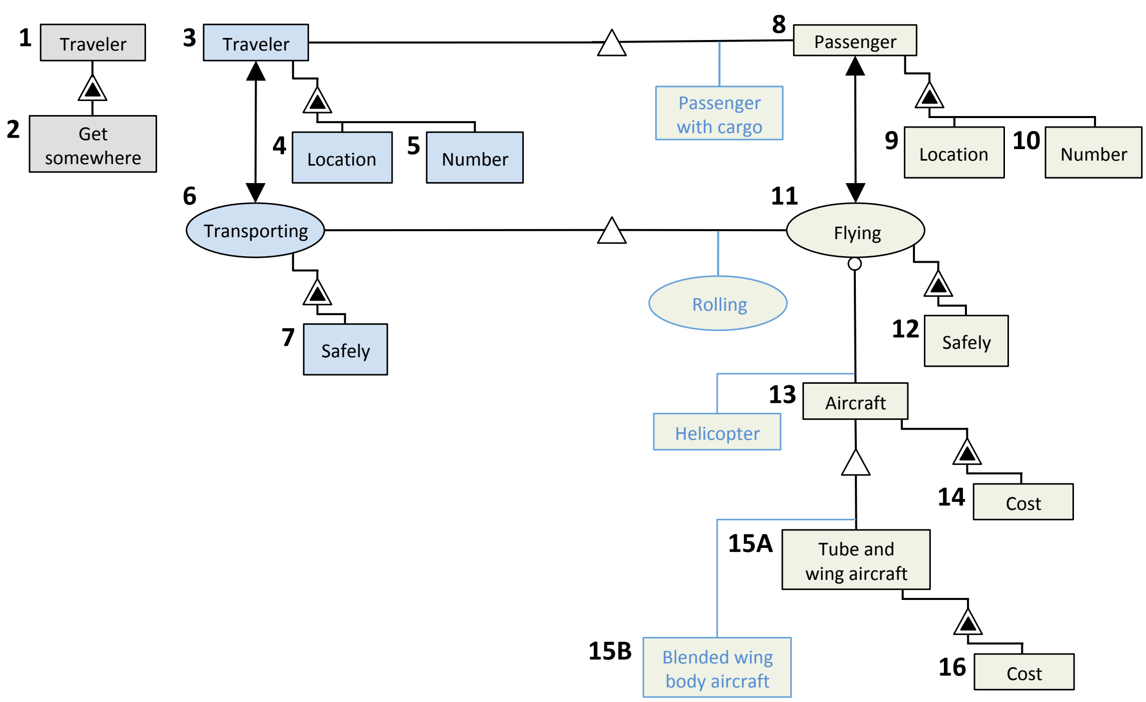


Figure 3.10: Graphical representation of solution-specific information for aircraft concept

We should note several rules that should be taken into account during the creation of model-based diagram of conceptual design, presented in Figure 3.10. These rules are:

1. Inheritance of attributes: attributes are inherited "from left to right" (between solution-neutral and solution-specific) and "from top to down" (between the generic form and specific form). For example, the attributes mentioned in the entries 9 and 10 of the Figure 3.10 are actually inherited ones from the solution-neutral environment (entries 4 and 5) and can be found in corresponding Table 3.5 and Figure 3.5. In future we will not present the entries 9 and 10 assuming that the attributes are inherited from the generic information;

2. The attributes should be moved as far "up and left" towards to the solution-neutral operand/process as possible;

3. The value-related attributes are those attributes that are changed by the process.

3.6 Proposition IV. Integrated Concept

It is also important that the actual information that distinguishes a "tube and wing aircraft" from a "blended wing body aircraft" is stored in the integrated concept, entries 17 to 25 of the framework. This session addresses the integrated concept, presenting the literature on the integrated concept and the rationale for including it into the concept framework in sub-section 3.6.1, providing the model-based representation in sub-section 3.6.2, and presenting the integrated concepts for the tube and wing aircraft concept and the blended wing body aircraft concept in sub-section 3.6.3.

3.6.1 Integrated Concept: Literature and Rationale

In the Theory of Technical Systems Hubka (1973) introduces the term organ, which is a “system that realizes a given internal function of a technical system”. He states that the organ can be identified at the different level of abstraction. Hubka considers the interconnections among organs as couplings. Thus, the output from one organ is the input to the next organ. Hubka formulated the Theory of Technical Systems (Hubka 1973), which has been further developed by Andreasen into the Domain Theory (Andreasen 1992). This theory has been discussed in sub-section 2.1.1. The three domains of domain theory are presented in Figure 2.1.

The role of each domain is various. As such, the activity domain explains how the product is used with the focus on the transformation of operands. The organ domain describes the functions in a product, and the part domain concentrates on the parts and their assemblies.

The fourth proposition of the framework of Table 3.1 is the integrated concept, the goal of which is to decompose a solution-specific concept into the set of internal elements of form, internal processes, and internal operands (with corresponding attributes). Through this decomposition the systems architect can rigorously *encode* all the information required to describe the integrated concept, indicated in entries 17 to 25 of framework in Table 3.1.

We find that such a decomposition of one level is necessary to describe a concept and distinguish it from other concepts (Menshenin and Crawley 2018b). In their work Maier et al. (2016) discussed the model granularity. They stated that “Depending on how, or to what degree, the target system is abstracted a model emerges with a certain level of abstraction and, related to this, granularity”. The integrated concept appears in a result of decomposition, which “generally leads to more fine-grained models” (Maier et al. 2016). Thus, we can say that the difference of our approach with Andreasen’s one is in the different levels of granularity of the model. The rationale to include the decomposition into the concept framework can be explained by example of such commonly recognized way of technology protection as patents. We have observed that the important “claims” of a patent are generally at the level of the integrated concept. This will be discussed in details in Chapter 4. The decomposition views to show the integrated concept for the aircraft concept will be shown later in this section.

Eppinger and Browning distinguish two categories of relationships: a hierarchical (vertical) and a lateral (horizontal) (Eppinger and Browning 2012). Vertical relationships deal with the “decomposition or breakdown of system into elements”. In the notation of Figure 3.10, this happens when system architect takes the solution-specific concept elements, presented in entries 8 to 16 of the concept framework and decomposes them into internal elements of form, process and internal operands, which are shown in entries 17-23 of Figure 3.11. It should be noted that entries 17-23 form a class, and can be filled in for each decomposed entity as an instance. Horizontal relationships “stem from interactions between elements, such as flows of material or information, at the same level”. This aspect of the concept framework is shown in entries 24 and 25 of Figure 3.12, in which we discuss “structure” and “interactions”. By structure we imply the formal relationships among the elements – such relationships have a static nature such as connectivity and spatial arrangement. By interactions we imply the functional relationships, which emphasize the dynamic interactive nature of the interchange of internal operands (Crawley et al. 2015).

The importance of structure and interactions in the integrated concept can also be explained by the work of Yassine et al. They mention that "decomposition helps in containing the technical complexity of the design; however, it increases its managerial complexity. The synthesis of the different elements (subsystems) into a final product (or system) requires the identification and understanding of the interrelationships among the different elements" (Yassine et al. 2003). This is exactly the information that we include in our concept framework, and the structure (entry 24 of framework) and interactions (entry 25 of framework) serve these purposes. These entries support the system architect with the information about which elements are connected and what is exchanged between them.

3.6.2 Model-Based Representation of Integrated Concept

The integrated concept part of the framework consists of the internal operands (entries 17), internal processes (entries 20), internal elements of form (entries 22), and their attributes (entries 18, 19, 21, 23); as well as structure (formal relationships) – entry 24 – and interactions (functional relationships) – entry 25. All this information is summarized in Table 3.8 and Figures 3.11 and 3.12.

Since the integrated concept deals with the decomposition of the specific form, it obviously represents the number of internal operands, internal processes, and internal elements. The structure represents how these internal elements are connected in a static way, while the interactions shows the dynamic nature of relationships.

Table 3.8: Entries of proposition IV (Integrated concept)

Integrated concept	
17	**Internal Operands (IO)**
18	IO value attribute
19	IO other attribute
20	**Internal Processes (IP)**
21	IP attribute
22	**Internal Elements of Form (IEoF)**
23	IEoF attribute
24	Structure
25	Interactions

The model-based representation of the entries 17-23 of integrated concept is demonstrated in Figure 3.11 (we omitted the representation of attributes). Note that this is a representation for three internal elements of form, yet the model allows including as many elements as required.

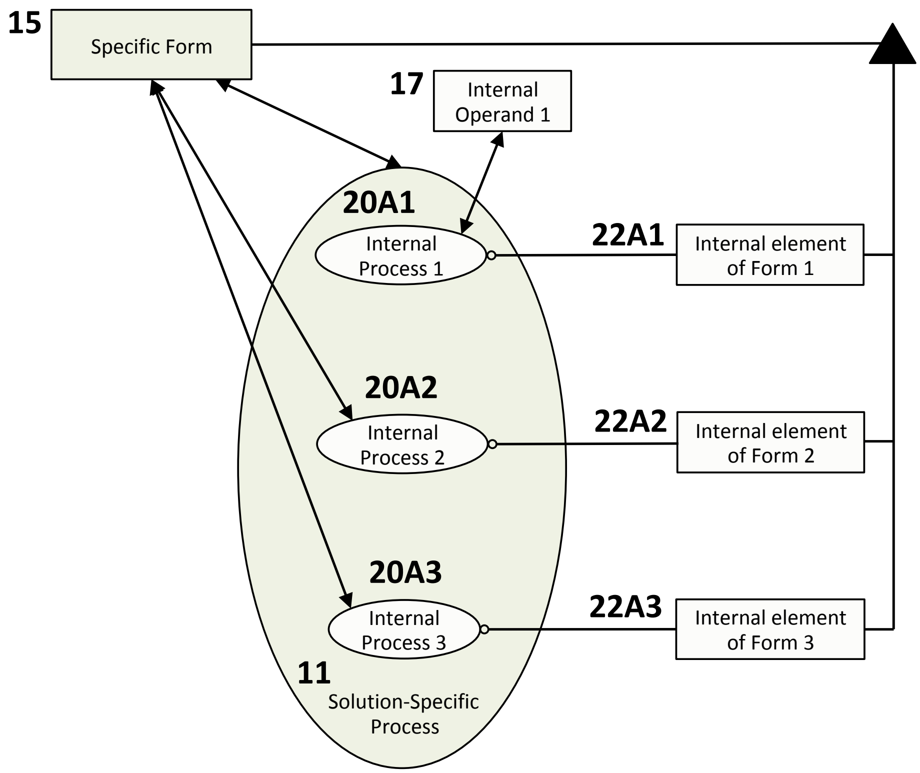


Figure 3.11: Model-based representation of integrated concept

The generic framework for structure (entry 24) and interactions (entry 25) is shown in Figure 3.12. In Figure 3.12 we demonstrate the examples of formal relationships (represented by such words as “attached”, “embedded”, etc.) and functional relationships (represented by such words as “provides thrust”, “transfers load”, etc.)

Figure 3.13 presents a general template for *encoding* information presented in Figure 3.12 into the DSM-based matrix. This view is important for a number of reasons. First, the one DSM matrix contains both types of relationships – specialization and decomposition. Secondly, the DSM representation facilitates a further formal analysis that will be discussed below.

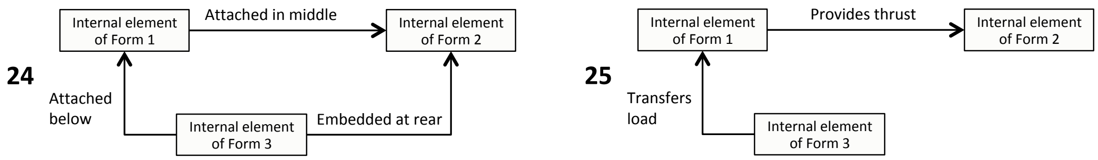


Figure 3.12: Generic framework for structure and interactions

	Internal element of Form 1	Internal element of Form 2	Internal element of Form 3
Internal element of Form 1			Attached below Transfers load
Internal element of Form 2	Attached in middle Provides thrust		Embedded at rear
Internal element of Form 3			

Figure 3.13: Structure (black) and interactions (colored) in DSM

3.6.3 Applying the Methodology to the Running Example of Aircraft Concept

In order to demonstrate the utility of the integrated concept representation, we will apply it to two specific forms (entry 15 of framework) – tube and wing aircraft concept (T&W aircraft concept) (entry 15A of Figure 3.10) and blended wing body aircraft concept (BWB aircraft concept) (entry 15B of Figure 3.10). We will present the core entries of integrated concept to demonstrate the conceptual difference of two concepts and how well the proposed concept framework can *encode* this difference and enable a formal analysis based on it.

Table 3.9 and Figure 3.14 demonstrate the entries 17 (internal operands), 20 (internal processes), and 22 (internal elements of form) of integrated concept for T&W aircraft concept. Note that we omitted the representation of the attributes.

In sub-section 3.5.3 we mentioned 3 rules that should be followed while constructing the concept framework. The fourth rule is related to the enumeration of the concept entries. In particular, the rule is that we first enumerate the internal processes (entries 20A1, 20A2, etc.), after that we enumerate the internal elements of form (entries 22A1, 22A2, etc.) Note that the element "tube and wing aircraft" for the integrated concept enumeration (entry 17A2) is the same as it is at the level of solution-specific statement (entry 15A); while the element "passenger" for the integrated concept enumeration (entry 17A1) is the same as it is for the solution-specific statement (entry 8).

Table 3.9: Integrated concept for T&W aircraft concept

17	**A1 Passenger**	**A2 Tube and wing aircraft**	**A2 Tube and wing aircraft**	**A2 Tube and wing aircraft**	**A2 Tube and wing aircraft**
20	**A1 Carrying**	**A2 Lifting**	**A3 Accelerating**	**A4 Stabilizing Pitch**	**A5 Stabilizing Yaw**
22	**A1 Fuselage**	**A2 Wing**	**A3 Engine**	**A4 Horizontal Tail**	**A5 Vertical Tail**

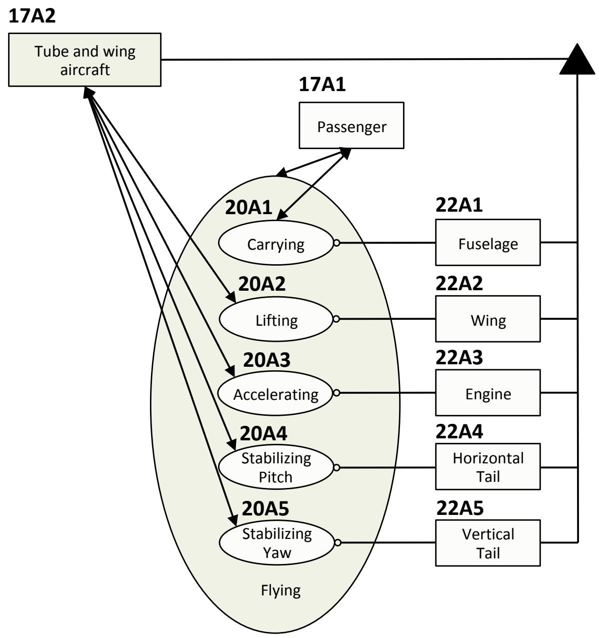


Figure 3.14: The integrated concept for T&W aircraft concept

As it can be seen from Table 3.9 and Figure 3.14, the specific form "tube and wing aircraft" is decomposed into internal elements of form fuselage (entry 22A1), wing (entry 22A2), engine (entry 22A3), horizontal tail (entry 22A4), and vertical tail (entry 22A5). These entries act on the following internal processes: carrying (entry 20A1), lifting (entry 20A2), accelerating (entry 20A3), stabilizing pitch (entry 22A4), and stabilizing yaw (entry 22A5), correspondingly. In case of T&W aircraft concept each internal element of form acts on its own internal process directly, and there is only one element per each process.

Table 3.10 and Figure 3.15 demonstrate the entries 17 (internal operands), 20 (internal processes), and 22 (internal elements of form) of integrated concept for BWB aircraft concept. Note that we omitted the representation of the attributes.

Table 3.10: Integrated concept for BWB aircraft concept

17	**B1 Passenger**	**B2 Blended wing body aircraft**	**B2 Blended wing body aircraft**	**B2 Blended wing body aircraft**	**B2 Blended wing body aircraft**	**B2 Blended wing body aircraft**
20	**B1 Carrying**	**B2 Lifting**	**B2 Lifting**	**B3 Accelerating**	**B4 Stabilizing Pitch**	**B5 Stabilizing Yaw**
22	**B1 Fuselage**	**B1 Fuselage**	**B2 Wing**	**B3 Engine**	**B2 Wing**	**B2 Wing**

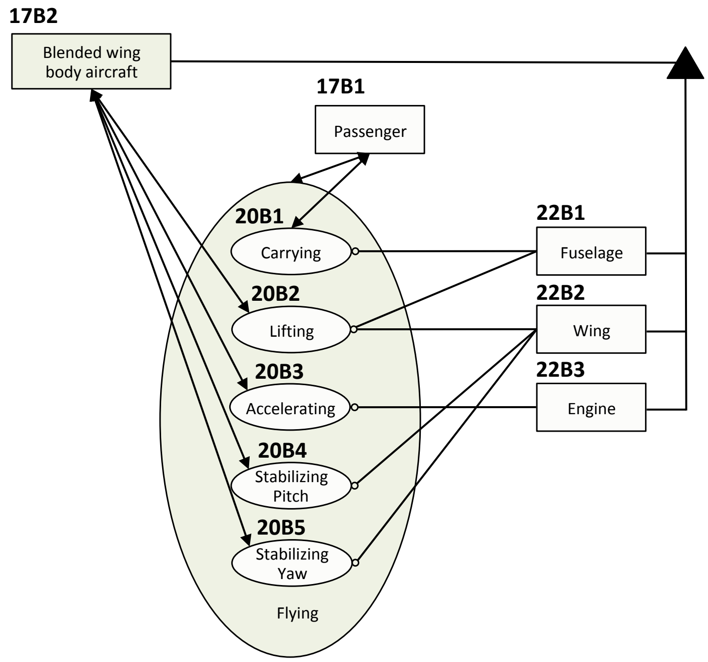


Figure 3.15: The integrated concept for BWB aircraft concept

The specific form "blended wing body aircraft" is decomposed into internal elements of form fuselage (entry 22B1), wing (entry 22B2), and engine (entry 22B3) (see Figure 3.15). Each internal element of form is an instrument for one or a few internal processes – carrying (entry 20B1), lifting (entry 20B2), accelerating (entry 20B3), stabilizing pitch (entry 20B4), and stabilizing yaw (entry 20B5). The conceptual differences with T&W aircraft concept are apparent – both in the number of elements of form, and their assignment as instruments of processes. The conceptual difference between T&W aircraft concept and BWB aircraft concept in details is shown in Figure 3.16.

			Transporting									
			Flying (T&W aircraft)					Flying (BWB aircraft)				
			Carrying	Lifting	Accelerating	Stabilizing Pitch	Stabilizing Yaw	Carrying	Lifting	Accelerating	Stabilizing Pitch	Stabilizing Yaw
Vehicle	Tube and wing (T&W) aircraft	Fuselage	V									
		Wing		V								
		Engine			V							
		Horizontal tail				V						
		Vertical tail					V					
	Blended wing body (BWB) aircraft	Fuselage						V	V			
		Wing							V		V	V
		Engine								V		

Figure 3.16: Conceptual difference between two alternative concepts: T&W aircraft concept and BWB aircraft concept

In Figure 3.16 the allocation of internal elements of form (in rows) to internal processes (in columns) is demonstrated through a DSM-based method advanced in our work (Menshenin and Crawley 2018a). This matrix has both concepts – a T&W aircraft and a BWB aircraft. In Figure 3.17 we present the same information in a model-based environment.

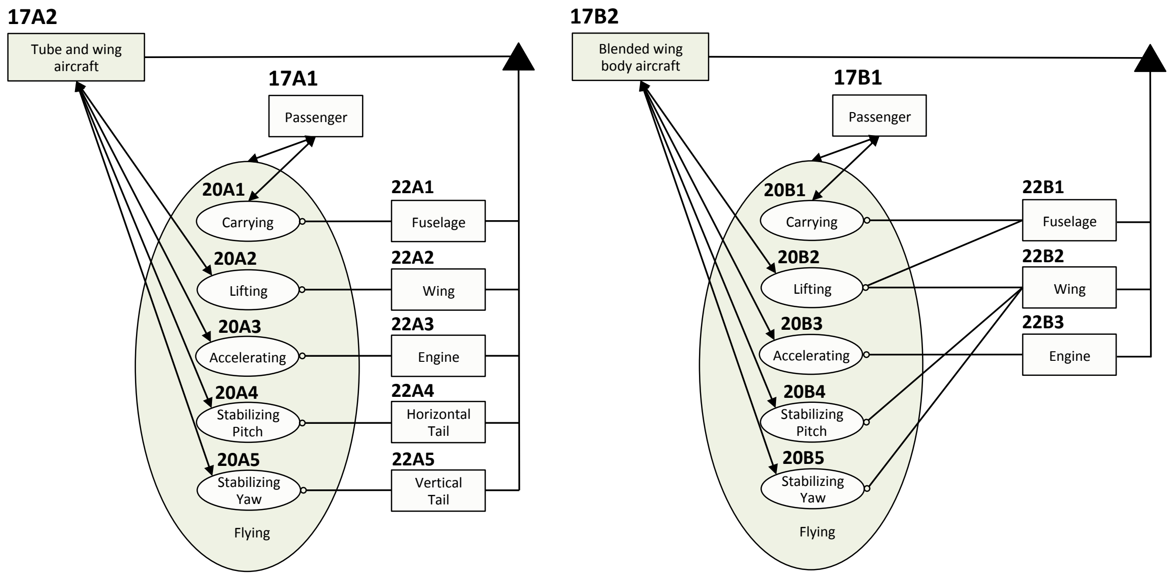


Figure 3.17: Integrated concept for T&W aircraft concept (left) and BWB aircraft concept (right) showing the differences in decomposition of form and form-process assignment. Attributes are not shown

One of the outcomes that can be seen from Figures 3.16 and 3.17 is that T&W aircraft concept has 5 internal elements of form, each one of which is an instrument of a particular internal process. In case of BWB aircraft concept, only 3 internal elements of form perform the same 5 internal processes, which is a coupled design (Suh 1990). One of the advantages of the proposed framework is the opportunity to see these dependencies at early design stages in the model-based environment, and to engage a formal analysis, such as conceptual similarities assessment presented in Figure 3.18 for both concepts – tube and wing (T&W) aircraft concept and blended wing body (BWB) aircraft concept.

Figure 3.18 contains two types of information. First, it assesses the level of similarity between T&W and BWB aircraft concepts. In particular, the one can see that out of 5 internal elements of form (fuselage, wing, engine, horizontal tail, and vertical tail), 3 are the same for both concepts: fuselage, wing, and engine.

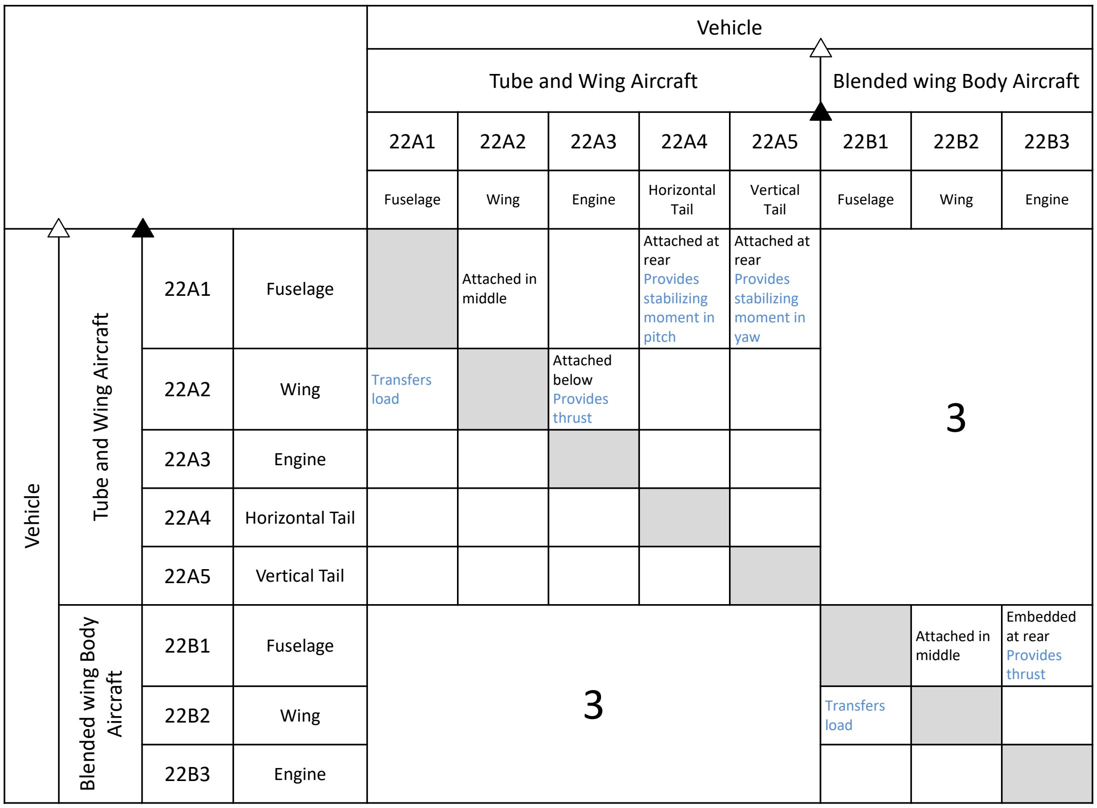
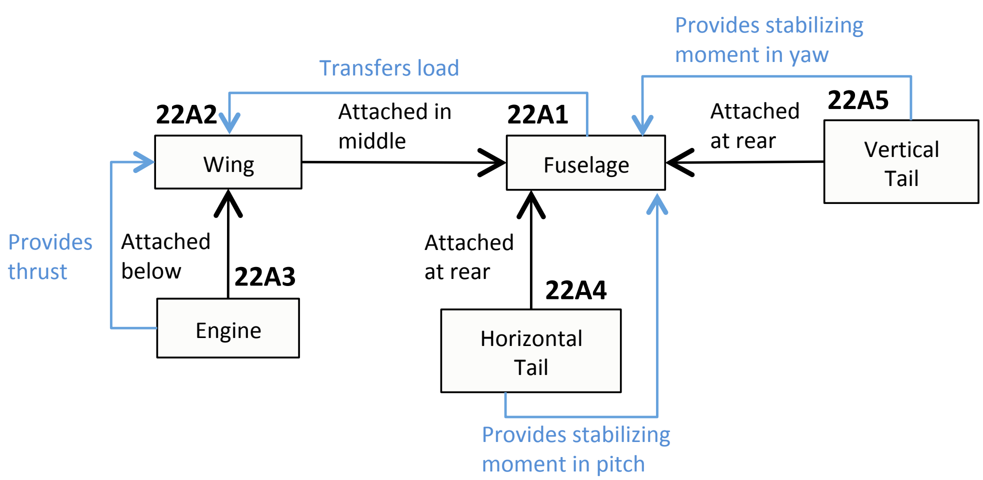
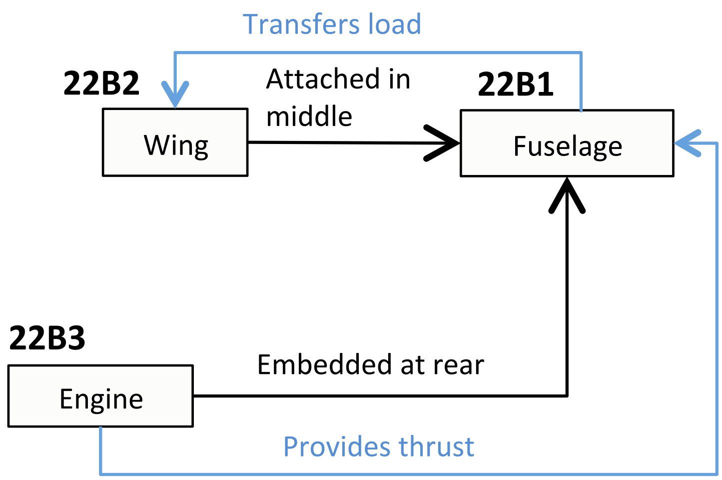


Figure 3.18: Conceptual similarity assessment and structure/interactions information

The second type of information contained in Figure 3.18 is structure (presented in black text inside the cells) and interactions (presented in colored text inside the cells). This representation corresponds to the information presented in Figure 3.19 demonstrating the information about entries 24 (structure) and 25 (interactions) of the concept framework in a model-based environment. Note that black lines in Figure 3.19 reflect the formal relationships, while the color lines reflect the functional relationships. The system architect, for example, can see from Figure 3.19(a) that the formal connection and alignment of the engine in case of T&W aircraft concept is “attached below” the wing; or that the vertical tail is “attached at rear” of the fuselage. Also, this diagram informs us about the interactions: the engine “provides thrust” to the wing; or the vertical tail “provides stabilizing moment in yaw” to the fuselage. In Figure 3.19(b), which informs us about BWB aircraft concept, we see a smaller number of internal elements of form. From this Figure we may see that in case of this alternative concept the engine is “embedded at rear” of the fuselage; while the wing is “attached in middle” of the fuselage. In terms of interactions the fuselage “transfers load” to the wing; while the engine “provides thrust” to the fuselage.



(a)



(b)

Figure 3.19: Structure (black) and interactions (colored) for T&W aircraft concept (a) and for BWB aircraft concept (b)

There should be some clarification in regards to the coupled design, the example of which is shown in Figure 3.15 and Figure 3.17 (right) for BWB aircraft concept. In case of coupled design, the same internal element of form can perform multiple functions. The *system concept representation framework* proposed in this dissertation opens up the possibility to *assess* the element (subsystem, component) criticality based on the number of functions that specific element (subsystem, component) acts on. In particular, we may quantify that the internal element of form wing (entry 22B2) acts on three processes: lifting (entry 20B2), stabilizing pitch (entry 20B4), and stabilizing yaw (entry 22B5). At the same time, the internal element of form fuselage (entry 22B1) acts on two processes: carrying (entry 20B1) and lifting (entry

20B2). Such assessment might be a starting point for evaluation of cost of change of one subsystem to another one.

There is a rule of how to represent multiple connections between the internal elements of form (entries 22), internal processes (entries 20), and internal operands (entries 17). Consider a Figure 3.20. Now we are dealing with connections between entities, either the lines with arrows or the lines with open circles, presented in Figure 3.20.

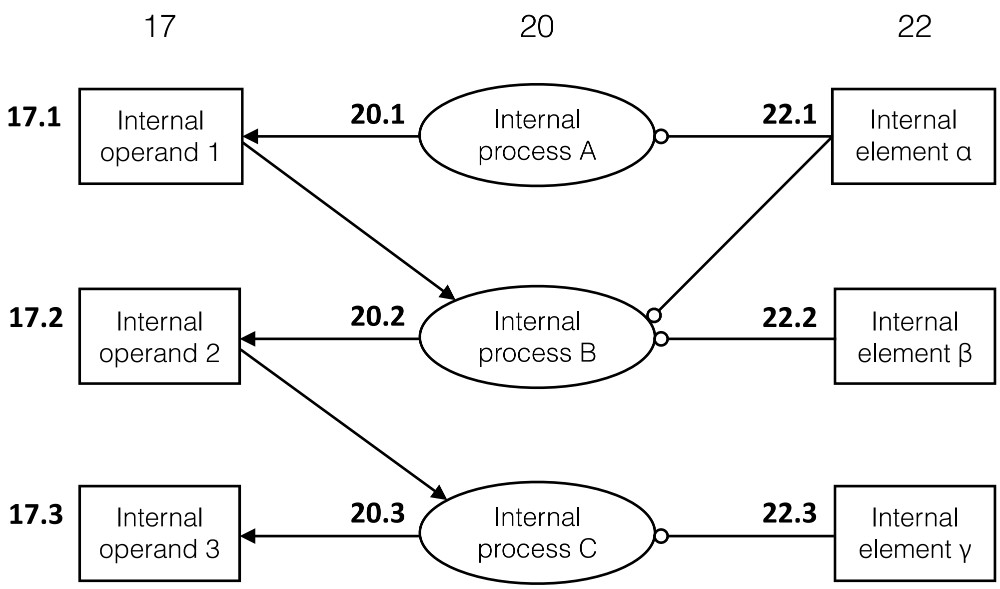


Figure 3.20: Generic view of relationships between internal elements of form, internal processes & internal operands

The common approach is shown in Figure 3.21 that explains this process in a generic form. Let us assume that there are three internal elements of form (α, β, and γ), three internal processes (A, B, and C), and three internal operands (1, 2, and 3) – See

Figure 3.20. As such, they all form the instances of the same class of concept framework entries:

- Internal operand 1 is the entry 17.1 of the concept framework
- Internal operand 2 is the entry 17.2 of the concept framework
- Internal operand 3 is the entry 17.3 of the concept framework
- Internal process A is the entry 20.1 of the concept framework
- Internal process B is the entry 20.2 of the concept framework
- Internal process C is the entry 20.3 of the concept framework
- Internal element α is the entry 22.1 of the concept framework
- Internal element β is the entry 22.2 of the concept framework
- Internal element γ is the entry 22.3 of the concept framework

In case of coupled design, the same internal instrument can perform the number of internal functions. In total the information contained in Figure 3.20 reveals seven relationships, which are summarized in Figure 3.21. This Figure explains how the relationships can be represented by means of the proposed concept framework. For example, the internal element of form α is the instrument of the internal process A that is assigned to the internal operand 1 (relationship 1 in Figure 3.21). At the same time the internal element of form α is the instrument of the internal process B that is assigned to the internal operand 2 (relationship 4 in Figure 3.21). The third relationship, connected with the internal element α is that it is the instrument of internal process B, which affects an internal operand 1 (relationship 2 in Figure 3.21). Having this logic for all internal operands, we can identify seven relationships that are highlighted in Figure 3.21. This representation allows the system architect representing any number of connections between the internal elements of form, internal processes, and internal operands.

№	Entry/Instance	Relationships						
		1	2	3	4	5	6	7
17	Internal operand	1	1	1	2	2	2	3
20	Internal process	A	B	B	B	B	C	C
22	Internal element of form	α	α	β	α	β	γ	γ

Figure 3.21: Representation of relationships in concept framework

Using the information from the generic view of relationships, presented in Figures 3.20 and 3.21, we can visualize the involvement of a particular internal element of form into the number of functions. As such, we can see that the internal element of form α is used 3 times, while the internal element of form β is used 2 times, and the internal element of form γ is also used twice. This data is presented in Figure 3.22 for the generic view of the relationships, and in Figure 3.23 for T&W aircraft concept (left) and the BWB aircraft concept (right).

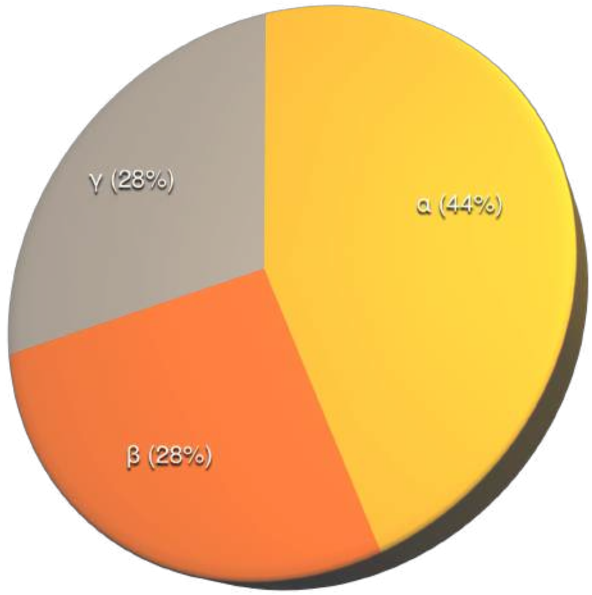
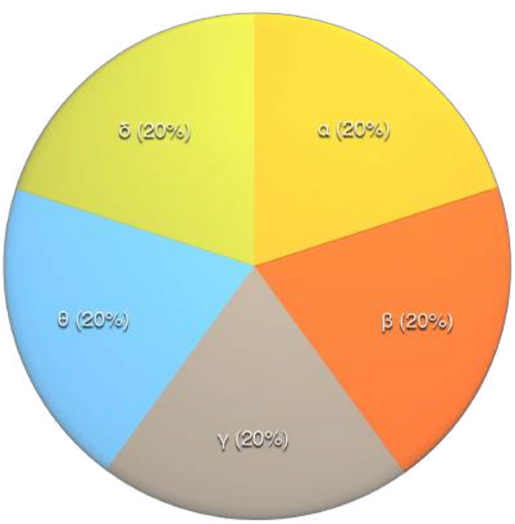


Figure 3.22: The internal element of form usage for the generic view example (α is used 3 times in the example of Figure 3.20; β - 2 times; and γ - 2 times)



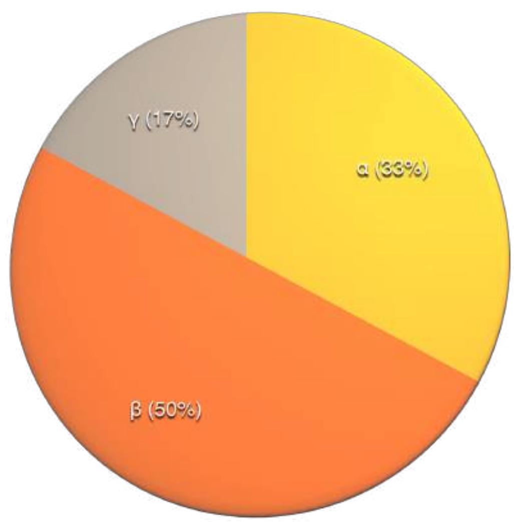


Figure 3.23: The internal element of form usage for T&W aircraft concept (left) and BWB aircraft concept (right). Note that α is the internal element of form *Fuselage*; β - *Wing*; γ - *Engine*; θ - *Horizontal tail*; and δ - *Vertical tail*

The diagram in Figure 3.23 illustrates important information on allocation of the internal elements of form to the internal functions. As such it allows system architect to define the most critical internal elements of form and relationships during the conceptual design phase.

3.7 Proposition V. Concept of Operations (ConOps)

In the previous sections we discussed four propositions, namely, (I) the stakeholders, (II) the solution-neutral problem statement, (III) the solution-specific solution statement, (IV) the integrated concept. The section 3.7 discusses the fifth proposition of the framework, a Concept of Operations (ConOps). Similarly to the previous sections, the sub-section 3.7.1 discusses the literature on the topic and rationale for including a ConOps into the concept framework. Sub-section 3.7.2 demonstrates the generic model-based representation of ConOps, and sub-section 3.7.3 presents ConOps for the aircraft concept.

3.7.1 Concept of Operations (ConOps): Literature and Rationale

The fifth proposition of the proposed *system concept representation framework* is the “concept of operations” (ConOps), which includes the ConOps itself (entry 26), the human operator (entry 27), and the context (entry 28) as indicated in Table 3.1.

According to NASA Systems Engineering Handbook, the concept of operations (ConOps) “... describes the overall high-level concept of how the system will be used to meet stakeholder expectations, usually in a time sequenced manner”. The Department of

Defense Architecture Framework (DODAF) also uses one of the viewpoints, namely, the Operational Viewpoint-1: High-Level Operational Concept Graphic (OV-1) as the summary of ConOps. The document states that “the purpose of OV-1 is to provide a quick, high-level description of what the architecture is supposed to do, and how it is supposed to do it”. By operator, the 27th entry of framework, we imply the person or group of people who will operate the concept.

It is especially noteworthy that one of the intended usages of the OV-1 is to “put an operational situation into context”. In context, we include the concepts that surround our central concept. There are several aspects to these surrounding concepts. The first is of other systems (concepts) that must be present in order to deliver the value of our central concept. For example, an aircraft will not take-off without airports providing infrastructure. If we do not consider the 5th proposition in the concept framework, we will lose all this essential information. For example, “airport” is not part of the “aircraft” concept itself, but must be considered when we talk about how the aircraft will operate in the real environment. At this level, the surrounding concepts are akin to the elements of the system-of-systems that must be present to deliver value.

At a larger radius, the context contains elements that inform design and operations, but are not essential for function. These include the environment in which the central concept operates. For example, the aircraft might be in a valley with strong winds, or might land routinely on snow and ice covered runways. Therefore, the ConOps proposition of framework supports the system architect with information on how the concept will operate, who will operate it, and what other concepts will surround it.

3.7.2 Model-Based Representation of ConOps

The generic representation of ConOps is shown in Table 3.11 and Figure 3.24. The idea of context can be explained by means of *accompanying systems* (Crawley et al. 2015), the generic representation of which is shown Figure 3.24. These systems are important for the system to deliver value, however each one of such systems is not part of the system that is under system architect's development. The *whole product system* is a combination of accompanying systems and the system of interest.

Table 3.11. Entries of proposition V (Concept of Operations)

Concept of Operations	
26	Concept of Operations
27	Operator
28	Context

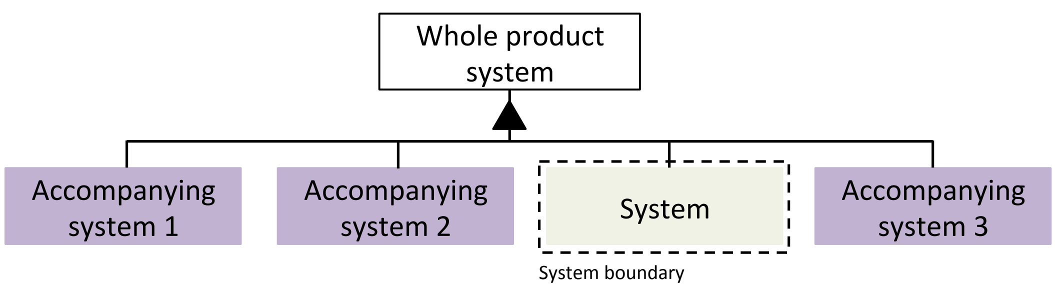


Figure 3.24: Whole product system example

3.7.3 Applying the Methodology to the Running Example of Aircraft Concept

In Figure 3.25 the entries of ConOps (the fifth proposition) are gathered together and presented. It includes the ConOps itself (entry26), the operator (entry 27), and the context (entry 28).

Figure 3.25 is a simplified model of sequence of operations of a transport aircraft (the 26th entry of the framework). This model-based graph can support system architects and decision makers with high-level information related to operations and the accompanying systems: runway, airport terminal, airport traffic control, maintenance (entries 28). The pilots are the operators, entry 27 of framework. All these elements aggregate to the whole product system.

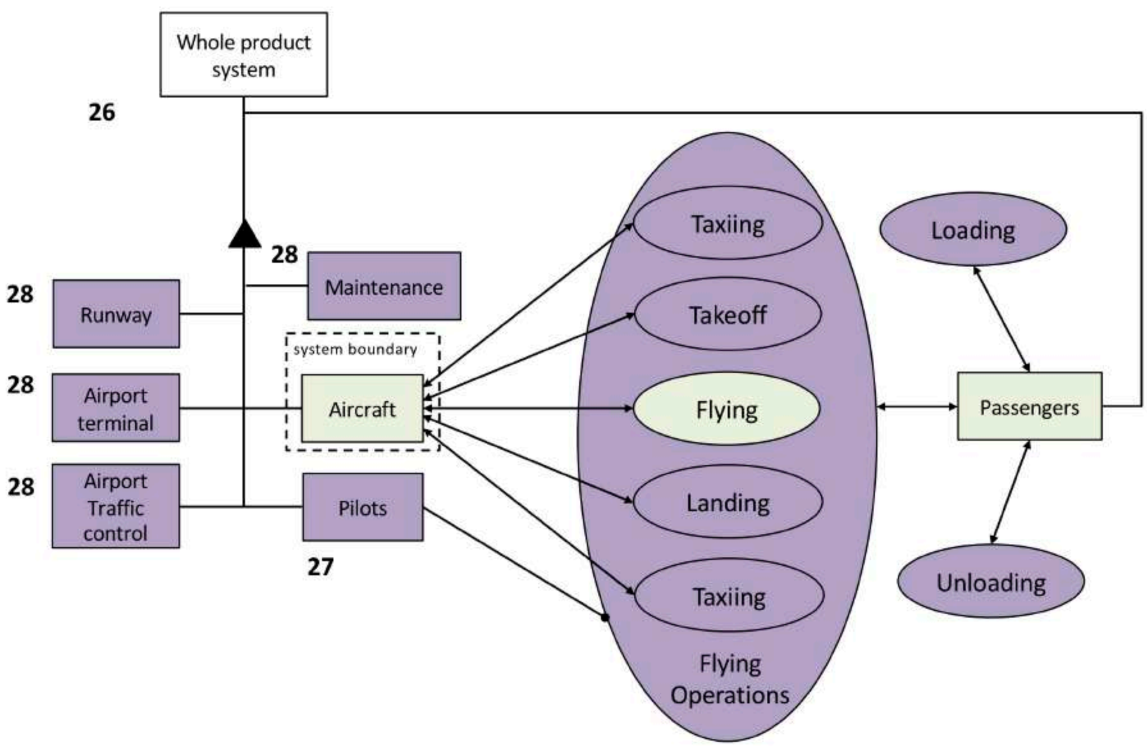


Figure 3.25: Concept of Operations for aircraft concept

3.8 Concept Representation in SysML Modeling Language

In this sub-section we demonstrate that the *system concept representation framework* can also be represented in SysML modeling language. Thus, the objective of this sub-section is to demonstrate that the framework has the universal principles regardless the chosen modeling language.

We will use SysML in order to represent one of the concepts presented in this chapter, namely, T&W aircraft concept. We will show that the combination of recursively used SysML diagrams enables to represent the concept and to keep track of the most important information about T&W aircraft concept.

As we discussed previously, the first two entries of the framework are the stakeholders (entry 1) and their needs (entry 2). The block definition diagram (bdd) of SysML can be perfectly used to represent the first proposition of the concept framework dealing with stakeholders. We have shown earlier that in case of aircraft concept the stakeholder is traveler, while the need of stakeholder is to get somewhere – see Figure 3.26, which is the SysML representation of the stakeholders information.

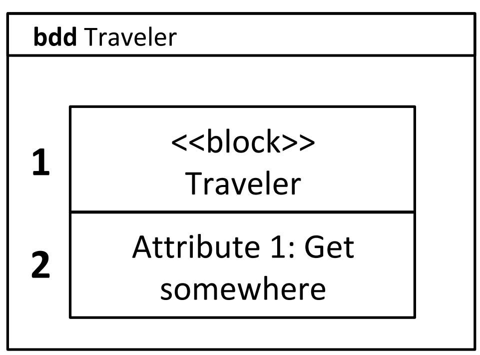


Figure 3.26: Stakeholders (I) proposition for aircraft concept in SysML

The second proposition of the concept framework is dealing with the solution-neutral environment, in which the solution-neutral operand (entry 3), its value attribute (entry 4) and other attribute (entry 5), solution-neutral process (entry 6), and solution-neutral process attribute (entry 7) are presented. The core idea behind the solution-neutral environment is that the instrument of function's execution is not known – this is the reason why we see the word "Form" at the top of Figure 3.27. The function is transporting traveler, while the attributes are safely (related to transporting) and location and number (related to traveler). This information is summarized in Figure 3.27.

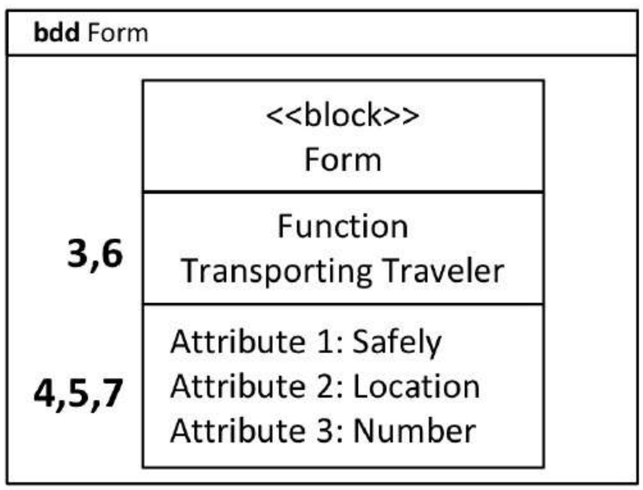


Figure 3.27: Solution-neutral (II) proposition for aircraft concept in SysML

The third proposition of concept framework is the solution-specific environment, in which the information about solution-specific operand and process, their attributes, the generic form and specific form and their attributes summarized. This covers the entries 8 to 16 of the framework.

The solution-specific proposition for T&W aircraft concept is presented in Figure 3.28 in bdd diagram of SysML.

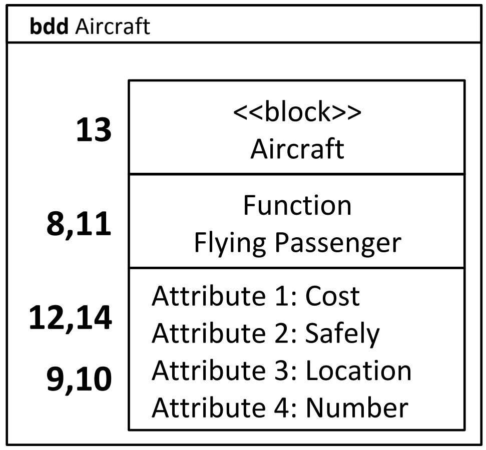


Figure 3.28: Solution-specific (III) proposition for aircraft concept in SysML

The fourth proposition of the framework is the integrated concept. It deals with the specific form and its decomposition into internal elements of form each one of which performs its own internal process and acts on internal operand. This covers the entries 17 to 23 of the framework. Applied to T&W aircraft concept, the integrated concept in SysML has a view presented in Figure 3.29.

Figure 3.29 informs the system architect that T&W aircraft concept is decomposed into five internal elements of form: fuselage, wings, engine, vertical tail, and horizontal tail. Under the function section we have the information about internal processes – for example, "carrying passengers".

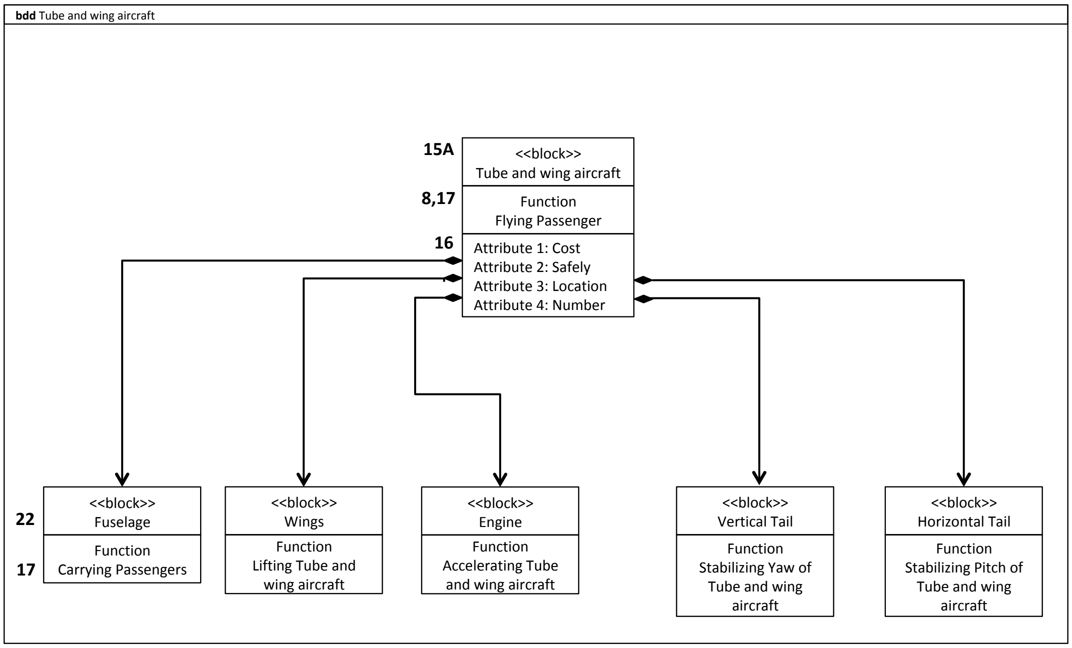


Figure 3.29: Integrated concept (IV) proposition for aircraft concept in SysML

In order to present the structure (entry 24) and interactions (entry 25) information of the framework the different type of SysML diagrams should be chosen – the internal block diagram (ibd). This diagram enables the connection of the internal elements of form to each other, representing the information about structural relationships (for example, engine "attached below" wings) and functional relationships (for instance, engine "provides thrust" wings). The detailed information summarized in Figure 3.30.

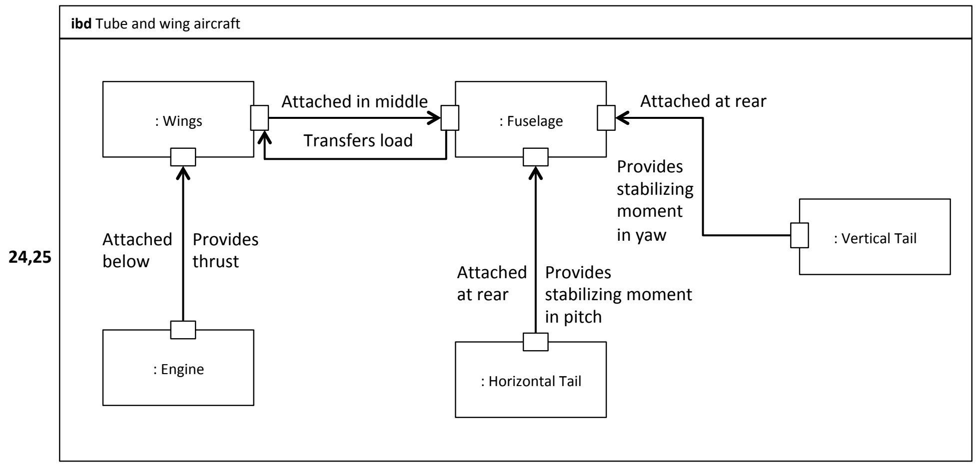


Figure 3.30: Structure and interactions for aircraft concept in SysML

The 5th proposition of the framework can be represented by another type of SysML diagram – activity diagram presented in Figure 3.31. From this diagram the one can see the concept of operations (entry 26) and its context (entry 28) – operator (entry 27) "pilots", the concept itself "tube and wing aircraft", supporting systems "runway", "airport terminal", and "airport traffic control".

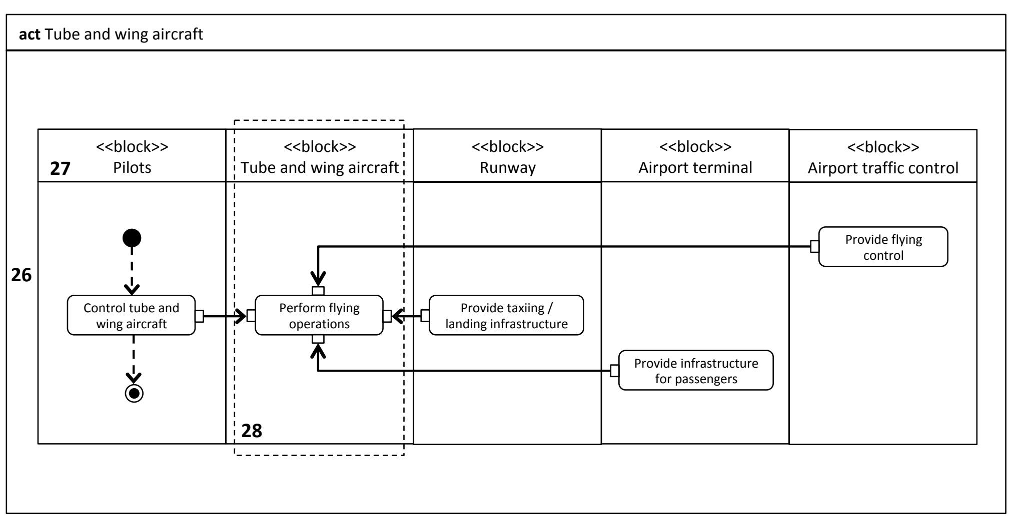


Figure 3.31: Concept of Operations (V) proposition for aircraft concept in SysML

3.9 Applying a System Concept Representation Framework to a Small Case Example

Our intention to present the applicability of the proposed framework to a small case example is motivated by desire to illustrate the utility of such representation. As such case study we have chosen a coffee maker example, demonstrating the alternative concepts development for this relatively simple system.

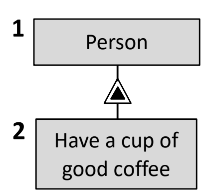
3.9.1 System Concept Representation Framework for a Coffee Maker

In the worked example of a coffee maker of Figure 3.32, the need "have a cup of good coffee" is shown as an attribute of the stakeholder "person". The information is presented in two formats – a table with both a prompt and the worked example (Figure 3.32(a)) and the OPM diagram – Figure 3.32(b). Note that the numbers "1" and "2" in Figure 3.32(b) correspond to the entries 1 and 2 of the framework presented in Table 3.1 – "stakeholders" and "need".

Proposition I: Stakeholders

1	**Stakeholders**	**Person**
2	Need	Increase enjoyment level

(a)



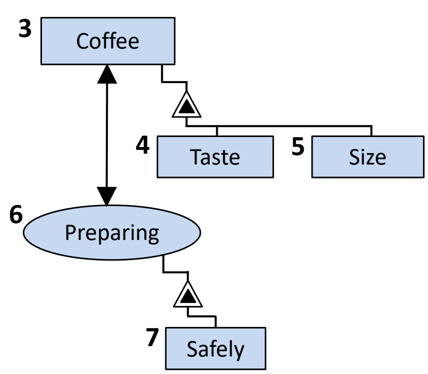
(b)

Figure 3.32: Stakeholders (I) proposition for a coffee maker concept: table (a) and OPM (b) views

The second proposition of the *system concept representation framework* is the solution-neutral environment, which rationalizes the problem we are trying to solve. For a coffee maker "Coffee (the liquid)" is the solution-neutral operand, and taste is the value-related attribute. The process of preparing a coffee changes its taste, this creates a value, thus taste is a value-related attribute. The other attribute is "size". The process of preparing should be performed "safely", this is an attribute of solution-neutral process. Thus, the solution-neutral statement is "safely prepare tasty coffee of some specified amount". This information is summarized in Figure 3.33.

Proposition II: Solution-neutral environment (Problem statement)		
3	**Solution-neutral operand (SNO)**	**Coffee**
4	SNO value attribute	Taste
5	SNO other attribute	Size
6	**Solution-neutral process (SNP)**	**Preparing**
7	SNP attribute	Safely

(a)



(b)

Figure 3.33: Solution-neutral (II) proposition for a coffee maker concept: table (a) and OPM (b) views

As we have seen in Figure 3.6 and Figure 3.7, the conceptual design is associated with the specialization process and with a movement from the solution-neutral environment to the solution-specific environment.

In our coffee maker example, the solution-neutral operand “Coffee” further specializes to “Ground coffee” (see Figure 3.34). The solution-neutral process “Preparing” specializes to “Extracting” that is further specializes into two conceptually different processes at the level of solution-specific process: “Pressurizing” (entry 11.1) and “Steeping” (entry 11.2). The generic form “Coffee maker” is an instrument of executing the function “Extracting ground coffee”. The generic form can be further specialized to “Espresso Machine” (entry 15.1) and “French Press” (entry 15.2). All this information is summarized in Figure 3.34. The generic form, the specific form, and their attributes are also part of the third proposition of the framework shown in Table 3.1.

In Figure 3.34, we can capture the first rule that should be followed during the creation of the model-based diagram: the attributes are inherited “from left to right” (between solution-neutral and solution-specific) and “from top to down” (between the generic form and specific form). This has been discussed in sub-section 3.5.3. For example, the solution-specific process “Extracting” has not only the attribute “Time” (entry 12), but also inherits the attribute “Safely” from the solution-neutral environment.

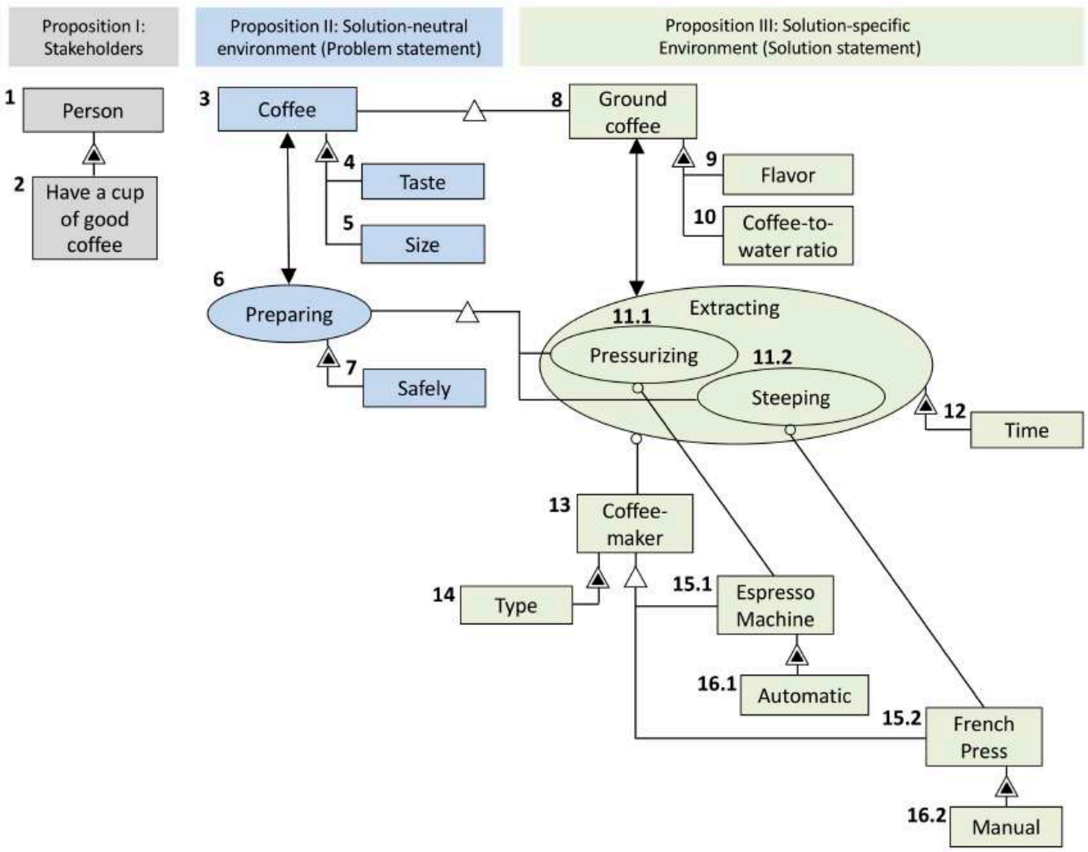
Right-hand side of Figure 3.34(a) illustrates two alternative solutions appearing at the level of specific form - an Espresso Machine and a French Press.

Proposition III: Solution-specific Environment (Solution statement)			
8	**Solution-specific operand (SSO)**	**Ground coffee**	
9	SSO value attribute	Flavor	
10	SSO other attribute	Coffee-to-water ratio	
11	**Solution-specific process (SSP)**	**Extracting**	
		11.1	**11.2**
		Pressurizing	**Steeping**
12	SSP attribute	Time	
13	**Generic Form**	**Coffee Maker**	
14	Generic Form attribute	Type	
15	**Specific Form**	**15.1**	**15.2**
		Espresso Machine	French Press
16	Specific Form attribute	**16.1**	**16.2**
		Automatic	Manual

Espresso Machine

French Press

(a)



(b)

Figure 3.34: Conceptual design for a coffee maker concept: table (a) and OPM (b) views

The fourth proposition of the *system concept representation framework* is the integrated concept, which is associated with the decomposition relationships applied to the specific forms. As such, Figure 3.35 *encodes* the integrated concept for Espresso Machine concept, while Figure 3.36 – *encodes* integrated concept for French Press concept.

From Figure 3.35 we notice that the specific form “Espresso Machine” is decomposed into its formal elements: “Portafilter”, “Heating element”, “Pump”, “Boiler”, and “Body” (see Figure 3.35). Each internal element of form is an instrument of the internal processes – “Transferring”, “Heating”, “Moving”, “Holding”, and “Containing”, respectively. This diagram contains important information that, as we will see later, provide a means to analyze conceptual differences between alternative solutions (e.g. the Espresso Machine and French Press). Figure 3.35 also contains information attributes which could reflect operational specification or design parameters. For example, from Figure 3.35 we may notice that the process “Heating” has an attribute of temperature “90-96°C”. Another attribute “30 ml” is associated with a “Liquid coffee”. We may also see that a pump moves water under a pressure of “9-15 bars”. The overall process “Pressurizing” takes about 2 minutes. In general, such attributes do not define conceptual difference, but are an outcome of conceptual or detailed design differences.

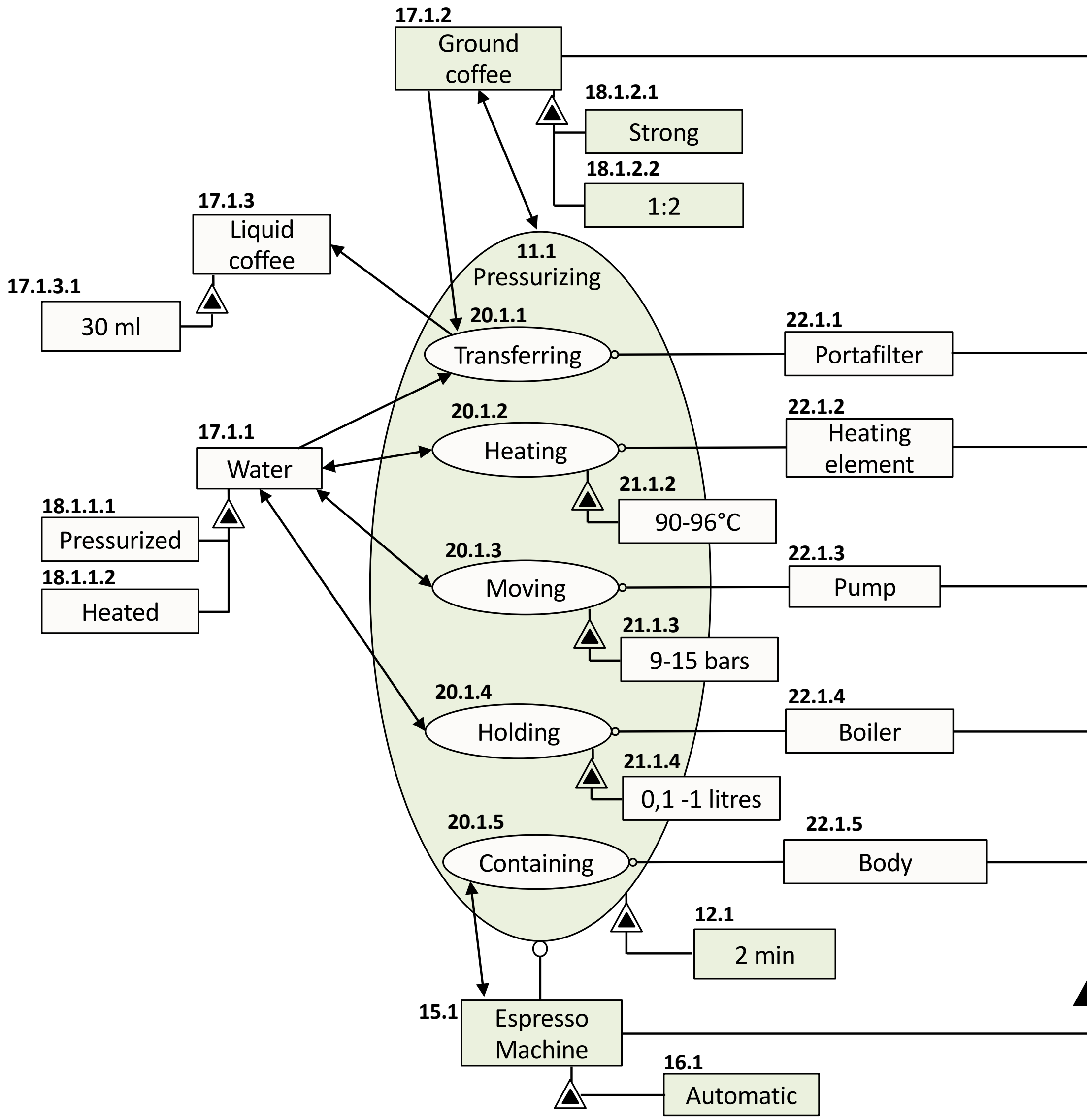


Figure 3.35: Integrated concept for Espresso Machine concept in OPM view

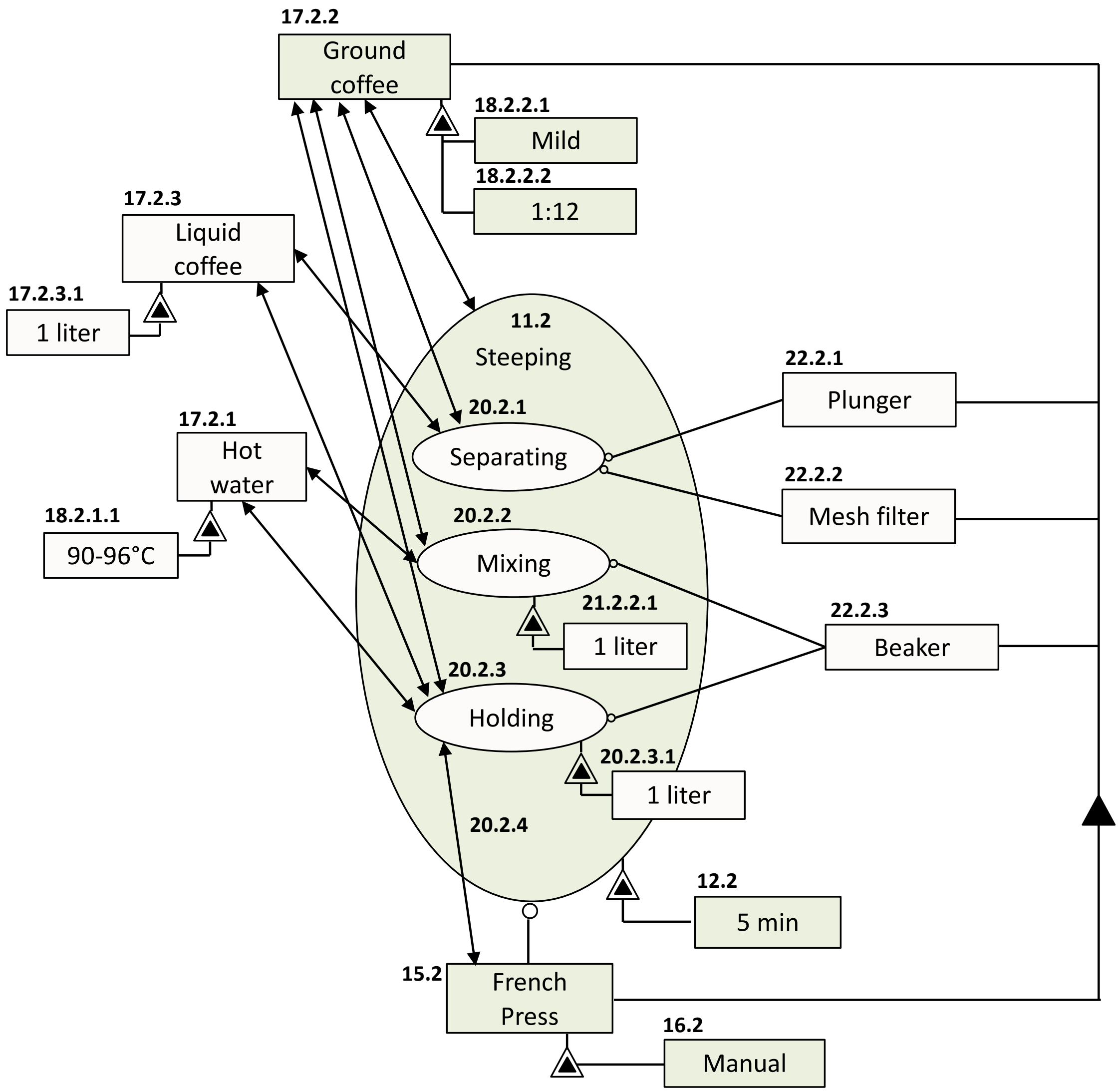


Figure 3.36: Integrated concept for French Press concept in OPM view

The specific form "French Press" is decomposed into its formal elements: "Plunger", "Mesh filter", and "Beaker" (see Figure 3.36). Each internal element of form is an instrument of the internal processes – "Separating", "Mixing", and "Holding", correspondingly. The operational attributes are "1 liter" for an operand "Liquid coffee", and “5 minutes” for a process “Steeping”. We should also note that an operand “Hot water” with an attribute “90-96°C” lies outside of product boundary.

As it was discussed in section 3.6, the other aspects of the integrated concept are structure (entry 24 in Table 3.1) and interactions (entry 25 in Table 3.1). These entries are important, as they inform the system architect which elements are connected and what is exchanged between them.

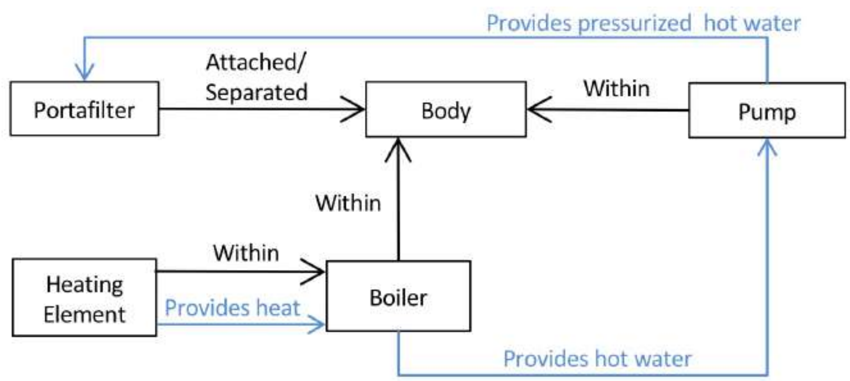


Figure 3.37: Structure and interactions for Espresso Machine concept

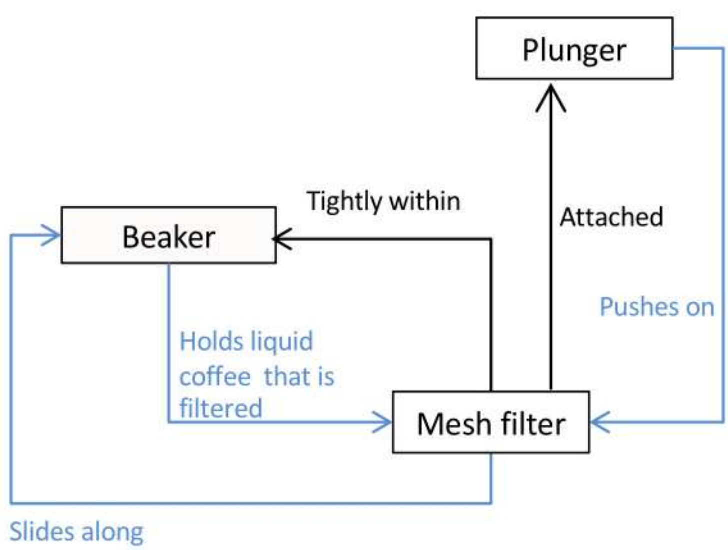


Figure 3.38: Structure and interactions for French Press concept

For the running example, Figure 3.37 and Figure 3.38 represent the aspects of the integrated concept that capture formal relationships and functional relationships. Figure 3.37 represents the formal/functional relationships information for the Espresso Machine concept, while Figure 3.38 – for the French Press concept. The system architect, for example, can see that the formal connection and alignment of the heating element (Figure 3.37) is “Within” a boiler. The functional relationships between heating element and boiler (Figure 3.37) is “Provides heat”. This representation of formal and functional relationships allows clear visualization of the information about formal and functional interactions.

The fifth proposition of the *system concept representation framework* is ConOps, shown in Figure 3.39 for a simplified model of sequence of operations of a coffee maker (the 26th entry of the framework presented in Table 3.1). This model-based representation can give system architects high-level information relating to operations and the context: person who is the operator and mug which is part of context (entries 27 and 28).

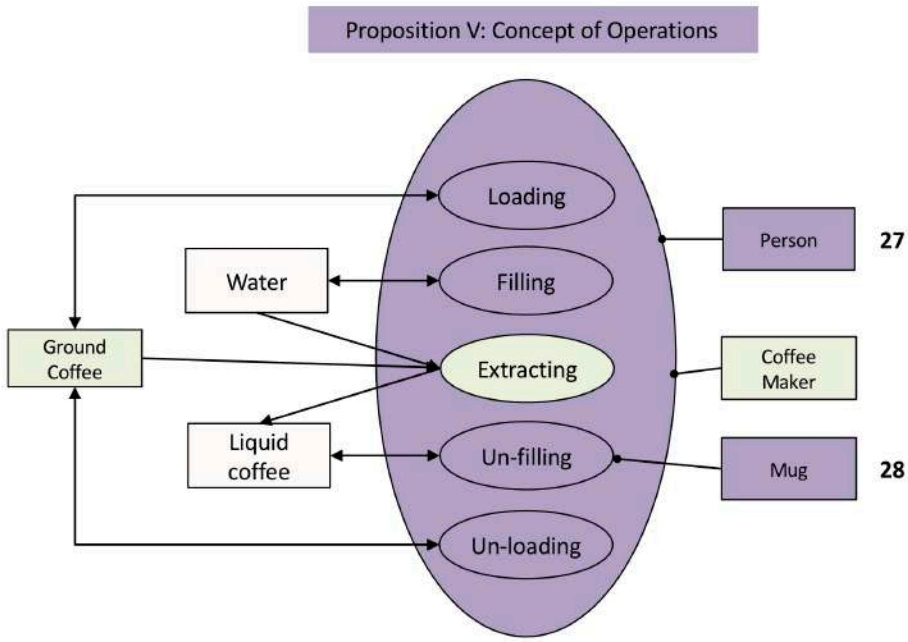


Figure 3.39: Concept of Operations for Coffee Maker concept

Figure 3.39 describes the operations associated with preparing a coffee. The first step performed by a person is to load ground coffee. After that a person should fill the coffee maker with water. As we know from the solution-specific environment, the function of "Coffee maker is "Extracting ground coffee". Both ground coffee and water are used for brewing a liquid coffee, which is later un-filled from the coffee maker and transferred to a mug. Finally, the grounds are un-loaded. This is a simplified version of a concept of operations.

3.9.2 Comparative Results of Alternative Concepts for a Coffee Maker

Comparing Figures 3.34, 3.35 and 3.36 we sense the ability to *encode* alternative concepts in the framework and to perform an analysis of differences between those concepts. The first important observation is that there is little difference in Figure 3.34. The stakeholder and solution-neutral information is identical, as is the operand information (8, 9 and 10). Only entries 11, 15 and 16 are different, and these are really just labels for the concept. You have to examine Figures 3.35 and 3.36 to really learn how a coffee machine works.

The next important result is that both concepts (the Espresso Machine and French Press) have conceptually different decomposition, in the sense that functions are present or missing in the decomposition. For example, for Espresso Machine there is a heating element that heats water. This heating process happens within the system, as the heating element is a part of an espresso machine concept. In case of French Press concept, we do not see any kind of heating element, as the water supplied to a beaker is already hot. The process of heating a water is executed outside of system boundary. This is an important distinction provided by the model-based representation, because this result helps a system architect to draw a line indicated functions within and outside the system. This has broad impact in systems engineering process, for example on sourcing and interface control.

Another conceptual difference is a presence of a pump for the Espresso Machine. A pump performs a function of energizing water by pressurizing it. A system architect could even specify the pressure of 9-15 bars that is associated with such water in a system. For a French Press, a related conceptual element is “Plunger” and the source of energy is the operator. There is a conceptual difference in energizing the water.

There is also a conceptual difference in the brewing process. In the Espresso Machine, the water/coffee moves through the portafilter, while in the French Press, the screen runs through the water/coffee.

Contrasting the two coffee makers, we see patterns that are discernable. The "simpler" French Press has fewer internal functions, since heating the water and providing power are both outside the system. But compared to the Espresso Machine, the internal functions of the French Press are more interconnected.

From Figure 3.35 we may see that the Espresso Machine concept is associated with a process "Pressurizing". About 2 minutes (attribute of a process) are required to prepare a liquid coffee of 30 ml (1 shot of espresso). For the French Press a process "Steeping" has an attribute "5 min", thus in French Press it takes more time to make a cup of coffee of the same amount as it is in Espresso Machine. However, in case of French Press the volume of liquid coffee that you could prepare for 5 min could reach 1 liter, instead of 1 (30 ml) or 2 (60 ml) shots that you could get in Espresso Machine. Thus, the utility of the proposed approach is that it provides this essential information to system architect or design team.

Having these diagrams in hands, the system architect or design team can easily see that if you have one stakeholder with a need "Have a cup of good coffee", an Espresso Machine is the option which will provide you a coffee faster. However, if you have a group of people, a French Press would look a more preferable option.

3.10 Concept Framework as a Set of Questions

Since the proposed framework contains relatively large number of entries, twenty-eight, which are spread among five propositions, in some cases it might be useful to consider a conceptual design process as the answers on the set of questions presented in Figure 3.40. These questions are based on the framework of Table 3.1.

Conceptual design → Decomposition →

	Stakeholders / Solution-neutral environment (Problem statement)		Solution-specific environment (Solution statement)		Integrated concept / Concept of Operations
1	Who are the **stakeholders?**				
2	What is the need of stakeholders?				
3	What is **solution-neutral operand (SNO)**?	**8**	What is **solution-specific operand (SSO)**?	**17**	What are the **Internal Operands (IO)**?
4	What is the value attribute of SNO?	9	What is the value attribute of SSO?	18	What are the value attributes of IO?
5	What is the other attribute of SNO?	10	What is the other attribute of SSO?	19	What are the other attributes of IO?
6	What is **solution-neutral process (SNP)**?	**11**	What is **solution-specific process (SSP)**?	**20**	What are the **Internal Processes (IP)**?
7	What is the attribute of SNP ?	12	What is the attribute of SSP?	21	What is the attribute of IP ?
		13	What is the **Generic Form**?	**22**	What are **Internal Elements of Form (IEoF)**?
		14	What is the attribute of Generic Form?	23	What are the attribute of IEoF?
		15	What is the **Specific Form**?	24	What is the structure?
		16	What is the attribute of Specific Form?	25	What are the interactions?
				26	What is the Concept of Operations?
				27	Who is the operator?
				28	In which context the concept operates?

Figure 3.40: Concept Framework as a Set of Questions

Each one of these questions is related to a particular proposition. In order to distinguish which question is related to which proposition, the colored labels are supporting the system architect. The purpose of Figure 3.40 is to support system architect keeping track of identification of the essential information about the concept of interest.

3.11 Summary

In this Chapter we presented a model-based *system concept representation framework*, which *encodes* the core information about concept and provides a means to represent a concept in a digital environment. Such a framework is built upon five propositions rooted in systems engineering and design theory. These propositions include (I) the stakeholders, (II) the solution-neutral problem statement, (III) the solution-specific solution statement, (IV) the integrated concept, and (V) the concept of operations (ConOps). The definitions of the key entries of the proposed framework and their representations in digital format were provided. The framework could contribute to the model-based conceptual design (MBCD) initiative, aiming at the development of methodology for model-based applications to the exploratory and concept stages. This framework could also support the system architect to *generate* the novel concepts in a systematic way. In this Chapter we answered on three research questions of the Thesis: What information about system concept is required in order to have a representation of a concept? How the information required for system concept representation can be encoded in a model-based manner to support system concepts and their alternatives development? How the information encoded in system concept representation framework can support the quantitative assessment of formal conceptual similarity between alternative concepts?

Throughout the Chapter and following the ontology and semantics of OPM, we applied the proposed model-based concept framework to an aircraft concept, in which we started from the stakeholders proposition and ended up with ConOps proposition. We have demonstrated that the proposed framework might support a *formal analysis*, such as conceptual similarity assessment. The graphical notation of the key entries makes easily visible conceptual differences, and could support the system architect with the development of new concepts and analysis of existing ones. The DSM-based methods provide a quantitative tool for these types of analysis.

Another outcome of this Chapter is that the framework could be used to support decision makers by capturing all of the alternative concepts that might be developed based on stakeholders' need. These alternatives and the decisions that distinguish them, could be presented in a model-based manner. In Chapter 3 we demonstrated the small case example, a coffee maker concept development.

In the following Chapter 4 the proposed methodology will be validated through the analytical surveys (patents, urban architectural patterns, and software patterns). After this we will demonstrate the utility of the proposed approach on two space-related case studies – the suborbital human spaceflight missions presented in Chapter 5; and the space communications missions to be discussed in Chapter 6.

Chapter 4. Validating System Concept Representation Framework Through the Analytical Surveys

4.1 Introduction to Chapter

In previous Chapter we presented the model-based *system concept representation framework* that is comprised of 28 entries spread among 5 propositions. According to the Design Research Methodology (DRM) used in this Thesis, Chapter 1 and Chapter 2 correspond to the Descriptive Study I stage of DRM, while Chapter 3 reflects Prescriptive Study stage of DRM framework (Blessing and Chakrabarti 2009). The goal of Chapters 1-2 is to assess the status of the research, while the purpose of Chapter 3 is to prescribe the methodology to be used in the following Chapters.

In Chapter 4 we validate the proposed framework through a wide variety of analytical surveys: patents, urban architectural patterns (Alexander 1977), and software patterns (Gamma et al. 1995) – see Figure 4.1.

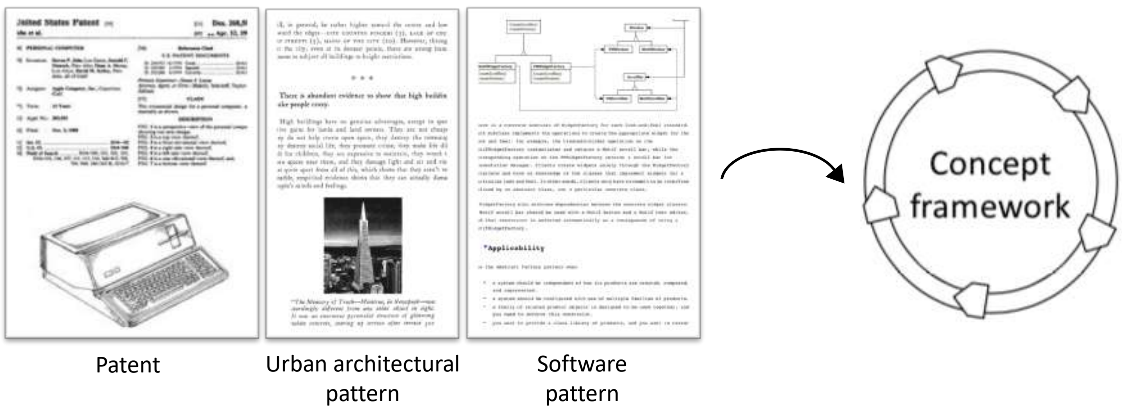


Figure 4.1: Analytical surveys of Chapter 4

We chose these quite different analytical surveys, as we observed that they should contain the core information about concept behind system, pattern, or event they represent. Thus, by mapping an analytical survey to the proposed framework we can validate it and examine how well the entries of framework are spread among specific sample under examination. According to DRM terminology, Chapters 4-6 are dedicated to the Descriptive Study II stage.

Table 4.1 summarizes the information about number of samples chosen for small-N and large-N studies of the analytical surveys.

Table 4.1: The number of samples for small-N/large-N studies for each analytical survey

Study	Analytical survey					
	Patents		Urban architectural patterns		Software patterns	
	Section 4.4.1 of Chapter					
Small-N analysis	8	2 biological	9	3 Towns	3	1 creational pattern
		2 thermodynamic		3 Buildings		1 structural pattern
		2 electro-mechanical				
		2 software		3 Construction		1 behavioral pattern
Study	Section 4.6.1 of Chapter					
Large-N analysis	25	Suborbital human spaceflight	27	9 Towns	12	4 creational patterns
				9 Buildings		4 structural patterns
				9 Construction		4 behavioral patterns

The purpose of small-N study for patents is to explore the structure and contextual information contained in patents, and to map eight randomly selected US patents to the proposed concept framework. The different keywords were used in order to choose samples and to differentiate types of patents from each other. The chosen samples represent four quite different types of patents: biological, thermodynamic, electro-mechanical, and software. The intersection of "small-N analysis" row and "Patents" column in Table 4.1 indicates that we have chosen 2 samples for each type of the patents, totaling 8 patents. During the small-N analysis of patents (as well as other analytical surveys) we were aiming at identification of whether or not the specific entry of framework (1 to 28) exists in patent (urban architectural pattern, or software pattern) which we analyze.

At the next step we generalized the observation conducting the large-N study, in which we analyzed 25 selected patents. This set of samples was dedicated to a specific topic and was found in Google Patents using the "suborbital spaceflight vehicle" keywords. 85 patents and patent applications were found, out of which 25 the most cited patents were chosen for further examination. The list of 85 patents is provided in Appendix A. Similarly to the methodology used in the small-N study, during the large-N study we mapped 25 selected patents to the proposed framework. We believe that our framework contains the "genome" of a concept – essential information necessary to *encode* the existing concept or to *generate* the novel one. Summarizing, we may conclude that during large-N study 30% of the full set of found patents was analyzed (25 out of 85).

The next column in Table 4.1 is dedicated to the urban architectural patterns (Alexander 1977). In his book that became widely recognized, Christopher Alexander had explored the pattern language of the architecture. He developed the set of 253 patterns each one of which belongs to one of the three types of architectural patterns: towns (94 patterns), buildings (110 patterns), and construction (49 patterns). 85 patterns out of 253 patterns are marked by two-asterisks. According to Alexander, this implies that "the solution we have stated summarizes a property common to all possible ways of solving the stated problem". In other words, these are the most developed patterns among all 253 ones. 28 patterns are marked by two-asterisks for the towns patterns; 37 patterns - for the buildings patterns; and 20 patterns – for the construction patterns (85 patterns in total). In both – small-N and large-N studies – we have randomly selected the patterns from the set of these 85 patterns. During the small-N study of the urban architectural patterns we analyzed 3 patterns for each one of the categories totaling 9 patterns. In the large-N study we focused on 9 patterns for each category. Thus, there are 27 patterns in total in the corresponding cell of Table 4.1. This means that in total, 42% of patterns were analyzed (36 out of 85).

The last column of Table 4.1 informs us about sampling of the software patterns (Gamma et al. 1995). The authors of the book have taken the Alexander's idea to define a universal pattern language and have applied it to software. They created three categories of patterns: creational patterns (5 patterns in total), structural patterns (7 patterns in total), and behavioral patterns (11 patterns in total). During the small-N analysis we have randomly selected 1 pattern per each category, thus in total we analyzed 3 patterns. For the large-N study we explored 4 patterns per each category, so in total 12 patterns were analyzed. Thus, in total 65% of software patterns were analyzed (15 out of 23).

For all three analytical surveys the purpose of large-N examination was to figure out whether the outcomes of the large-N examination correlate with results of the small-N study. During the large-N study we also focused on "yes/no" type of questions: whether or not the specific entry of the framework present in specific patent, urban architectural pattern, or software pattern.

The remainder of this Chapter is organized as follows. In Section 4.2 we provide the criteria based on which the entries of *system concept representation framework* were filled in for the specific patents, urban architectural patterns, or software patterns. In Section 4.3 we overview the analytical surveys. In sub-section 4.3.1 we discuss patents and their structure; in sub-section 4.3.2 we review urban architectural patterns and their structure; and in sub-section 4.3.3 we discuss software patterns and their structure.

Section 4.4 explains the methodology of mapping analytical surveys to framework for small-N study. Sub-section 4.4.1 is dedicated to the sampling procedure for analytical surveys for small-N study; sub-section 4.4.2 explains the process of mapping analytical surveys to framework. The results of small-N study are presented in section 4.5: sub-section 4.5.1 discusses the results of mapping patents to the proposed framework; sub-section 4.5.2 – of mapping urban architectural patterns; and sub-section 4.5.3 – of mapping software patterns. Section 4.6 provides a methodology of mapping analytical surveys to framework for large-N study. The process of sampling analytical surveys for large-N study is explained in sub-section 4.6.1; and the procedure of mapping analytical surveys to framework – in sub-section 4.6.2. The results of large-N study are demonstrated in section 4.7: for patents (sub-section 4.7.1), urban architectural patterns (sub-section 4.7.2), and software patterns (sub-section 4.7.3). Cross-cutting results for the *system concept representation framework* are discussed in section 4.8. Summary and conclusions are provided in section 4.9.

4.2 Criteria for Including Data into Concept Framework

Table 4.2 summarizes the criteria based on which the specific information is considered relevant for inclusion as the entries of concept framework. This Table aims at reducing the potential subjectivism in choosing data.

Table 4.2: Criteria for including the data from patent/pattern into the concept framework

Concept framework entries	Criteria
1. Stakeholders	Whether the stakeholders are mentioned in the text (see the definition of stakeholders in 3.3.1). The stakeholders are usually represented by noun – "organization", "user", "community", etc.
2. Needs	Whether the stakeholders' needs are mentioned in the text (see the definition of needs in 3.3.1). The needs are usually represented by verb + noun and are stated in an abstract way – "have fun", "create a healthy environment", etc.
3. Solution-neutral operand (SNO)	Whether the object, whose state is changed by the process – is mentioned in the text (for the detailed definition of operand see 3.4.1). The operand is usually represented by noun. The fact that it is solution-neutral implies that the operand is mentioned in an abstract way. The example is "person", etc.
4. SNO value attribute	Whether the value-related attribute is mentioned in the text (for the detailed definition of value attribute see 3.4.1). It can be represented by noun ("location", "number"), or adjective ("safe"), and the choice of the attribute depends on the context.
5. SNO other attribute	Whether the other attribute is mentioned in the text (for the detailed definition of other attribute see 3.4.1). It can be represented by noun ("speed"), or adjective ("safe"), the choice of the attribute depends on the context.
6. Solution-neutral process (SNP)	Whether the process that acts on operand can be found in the text (for the detailed definition of process see 3.4.1). The process is usually represented by the adjective. The fact that it is solution-neutral implies that the process is mentioned in an abstract way. The examples are "entertaining", "moving", etc.
7. SNP attribute	Whether the attribute is mentioned in the text (for the detailed definition of attribute see 3.4.1). The attribute related to the process is usually represented by the adverb – such as "safely", the choice of the attribute depends on the context.
8. Solution-specific operand (SSO)	The same principles as for №3 are applied. The core difference is that the solution-specific operand is the specialization of the solution-neutral operand. Thus, the example is "passenger", etc. See 3.5.1
9. SSO value attribute	Whether the value-related attribute is mentioned in the text

	(for the detailed definition of value attribute see 3.4.1). It can be represented by noun ("location", "number"), or adjective ("safe"), and the choice of the attribute depends on the context. It can be inherited from №4 or newly appeared.
10. SSO other attribute	Whether the other attribute is mentioned in the text (for the detailed definition of other attribute see 3.4.1). It can be represented by noun ("speed"), or adjective ("safe"), the choice of the attribute depends on the context. It can be inherited from №5 or newly appeared.
11. Solution-specific process (SSP)	The same principles as for №6 are applied. The core difference is that the solution-specific process is the specialization of the solution-neutral process. Thus, the examples are "flying", "floating", etc. See 3.5.1
12. SSP attribute	Whether the attribute is mentioned in the text (for the detailed definition of other attribute see 3.4.1). The attribute related to the process is usually represented by the adverb – such as "safely", the choice of the attribute depends on the context. It can be inherited from №7 or newly appeared.
13. Generic Form (GF)	Whether the generic instrument that executes the function (№11 plus №8) is appearing in the text (for the detailed definition of form/instrument see 3.5.1). The generic form is usually represented by noun. The example is "land vehicle", or "flying vehicle", etc.
14. GF attribute	Whether the attribute is mentioned in the text (for the detailed definition of attribute see 3.4.1). The attribute can be represented by noun ("location", "number"), or adjective ("safe"), and the choice of the attribute depends on the context.
15. Specific Form (SF)	Whether the specific instrument that executes the function (№11 plus №8) is appearing in the text. The specific instrument is usually represented by noun. The core difference comparing with №13 is that specific form is the specialization of generic form. The examples are "car", "train", or "aircraft", "helicopter", etc. See 3.5.1
16. SF attribute	Whether the attribute is mentioned in the text (for the detailed definition of attribute see 3.4.1). The attribute can be represented by noun ("location", "number"), or adjective ("safe"), and the choice of the attribute depends on the context. It can be inherited from №14 or newly appeared.
17. Internal Operands (IO)	Whether we can identify in the text the internal operand, on which the internal process (№20) is acting. Similarly to the other operands, it is usually represented by noun. See 3.6.1
18. IO value attribute	Whether the value-related attribute is mentioned in the text

	(for the detailed definition of value attribute see 3.4.1). It can be represented by noun ("location", "number"), or adjective ("safe"), and the choice of the attribute depends on the context.
19. IO other attribute	Whether the other attribute is mentioned in the text (for the detailed definition of other attribute see 3.4.1). It can be represented by noun ("speed"), or adjective ("safe"), the choice of the attribute depends on the context.
20. Internal Process (IP)	Whether we can identify the internal process, which is acting on internal operand (№17), and which is related to the corresponding internal element of form (№22). Similarly to the other processes, it is usually represented by the adjective. See 3.6.1
21. IP attribute	Whether the attribute is mentioned in the text (for the detailed definition of other attribute see 3.6.1). The attribute related to the process is usually represented by the adverb – such as "safely", the choice of the attribute depends on the context.
22. Internal Element of Form (IEoF)	Whether the Specific Form (№15) is decomposed into internal element(s) of form and if so, whether we can find such decomposed element(s) in the text. IEoFs are usually represented by nouns. See 3.6.1
23. IEoF attribute	Whether the attribute is mentioned in the text (for the detailed definition of attribute see 3.6.1). The attribute can be represented by noun ("location", "number"), or adjective ("safe"), and the choice of the attribute depends on the context.
24. Structure	Whether we can identify the information about how the decomposed elements are physically connected. Usually it is explained in the text with the support of such words as "connected", "attached", "within", etc. See 3.6.1
25. Interactions	Whether we can identify the information about what is exchanged among the decomposed elements. Usually it is explained in the text with the support of such words as "provides power", "sends signal", etc. See 3.6.1
26. Concept of Operations	ConOps embraces the key operational phases of the system in terms of sequence of processes with corresponding operands, on which the processes are acting, and instruments, which are executing the functions (processes plus operands). In the text this information usually explained in the sequence of sentences, that are gradually uncovering the way how the system works. See 3.7.1
27. Operator	Whether we can identify the individual, or organization, or

	machine that operates the system. See 3.7.1
28. Context	Whether we can identify the information about accompanying/supporting systems that are not directly related to our system, but which sheds light on how it is intended to operate. See 3.7.1

4.3 Analytical Surveys Overview

In this section we provide a brief overview of patents (sub-section 4.3.1), urban architectural patterns (sub-section 4.3.2), and software patterns (sub-section 4.3.3). A short introduction to each one of these analytical surveys is provided and the structures of them are explained. This overview informs us that each one of these sources has a rich body of knowledge contained in them, and therefore they must logically contain a description of the concepts underlying them.

4.3.1 Patents

Patents are one of the most common ways to protect the rights of an inventor. According to the United States Patent and Trademark Office (USPTO 2007), a patent is "a property right granted by the Government of the United States of America to an inventor 'to exclude others from making, using, offering for sale, or selling the invention throughout the United States or importing the invention into the United States' for a limited time, in exchange for public disclosure of the invention when the patent is granted" (USPTO 2007). Note that the patent doesn't provide a right to make, use, offer for sale, sell or import, but provides the right to exclude others from making, using, offering for sale, selling or importing the invention.

Patents also have utility to other users. In accordance with the World Intellectual Property Organization (WIPO 2016), "it is estimated that some 70% of the information disclosed in patent documents have never been published anywhere else" (WIPO 2016).

From the perspective of this study, patents represent a large database of publicly accessible documents, describing new inventions in some detail. The patents must logically describe the concept underlying or contained within the invention. Success in demonstrating that the patent can be mapped to the concept framework is therefore a necessary condition to demonstrating the utility of the framework.

US patents are structured according to internationally agreed standards. The information in a patent is presented as combination of structured and unstructured data. The structured data includes the template of a patent while the unstructured data includes the text. In our study we have analyzed the texts of the patents.

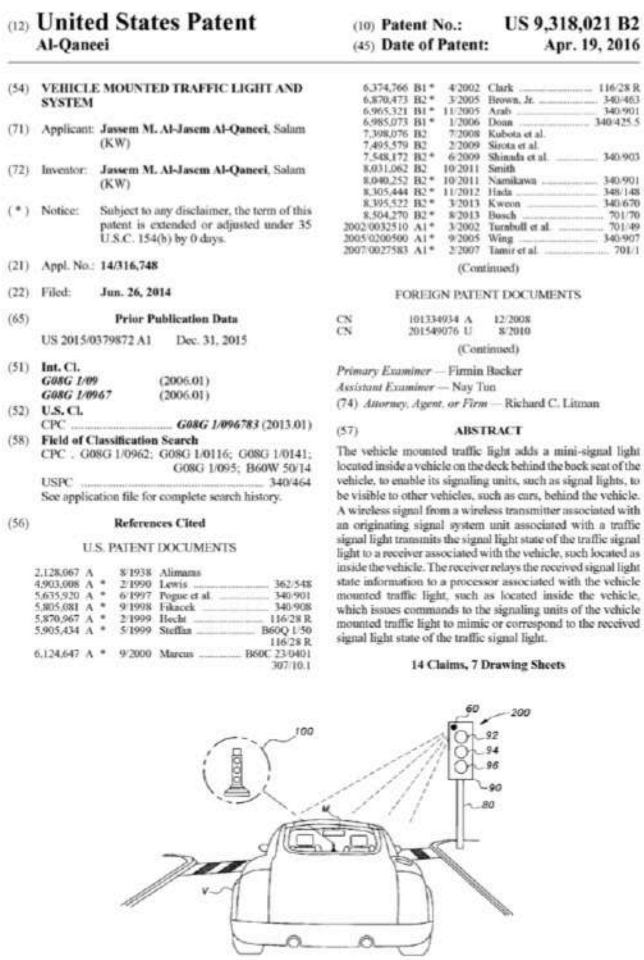
From the formal point of view, we may highlight that the patents consist of three main parts: the abstract in the front page, the description of the invention, and the claims. The roles of these parts are:

- The abstract at the first page of the patent briefly explains the core of the invention and the elements of form;
- The description of the invention outlines the technical field of the invention. It also contains a detailed description of the invention, including its form and function; it also provides an information about concept of operations of the system;

- The claims describe the scope of the invention and the technical features of it. The claims identify the features of the invention that are distinct from all previous inventions. The claims section is the part of the patent that has a legal importance.

Patents can be issued for systems (machines), methods (processes, acts), and composition of matter (chemical compounds, chemical compositions).

The example of the patent's front page is demonstrated in Figure 4.2.



(12) **United States Patent**
Al-Qaneei

(10) **Patent No.: US 9,318,021 B2**
(45) **Date of Patent: Apr. 19, 2016**

(54) **VEHICLE MOUNTED TRAFFIC LIGHT AND SYSTEM**

(71) Applicant: **Jassem M. Al-Jasem Al-Qaneei**, Salam (KW)

(72) Inventor: **Jassem M. Al-Jasem Al-Qaneei**, Salam (KW)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **14/316,748**

(22) Filed: **Jun. 26, 2014**

(65) **Prior Publication Data**

US 2015/0379872 A1 Dec. 31, 2015

(51) **Int. Cl.**
G08G 1/09 (2006.01)
G08G 1/0967 (2006.01)

(52) **U.S. Cl.**
CPC ***G08G 1/096783*** (2013.01)

(58) **Field of Classification Search**
CPC . G08G 1/0962; G08G 1/0116; G08G 1/0141; G08G 1/095; B60W 50/14
USPC 340/464
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

2,128,067	A	8/1938	Alimanas	
4,903,008	A *	2/1990	Lewis	362/548
5,635,920	A *	6/1997	Pogue et al.	340/901
5,805,081	A *	9/1998	Fikacek	340/908
5,870,967	A *	2/1999	Hecht	116/28 R
5,905,434	A *	5/1999	Steffan	B60Q 1/50 116/28 R
6,124,647	A *	9/2000	Marcus	B60C 23/0401 307/10.1
6,374,766	B1 *	4/2002	Clark	116/28 R
6,870,473	B2 *	3/2005	Brown, Jr.	340/463
6,965,321	B1 *	11/2005	Arab	340/901
6,985,073	B1 *	1/2006	Doan	340/425.5
7,398,076	B2	7/2008	Kubota et al.	
7,495,579	B2	2/2009	Sirota et al.	
7,548,172	B2 *	6/2009	Shinada et al.	340/903
8,031,062	B2	10/2011	Smith	
8,040,252	B2 *	10/2011	Namikawa	340/901
8,305,444	B2 *	11/2012	Hada	348/148
8,395,522	B2 *	3/2013	Kweon	340/670
8,504,270	B2 *	8/2013	Busch	701/70
2002/0032510	A1 *	3/2002	Turnbull et al.	701/49
2005/0200500	A1 *	9/2005	Wing	340/907
2007/0027583	A1 *	2/2007	Tamir et al.	701/1

(Continued)

FOREIGN PATENT DOCUMENTS

CN	101334934 A	12/2008
CN	201549076 U	8/2010

(Continued)

Primary Examiner — Firmin Backer
Assistant Examiner — Nay Tun
(74) *Attorney, Agent, or Firm* — Richard C. Litman

(57) **ABSTRACT**

The vehicle mounted traffic light adds a mini-signal light located inside a vehicle on the deck behind the back seat of the vehicle, to enable its signaling units, such as signal lights, to be visible to other vehicles, such as cars, behind the vehicle. A wireless signal from a wireless transmitter associated with an originating signal system unit associated with a traffic signal light transmits the signal light state of the traffic signal light to a receiver associated with the vehicle, such located as inside the vehicle. The receiver relays the received signal light state information to a processor associated with the vehicle mounted traffic light, such as located inside the vehicle, which issues commands to the signaling units of the vehicle mounted traffic light to mimic or correspond to the received signal light state of the traffic signal light.

14 Claims, 7 Drawing Sheets

Figure 4.2: Example of patent's front page (US Patent 9,318,021)

4.3.2 Urban Architectural Patterns

In his book that became widely recognized, Christopher Alexander had explored the pattern language of the architecture (Alexander 1977). In this book he studied patterns in towns, buildings, and construction, and developed a pattern language. He mentioned that "Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing in the same way twice". The merit of Alexander's work is that he observed existing architecture looking for patterns in it and formulating the recommendations how to build your own architecture.

In his book, each pattern has the same format provided below.

NAME OF THE PATTERN

PICTURE, which shows an archetypal example of that pattern

INTRODUCTORY PARAGRAPH, which sets the context for the pattern

***

ESSENCE of the problem (1-2 SENTENCES)

BODY of the problem (longest section)

SOLUTION (instruction)

DIAGRAM (solution in diagram)

***

PARAGRAPH, which ties the pattern to all subsequent patterns in the language

Similarly to the previous section, during the small-N study of urban architectural patterns we have focused on answering the following question: “Is this entry of the concept framework present in this urban architectural patterns?” The criteria based on which we included or not included the data from urban architectural patterns into the concept framework is presented in Table 4.2.

The example of urban architectural pattern is demonstrated in Figure 4.3.

. . . within an urban area, the density of building fluctuates. It will, in general, be rather higher toward the center and lower toward the edges—CITY COUNTRY FINGERS (3), LACE OF COUNTRY STREETS (5), MAGIC OF THE CITY (10). However, throughout the city, even at its densest points, there are strong human reasons to subject all buildings to height restrictions.

✣ ✣ ✣

There is abundant evidence to show that high buildings make people crazy.

High buildings have no genuine advantages, except in speculative gains for banks and land owners. They are not cheaper, they do not help create open space, they destroy the townscape, they destroy social life, they promote crime, they make life difficult for children, they are expensive to maintain, they wreck the open spaces near them, and they damage light and air and view. But quite apart from all of this, which shows that they aren't very sensible, empirical evidence shows that they can actually damage people's minds and feelings.

"The Ministry of Truth—Minitrue, in Newspeak—was startlingly different from any other object in sight. It was an enormous pyramidal structure of glittering white concrete, soaring up terrace after terrace 300 metres in the air." (*George Orwell*, 1984)

Figure 4.3: Example of urban architectural pattern. Excerpt from №21 “Four-Story Limit” pattern (Alexander 1977)

4.3.3 Software Patterns

In the book entitled "Design Patterns: Elements of Reusable Object-Oriented Software" the 23 patterns are discussed (Gamma et al. 1995). These patterns represent solutions to specific problems in object-oriented software design. Each one of the patterns is related to one of the three different types, namely, the creational patterns (5 patterns: Abstract Factory, Builder, Factory Method, Prototype, and Singleton), the structural patterns (7 patterns: Adapter, Bridge, Composite, Decorator, Façade, Flyweight, and Proxy), and the behavioral patterns (11 patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, and Visitor).

The role of each type of the patterns is as follows. The creational patterns are related to the process of object creation. The structural patterns deal with the composition of classes or objects. The behavioral patterns "characterize the ways in which classes or objects interact and distribute responsibility" (Gamma et al. 1995).

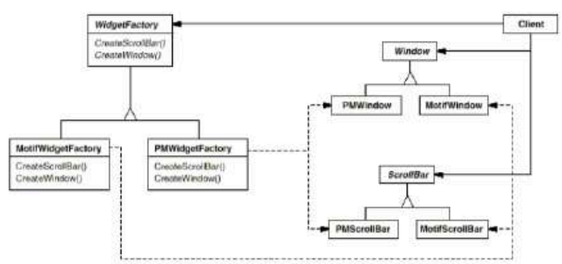
The general structure of software patterns is as follows:

Form	Function
PATTERN NAME AND CLASSIFICATION	Essence of the pattern
INTENT	Short statement that answers the following questions: "What does the design pattern do? What is its rationale and intent? What particular design issue or problem does it address?"
ALSO KNOWN AS	Other well-known names for the pattern, if any
MOTIVATION	Scenario that illustrates a design problem and how the class and object structures in the pattern solve the problem
APPLICABILITY	When the design pattern can be applied?
STRUCTURE	Graphical representation of the classes in the pattern
PARTICIPANTS	Classes/objects participating in the design pattern and their responsibilities

COLLABORATIONS	How the participants collaborate to carry out their responsibilities
CONSEQUENCES	What are the results of using the pattern?
IMPLEMENTATION	What hints or techniques should you be aware of when implementing the pattern?
SAMPLE CODE	Code fragments that illustrate how you might implement the pattern
KNOWN USES	Examples of the pattern found in real systems
RELATED PATTERNS	What design patterns are closely related to this one?

We have conducted small-N study of software patterns in order to identify which entries of the concept framework are used in software patterns.

The example of software pattern is demonstrated in Figure 4.4.



There is a concrete subclass of WidgetFactory for each look-and-feel standard. Each subclass implements the operations to create the appropriate widget for the look and feel. For example, the CreateScrollBar operation on the MotifWidgetFactory instantiates and returns a Motif scroll bar, while the corresponding operation on the PMWidgetFactory returns a scroll bar for Presentation Manager. Clients create widgets solely through the WidgetFactory interface and have no knowledge of the classes that implement widgets for a particular look and feel. In other words, clients only have to commit to an interface defined by an abstract class, not a particular concrete class.

A WidgetFactory also enforces dependencies between the concrete widget classes. A Motif scroll bar should be used with a Motif button and a Motif text editor, and that constraint is enforced automatically as a consequence of using a MotifWidgetFactory.

Applicability

Use the Abstract Factory pattern when

- a system should be independent of how its products are created, composed, and represented.
- a system should be configured with one of multiple families of products.
- a family of related product objects is designed to be used together, and you need to enforce this constraint.
- you want to provide a class library of products, and you want to reveal just their interfaces, not their implementations.

Figure 4.4: Example of software pattern. Excerpt from “Abstract Factory” pattern

(Gamma et al. 1994)

4.4 Methodology of Mapping Analytical Surveys to Framework for Small-N Study

This section describes the methodology of mapping patents, urban architectural patterns, and software patterns to the proposed *system concept representation framework* for small-N study. The first step is to construct the set of samples explained in sub-section 4.4.1 for each analytical survey. After this we explain the mapping process of analytical surveys to the *system concept representation framework* in sub-section 4.4.2.

4.4.1 Sampling Analytical Survey for Small-N Study

For the purpose of selecting patents, we were guided by sampling techniques for "small N" qualitative studies (Trost 1986). By "small N" we imply the number of samples that is relatively small comparing to a whole population of samples (in this case, the entire amount of patents). We listed the "independent" variables appropriate for the purposes of our study. These "independent" variables focused on the types of systems and methods that are represented in patents. In order to test for broad applicability, we chose four quite different types of patents: biological, thermodynamic, electro-mechanical, and software. In addition, we wanted the set to include methods patents, systems patents, and some that were both method and system patents.

This yielded eight patents in the four types:

1. Vehicle mounted traffic light and system (Al-Qaneei 2016)
2. Traffic signal device for driver/pedestrian/cyclist advisory message screen at signalized intersections (Holzmac Llc 2015)

3. System and method for launching a browser in a safe mode (Chebakov and Maslov 2016)
4. System and methods for detection of fraudulent online transactions (Golovanov and Monastyrsky 2016)
5. Method to generate novel bioactive molecules (Palsson and Charusanti 2015)
6. Methods and compositions for enhanced delivery of bioactive molecules (Lewis et al. 2004)
7. Heat exchanger arrangement for turbine engine (Murphy and Zamora 2015)
8. Heat engine and heat to electricity systems and methods (Held et al. 2012)

For the urban architectural patterns by "independent" variables, we applied the classification proposed by Alexander: the patterns are related to the towns, buildings, and construction. As a result, we randomly selected the following nine urban architectural patterns for the small-N analysis:

1. Pattern №11: Local transport areas
2. Pattern №21: Four-story limit
3. Pattern №22: Nine per cent parking
4. Pattern №110: Main entrance
5. Pattern №124: Activity pockets
6. Pattern №127: Intimacy gradient
7. Pattern №221: Natural doors and windows
8. Pattern №241: Seat spots
9. Pattern №242: Front door bench

Similarly to the urban architectural patterns, there are three types of software patterns: creational patterns, structural patterns, and behavioral patterns. We randomly selected the following three software patterns for the small-N analysis:

1. Abstract Factory
2. Adapter
3. Strategy

4.4.2 Mapping Analytical Surveys to Framework for Small-N Study

In order to introduce a rigorous process for mapping patents, urban architectural patterns, or software patterns to proposed *system concept representation framework*, we assigned a number to each one of the 28 entries of framework that is presented in Table 3.1. Entries 1 and 2 are related to stakeholders and their needs (proposition I in Table 3.1); 3-7 deal with the solution-neutral environment (proposition II in Table 3.1); 8-16 are related to the solution-specific environment or concept (proposition III in Table 3.1); entries 17-25 are concerned with integrated concept, including structure and interactions (proposition IV in Table 3.1); and entries 26-28 deal with the concept of operations (proposition V in Table 3.1), respectively. Note that if only the information suggested in Figure 3.7 (the solution-neutral problem and core solution-specific concept) were represented, entries would only be found in the patent, urban architectural pattern, or software pattern under entries 3-16.

The text of each of the eight patents, nine urban architectural patterns, and three software patterns was analyzed, and the outcome summarized as the answer to the following question: “Is this specific entry used to map the patent, urban architectural pattern, or software pattern to the framework?” If the answer is yes then we have put “V” against the corresponding entry and, consequently, we counted this entry as present at the patent, urban architectural pattern, or software pattern. This allows us to see how the information on the 28 entries is spread throughout the study. Table 4.2 provides the criteria applied during the analysis in order to define whether the specific information in the text corresponds to the term of concept framework entry.

4.5 Results of Small-N Study

4.5.1 Mapping Patents to Framework

The results of mapping for small-N sample of patents (Menshenin and Crawley 2020) is shown in Figure 4.5, which summarizes the outcomes for the question “is this specific entry used to map the patent to the framework?” The criteria based on which the text was analyzed and the specific entry was counted or not counted for inclusion into the framework could be found in Table 4.2. The axis X of the plot of Figure 4.5 contains all 28 entries of the framework, while the axis Y represents the frequency of references in percentage. For example, if one sees 100% frequency of references against a specific entry, it means that this entry is present in all patents that were analyzed during the study.

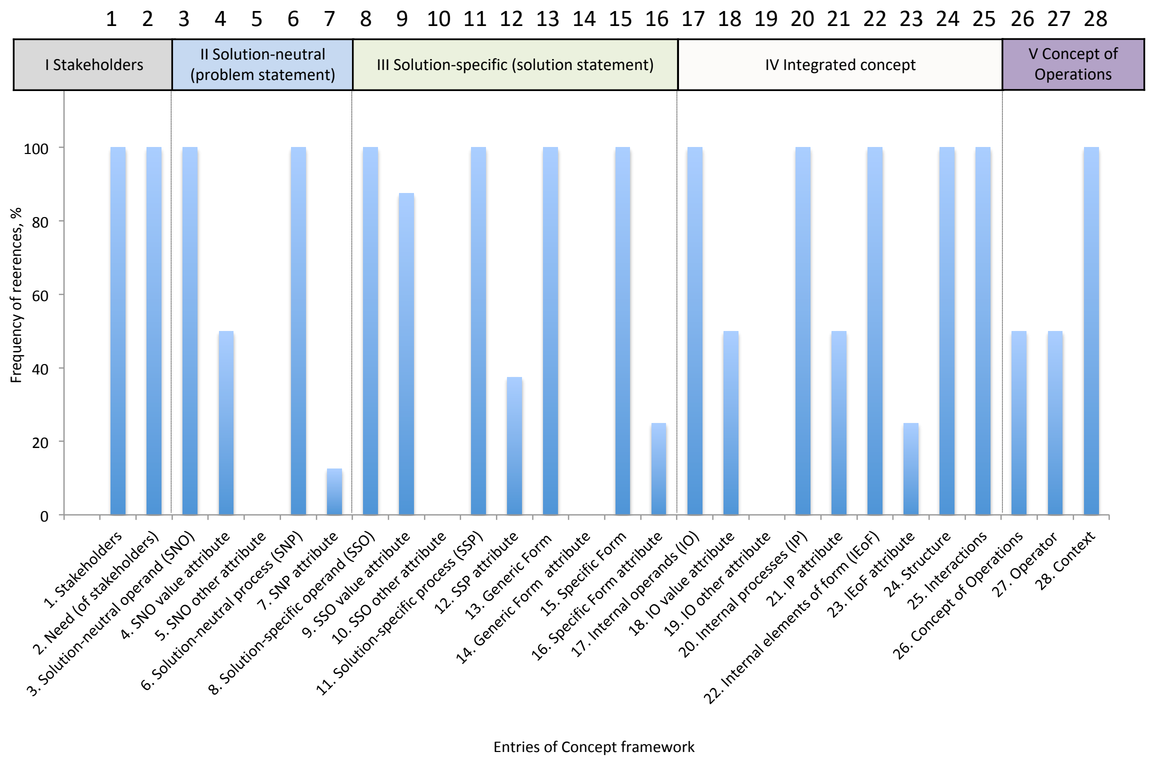


Figure 4.5: Occurrence of concept framework's entries in patents (small-N study)

The first important conclusion is that the broad concept framework is needed for mapping a patent, and not just the information at the core of the concept (entries 3-16 indicated in Figure 3.7). In particular, the stakeholder information (entries 1-2) is vital, as is the integrated concept (entries 17-25), including the relationship information (structure, interactions), and concept of operations – entries 26-28. Using the more restrictive definition of concept in Figure 3.7 would not be sufficient, as it only partly describes the concept. As it can be seen in Figure 4.5, such entries of framework as solution-neutral operand and process (entries 3 and 6), solution-specific operand and process (entries 8 and 11), generic form (entry 13), and specific form (entry 15) are always present in patents.

An especially interesting result is that the claims, the intrinsic parts of any patent, are primarily reflected in the first level decomposition of the integrated concept (17-25). Every patent maps claims to the internal operands, processes and elements. There is clearly an underlying force at work here. For a patent to describe the invention in a legally defensible way, the patent must decompose the invention into a number of pieces and explain what each one of the pieces does. Each one of such pieces composed of internal element that is used to execute the internal function (internal process plus internal operand in the notation of the proposed framework). It is also an opportunity to engage the model-based conceptual design to represent the integrated concept.

Another observation is that the "other attributes" are never used in the mapping (entries 5, 10, 19 indicated in Table 3.1), while "attributes" are seldom used to characterize the corresponding operand, process, or form (entries 7, 12, 14, 16, 21, 23 indicated in Table 3.1) – the average appearance is 25%. The value-related attributes are appearing more often – in 62.5% cases (entries 4, 9, 18 indicated in Table 3.1). There is some sense to this. Since the person applying for a patent wants the coverage to be as broad as possible, it is in their interest to not restrict or constrain it by qualifications implied by these entries. It is likely that for patent analysis the nine entries related to the "other attributes" and "attributes" could be omitted without loss of important information.

4.5.2 Mapping Urban Architectural Patterns to Framework

The small-N analysis for the urban architectural patterns is presented in Figure 4.6, during which the same question was asked - "is this specific entry used to map the urban architectural pattern to the framework?" The 28 entries of the concept framework are contained in the axis X of the plot of Figure 4.6, while the axis Y represents the frequency of references in percentage. The criteria based on which the decision on inclusion of specific information from the text of urban architectural patterns was made could be found in Table 4.2.

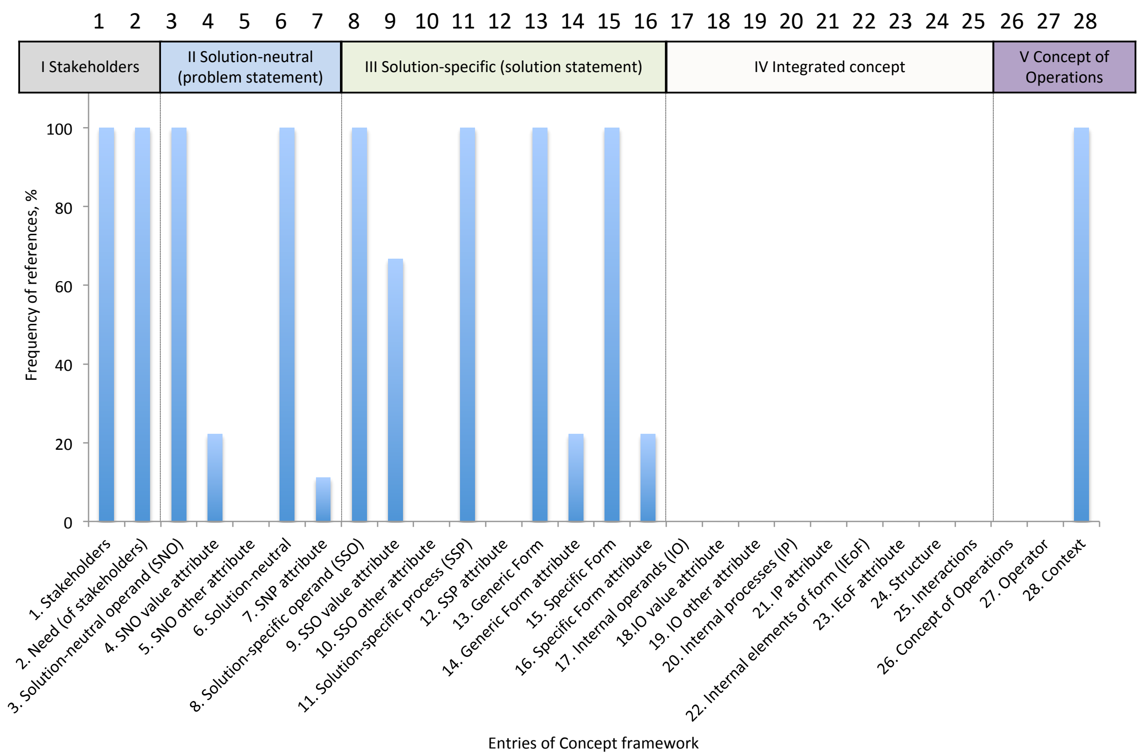


Figure 4.6: Occurrence of concept framework's entries in urban architectural patterns (small-N study)

From Figure 4.6 we may notice that the urban architectural patterns are mainly focusing on the first propositions of the proposed framework: the stakeholders (entries 1-2), the solution-neutral problem statement (entries 3-7), and the solution-specific solution statement (entries 8-16). In particular, such entries as stakeholders and stakeholders' need (entries 1-2), solution-neutral's operand and process (entries 3 and 6), solution-specific's operand and process (entries 8 and 11), as well as generic form and specific form (entries 13 and 15) are always present in urban architectural patterns. Additionally, in Alexander's work we always found the context (entry 28).

A pertinent observation is that comparing with the outcomes of patents analysis, the core difference of urban architectural patterns analysis is that there is no information about the integrated concept. Partly this can be explained by highly abstracted nature of Christopher Alexander's work: his goal was to identify the problem and propose the solution – both (problem and solution) are highly interwoven with societal structures. In urban architectural patterns there is no such part as claims, as we find in patents. Claims require very clear evidence, since they are legally significant. Another reason of the absence of the integrated concept in the texts of urban architectural patterns is that its elements sometimes appear in accompanying figures. However, the scope of our study was to only analyze the text. Of course, the patents also contain figures, and we consistently did not use them in analysis above.

Another outcome of urban architectural patterns analysis is that the context (entry 28) is always present in them. There is some explanation to this. Alexander's work is closely related to the issues appearing in human's daily life – either while constructing the personal houses, or during the process of searching for solution to improve the communities where people live. Thus, it is important to provide the context under which urban architectural pattern is considered.

4.5.3 Mapping Software Patterns to Framework

The small-N analysis for the software patterns is presented in Figure 4.7. Similarly to the previous studies, the same question was tested: "Is this specific entry used to map the software pattern to the framework?" The criteria and key definitions of the concept framework are summarized in Table 4.2.

The analysis of the small-N study results reveals that the first proposition of proposed framework (stakeholders) is always present in the software patterns; the second proposition (solution-neutral problem statement) is presented in its core entries – process and operand; the third proposition (solution-specific solution statement) is also present in its core entries – process, operand, and form. The fourth proposition (integrated concept), including structure and interactions, is always contained in the software patterns; and finally, the fifth proposition (concept of operations) can always be found in the software patterns.

Another important result is that the attributes related to the two propositions of the concept framework (solution-neutral environment, and solution-specific environment) regularly appear in the software patterns – in 54% of cases. The attributes related to the fourth proposition of the framework (integrated concept) appear much more seldom – in 17% of the cases. This result could be explained by the nature of software patterns: they tend to propose the solution in the solution-specific domain (third proposition of proposed framework). The information contained in the decomposed elements (integrated concept) aims to support the solution at a higher level. Thus, there is no need to define the attributes of each one of the subclasses – this should be done at later stages of the design process. These results are illustrated in Figure 4.7.

Comparing to results of previous studies we may conclude that every proposition of proposed concept framework is present in software patterns.

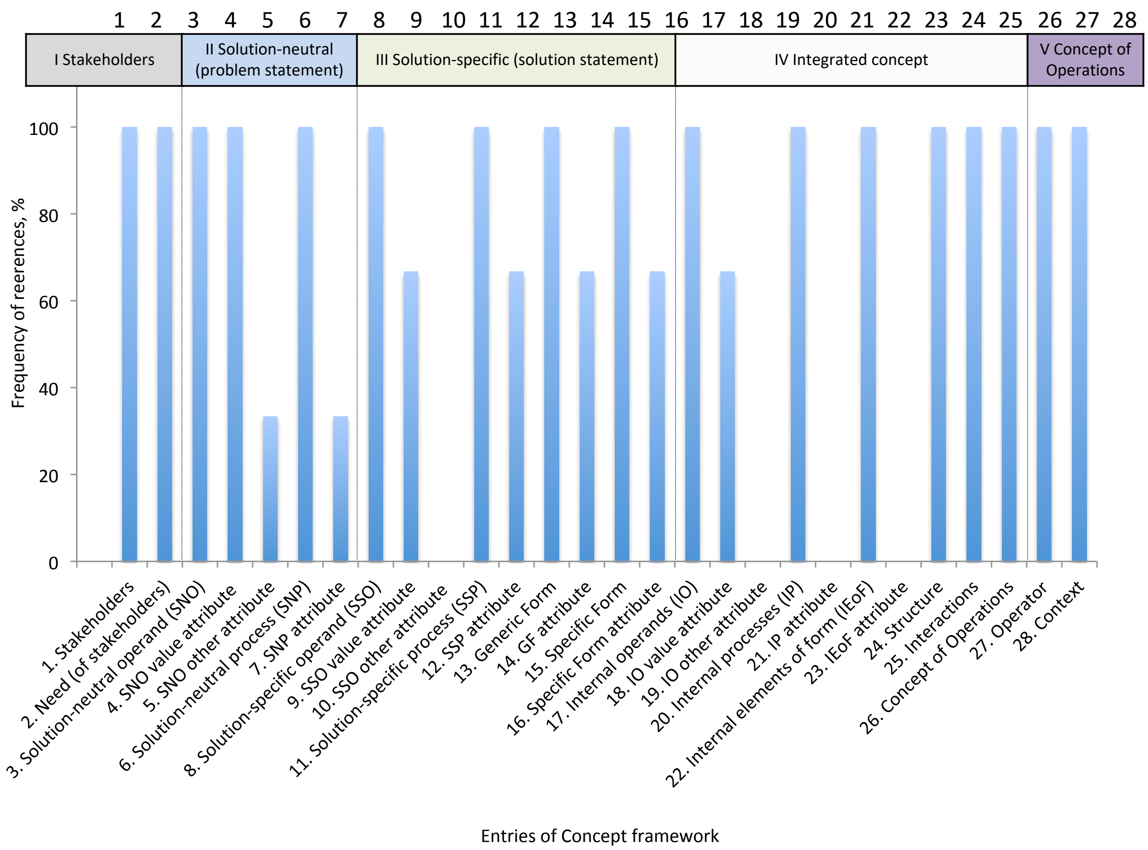


Figure 4.7: Occurrence of concept framework’s entries in software patterns (small-N study)

4.6 Methodology of Mapping Analytical Surveys to Framework for Large-N Study

This section describes the methodology of mapping patents, urban architectural patterns, and software patterns to the proposed *system concept representation framework* for large-N study. In general, this procedure is close to the one explained for small-N study in section 4.4. Firstly, we construct the set of samples for each analytical survey – it is explained in sub-section 4.6.1. After this we explain mapping process of analytical surveys to *system concept representation framework* in sub-section 4.6.2.

4.6.1 Sampling Analytical Survey for Large-N Study

The purpose of the large-N study is to validate the results of small-N analysis by exploring the larger set of patents, urban architectural patterns, and software patterns.

In order to choose the relevant and specific patents, we were guided by the recommendations that were given to us by the Intellectual Property Center of the Skolkovo Foundation. This Center is specialized in the patents' analysis, preparation, and filing. The first step is to target some group of patents, which is identified by keywords "suborbital spaceflight vehicle". These keywords were chosen for two reasons. First of all, this topic is correlated with one of the case studies to be deeply explored in Chapter 5 – suborbital human spaceflight missions. Secondly, this topic enriches the electromechanical set of patents explored during the small-N study. In total, 85 patents and patent applications with corresponding keywords were found using Google Patents resource. These patents and patent applications are summarized in Appendix A. Out of these 85 patents 25 the most cited ones were chosen for further analysis. The analysis was following the procedure explained in previous sub-sections: each patent was mapped against the list of criteria from Table 4.2 (see section 4.2) to identify how many system concept representation framework's entries appear in the patent. The list of analyzed patents is provided below:

1. Aerospace vehicle having multiple propulsion systems on a relatively rotatable flying wing (Criswell 1989)
2. Space vehicle apparatus including a cellular sandwich with phase change material (Hickey 1997)
3. Hypersonic and orbital vehicles system (Redding 2003)
4. Hypersonic and orbital vehicles system (Redding 2001)
5. Variable-altitude testing systems (MacCallum and Anderson 2008)
6. In-line staged horizontal takeoff and landing space plane (Luther 2013)
7. Flyback booster with removable rocket propulsion module (Aldrin and Davis 2003)
8. Movable ground based recovery system for reusable space flight hardware (Sarver 2013)
9. Failure resistant multiline tether (Hoyt and Forward 2001)
10. Failure resistant multiline tether (Hoyt and Forward 2002)
11. Failure resistant multiline tether (Hoyt and Forward 2001)
12. Failure resistant multiline tether (Hoyt and Forward 2002)

13. Multistage launch vehicle employing interstage propellant transfer and redundant staging (Leonard 1992)
14. Planar hoytether failure resistant multiline tether (Hoyt and Forward 2001)
15. Electromagnetic transportation system for manned space travel (Minovitch 1989)
16. System for the delivery and orbital maintenance of micro satellites and small space-based instruments (Lopata and Kamel 2005)
17. Electrically powered spacecraft/airship (Provitola 2002)
18. Airborne space simulator with zero gravity effects (Lewis and Mascia 2012)
19. Reusable flyback satellite (Cervisi et al. 1995)
20. Method of earth orbit space transportation and return (Toliver 2000)
21. Electrodynamic tether control (Hoyt and Forward 2001)
22. Balloon device for lowering space object orbits (Nock et al. 2004)
23. Composite structures for aerospace vehicles, and associated systems and methods (Grillos 2015)
24. Techniques for optimizing an autonomous star tracker (van Bezooijen 1998)
25. Non-propellant fluid cooled spacecraft rocket engine (Dressler 2000)

For large-N study we have chosen 27 urban architectural patterns. We have randomly selected nine patterns per each type of the architecture. We have chosen the only those patterns that were marked by two-asterisks.

The list of chosen urban architectural patterns is summarized below.

Towns:

1. Pattern №1: Independent regions

2. Pattern №3: City country fingers

3. Pattern №9: Scattered work

4. Pattern №30: Activity nodes

5. Pattern №31: Promenade

6. Pattern №37: House cluster

7. Pattern №41: Work community

8. Pattern №46: Market of many shops

9. Pattern №67: Common land

Buildings:

10. Pattern №112: Entrance transition

11. Pattern №117: Sheltering roof

12. Pattern №140: Private terrace on the street

13. Pattern №148: Small work groups

14. Pattern №155: Old age cottage

15. Pattern №159: Light on two sides of every room

16. Pattern №161: Sunny place

17. Pattern №167: Six-foot balcony

18. Pattern №171: Tree places

Construction:

19. Pattern №205: Structure follows social spaces

20. Pattern №212: Columns at the corners

21. Pattern №219: Floor-ceiling vaults

22. Pattern №225: Frames as thickened edges

23. Pattern №227: Column connections

24. Pattern №233: Floor surface

25. Pattern №247: Paving with cracks between the stones

26. Pattern №249: Ornament

27. Pattern №250: Warm colors

For large-N study of software patterns we've randomly selected 12 patterns, identifying 4 samples per each type (creational, structural, and behavioral patterns). The list of patterns is provided below.

Creational patterns:

1. Builder

2. Prototype

3. Singleton

4. Factory Method

Structural patterns:

5. Bridge

6. Decorator

7. Façade

8. Composite

Behavioral patterns:

9. Command

10. Iterator

11. State

12. Template Method

4.6.2 Mapping Analytical Surveys to Framework for Large-N Study

We followed the same strategy as the one introduced in sub-section 4.4.2. We assigned a number to each one of the 28 entries of the framework that is presented in Table 3.1. Entries 1 and 2 are related to stakeholders and their needs (proposition I in Table 3.1); 3-7 deal with the solution-neutral environment (proposition II in Table 3.1); 8-16 are related to the solution-specific environment or concept (proposition III in Table 3.1); entries 17-25 are concerned with integrated concept, including structure and interactions (proposition IV in Table 3.1); and entries 26-28 deal with the concept of operations (proposition V in Table 3.1), respectively.

The text of each of the 25 patents, 27 urban architectural patterns, and 12 software patterns was analyzed, and the outcome summarized as the answer to the following question: "Is this specific entry used to map the patent, urban architectural pattern, or software pattern to the framework?" If the answer is yes then we have put "V" against the corresponding entry and, consequently, we counted this entry as present at patent, urban architectural pattern, or software pattern. This allows us to see how the information on the 28 entries is spread throughout the study. Table 4.2 provides the criteria applied during the analysis in order to define whether the specific information in the text corresponds to the term of concept framework entry.

4.7 Results of Large-N Study

4.7.1 Mapping Patents to Framework

The results of mapping for large-N sample of patents is shown in Figure 4.8. This Figure informs us about frequency of references in percentage: 100% means that a specific entry is present in all patents that were analyzed during the large-N study.

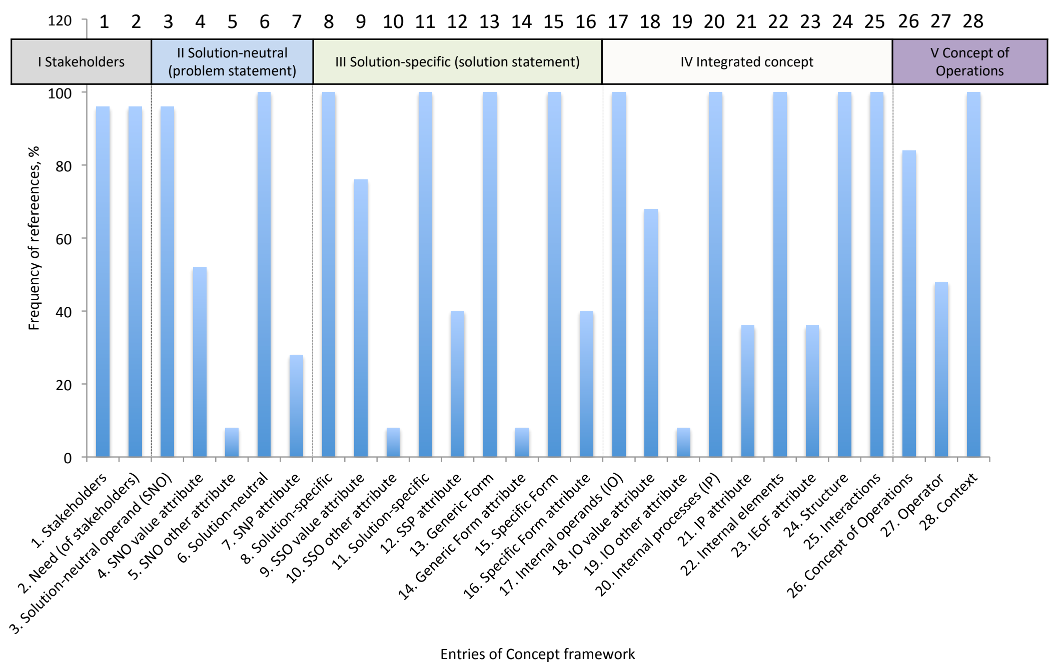


Figure 4.8: Occurrence of concept framework's entries in patents (large-N study)

The main outcome of Figure 4.8 is that the results of large-N study correlate with results of small-N study for patents. In particular, Figure 4.8 demonstrates that the stakeholders information (entries 1-2) is important, as well as the integrated concept (entries 17-25) and the concept of operations (entries 26-28). Also, the results of large-N study for patents confirm that the attributes do not play a major role for patents.

4.7.2 Mapping Urban Architectural Patterns to Framework

Figure 4.9 illustrates the results of mapping of large-N study of urban architectural patterns to proposed framework. The results correspond to the results of small-N study for urban architectural patterns: an emphasis is made on first propositions of framework – stakeholders (entries 1-2), the solution-neutral environment (entries 3-7), and the solution-specific environment (entries 8-16).

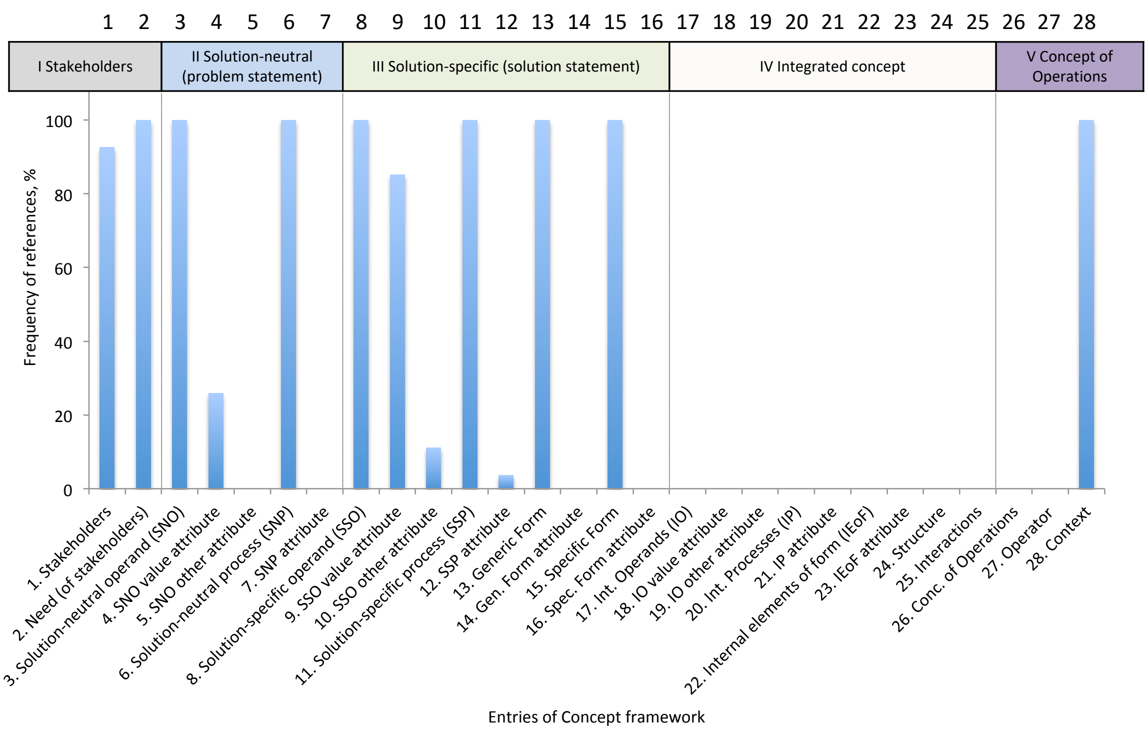


Figure 4.9: Occurrence of concept framework's entries in urban architectural patterns (large-N study)

4.7.3 Mapping Software Patterns to Framework

Figure 4.10 contains the results of the software patterns analysis for large-N study. This analysis correlate with the small-N study for software patterns. Notice that each proposition of the proposed framework is present in Figure 4.10, while the emphasis is made on the core entries – operands and processes. Software patterns also have the information about concept of operations, usually explaining how code is implemented.

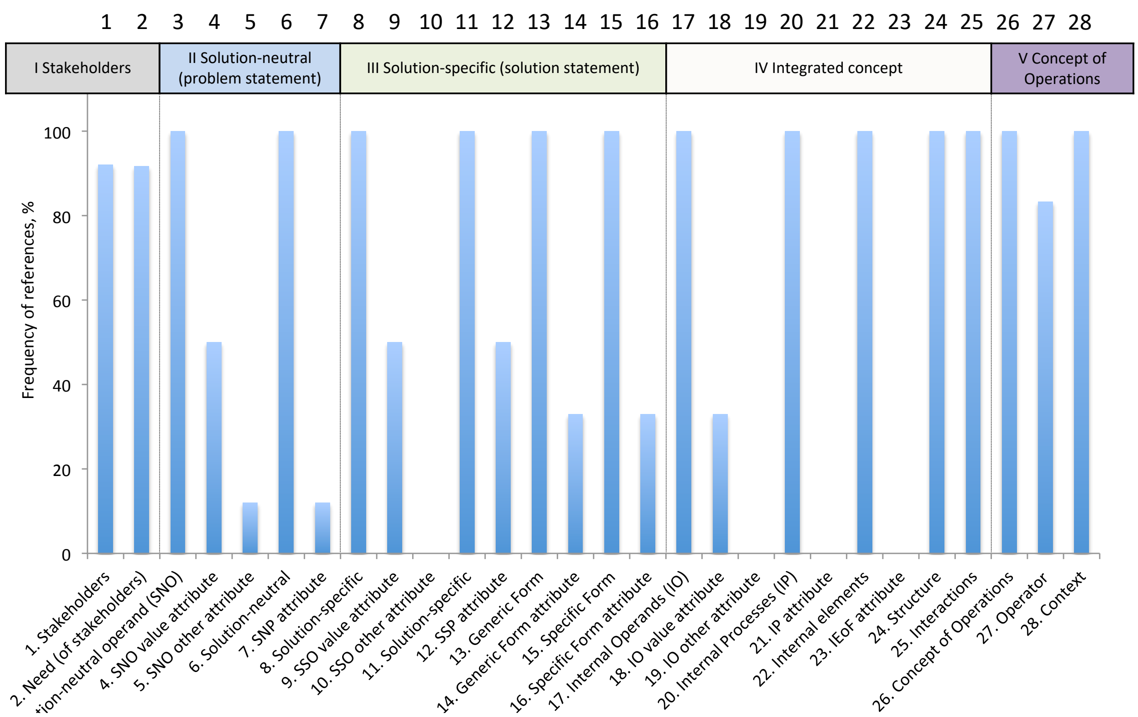
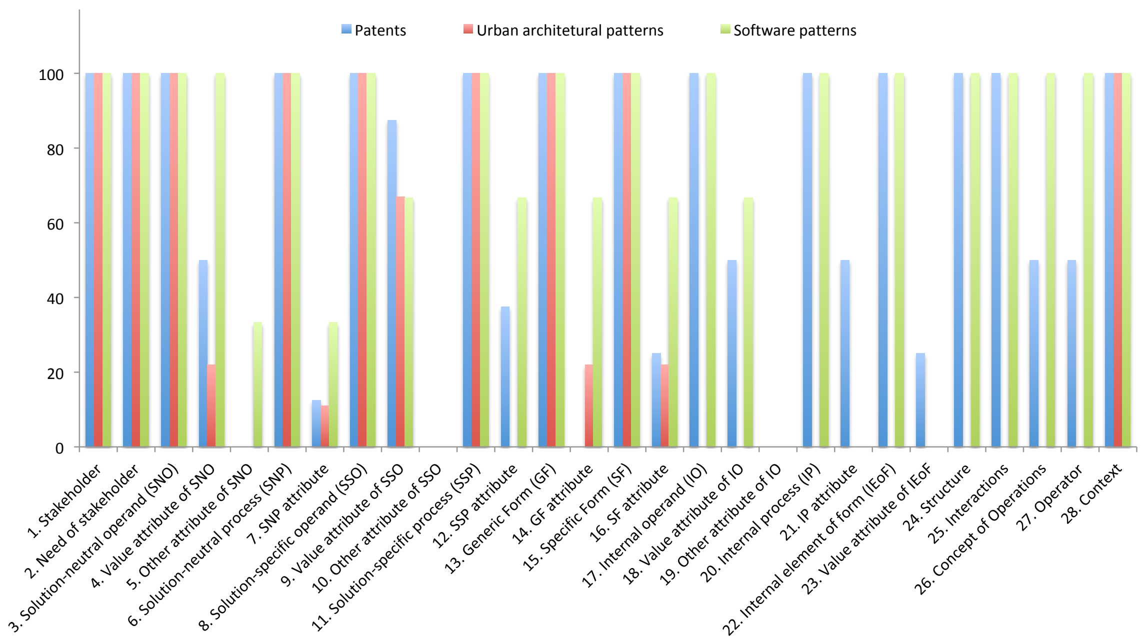


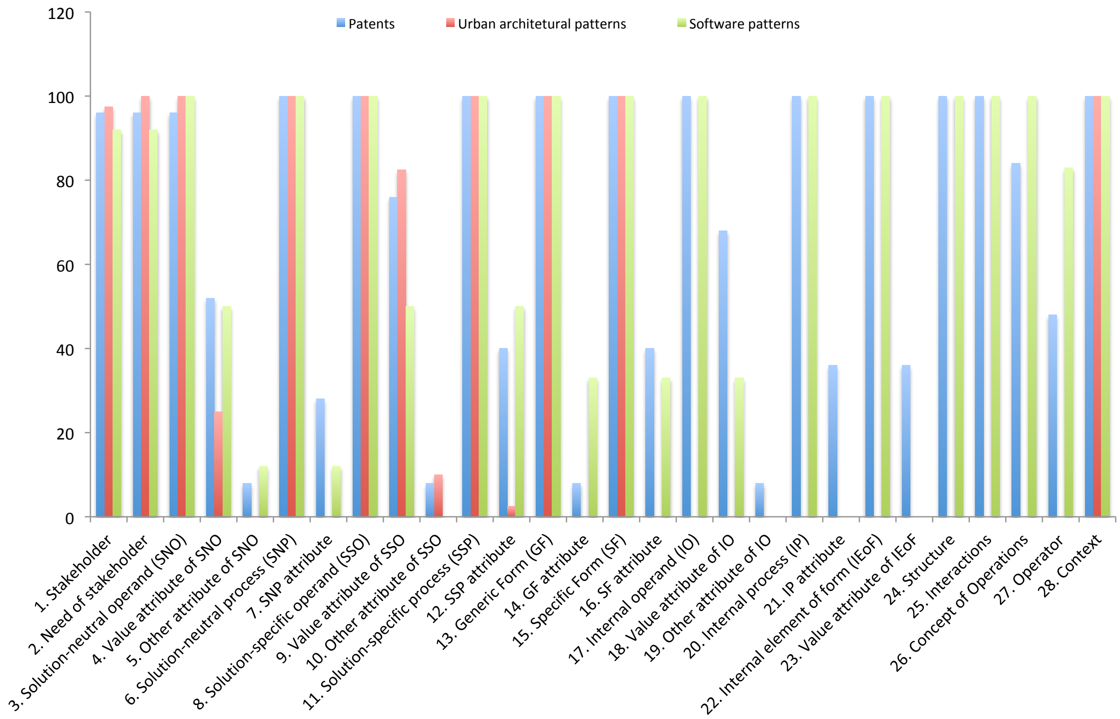
Figure 4.10: Occurrence of concept framework's entries in software patterns (large-N study)

4.8 Cross-Cutting Results for System Concept Representation Framework

The small-N analysis and large-N analysis of patents, urban architectural patterns, and software patterns revealed important outcome – that they can all be successfully mapped to the framework. The entries in the framework largely describe the concept represented in patent, urban architectural pattern, or software pattern (see Figure 4.11(a) for small-N analysis and Figure 4.11(b) for large-N analysis). Therefore, mapping the information from these sources of knowledge to the concept framework *encodes* valuable information about their concepts.



(a)



(b)

Figure 4.11: Occurrence of concept framework entries in patents, urban architectural patterns, and software patterns for (a) small-N analysis, and (b) large-N analysis

We have demonstrated that the conceptual content of a patent, urban architectural pattern, or software pattern can be mapped to the proposed framework. Virtually all of the entities of the framework are used in some or another patent, urban pattern or software pattern. But not all entries are used universally, and there are profiles of the entries used by patents, by urban architectural patterns and by software patters, as discussed in previous sections. This result validates the design of our small-N and large-N experiments. Our choice of diverse examples to map to the framework provided this breadth of coverage.

Examining Figure 4.11 in more detail, we see that some entries are universal:

- The stakeholders and their needs (entries 1, 2) of the stakeholders proposition
- The solution-neutral operand and process (entries 3, 6) of the solution-neutral proposition
- The solution-specific operand, process, generic form and specific form (entries 8, 11, 13, 15) of the solution-specific proposition
- The context (entry 28) of the concept of operation proposition.

If we ignore the absence of urban architectural patterns, which convey much of the following information by figure, the other universal entries are the internal operand, internal process and internal elements of form, and the structure and interactions (entries 17, 20, 22, 24, 25) of the integrated concept proposition. This is surprising, as a conventional definition of concept may not contain this level of detail. But we have seen it is key in patents and software patterns.

Working from the low side, we identify two attributes that are almost never used: the other attributes of the solution-specific operand and internal operands (entries 10, 19). This is probably due to the fact that the value-related attributes of the operands are more important, and the "others" go unnoticed.

In between important and rare are 12 entries that are sometimes used and sometimes not. Ten of these are various attributes – not critical but useful. Two of them are aspects of operations: concept of operations and operator (entries 26, 27). Due to the importance of operations in delivering value, one would think these would be more universal.

One sense of the potential completeness can be gained by comparison with representation of human thought in linguistics. Noam Chomsky discovered three intrinsic parts of human natural language (Chomsky 1956): a noun that is the instrument of the action, a verb describing the action, and a noun that is the object of the action. In order to describe some idea or concept, the human uses nouns and verbs to form a meaningful sentence. The framework proposed in our work is useful, as it provides a means to *encode* the system concept – expressed as object nouns (the operand), verbs (the processes), instrument nouns (the form), along with adverbs, and adjectives (the last two are represented by the "attributes" in the framework). For example, by taking the proposed system models of the specific patent/urban architectural pattern/software pattern the systems engineer could reconstruct the claims part of the patent, or to identify the concrete solution for the societal problem explained in an urban pattern, or the appropriate software pattern allowing to solve a specific problem in software implementation. The framework is a useful tool in the representation of knowledge contained in analyzed patents and patterns.

4.9 Summary and Conclusion

The objective of this Chapter was to validate a *system concept representation framework* through the set of analytical surveys. This was the observation of authors that the proposed concept framework comprises 28 entries, which span from stakeholders proposition, solution-neutral function and solution-specific concept to integrated concept and concept of operations.

To achieve this objective, two studies were conducted: small-N analysis at which eight patents, nine urban architectural patterns, and three software patterns representing a broad spectrum of engineering and urban systems were mapped to the concept framework; and large-N analysis at which 25 patents, 27 urban architectural patterns, and 12 software patterns were analyzed. All were successfully mapped to the framework, in the sense that the key entries were identified. This confirmed that the content of the patent, urban architectural pattern, or software pattern could be mapped to the proposed framework. This success is a necessary condition to showing utility of the framework.

Within the results there were some interesting details. Virtually all 28 entries were used by the patents, urban architectural patterns and software patterns, but each one did not use them all. There were some profiles to the usage.

For patents, an important specific finding is that the claims section of a patent, the key area of legal protection, contain information on form and function at one level of decomposition below the concept itself, in what we call the integrated concept.

The urban architectural patterns primarily addressed stakeholder needs, solution-neutral and specific domains and context, but left the integrated concept and concept of operations to figures, which were not analyzed.

The software patterns propose the way to satisfy the stakeholders needs focusing on the detailed explanation of stakeholders, solution-neutral/specific environments, and context; and only on the most essential entries of the integrated concept (operands, processes, and forms). But there was little information about attributes.

Looking across all of the mappings, we can identify a core of entries that appeared in all cases. These included stakeholders, solution-neutral and specific elements and context. More details on the integrated concept were identified than might be expected.

The proposed framework might have several forms of utility: *encoding* the core information about concepts such as contained in patents, urban architectural patterns, and software patterns; and *generating* concepts at early phases of the design process in a model-based environment, thus contributing to INCOSE Model-Based Conceptual Design initiative. In addition to these two forms of utility, the framework might be useful as it can help to measure the distance between concepts, allowing the formal analysis, such as identification of similarity between alternative concepts. Another utility of the framework is that the concept knowledge is *reused* in later stages of the design process – during the architecture development. Thus, the information from the conceptual design phase is spread throughout the other design stages.

In this work the information from patents, urban architectural patterns, and software patterns was mapped into the proposed framework by means of human reasoning supported by a clear criteria listed in Table 4.2. However, such mapping could be done with textual analysis by means of machine learning. Thus, one of the directions of future work would be exploring the knowledge description in a machine-accessible way, which would imply the usage of OWL ontology (Antoniou and van Harmelen 2004) and a concept classification scheme representation in a SysML language.

Chapter 5. Case Study I: Suborbital Human Spaceflight Mission

Image: Virgin Galactic

5.1 Introduction

The objective of Chapter 5 is to demonstrate the utility of proposed framework through its application to suborbital human spaceflight systems. These projects represent the paradigm of “New Space” commercial market. A huge interest to “New Space” paradigm is connected with the potential market opportunities that will appear as soon as the suborbital human spaceflight systems are available for customers on a regular base. Suborbital vehicles, for example, enable such markets as space tourism and point-to-point transportation.

Applying model-based *system concept representation framework* to suborbital human spaceflight systems we would demonstrate a practical utility of concepts' *encoding* and how this process supports a *formal analysis*, such as concept similarity assessment.

In Chapter 5 we focus on model-based representations of three conceptually different suborbital projects, namely, Virgin Galactic, Blue Origin, and XCOR. These concepts are *encoded* in a systematic way – starting from stakeholders and their needs and ending with the concept of operations of the systems. These concept models are connected with architectural decisions that are developed by systems engineer. The model-based concepts for suborbital systems demonstrated in Chapter 5 are built based on the methodology, explained in Chapter 3. The Chapter 5 corresponds to the Descriptive Study II, according to the DRM framework that is used in this Thesis.

In Chapter 5 we also propose a new method to document the information about specialization and decomposition relationships through the DSM approaches (Menshenin and Crawley 2018a). This method is utilized in current Chapter to demonstrate the *formal analysis*, such as conceptual similarity assessment.

The case study presented in Chapter 5 was developed in the System Architecture Lab at MIT in 2016. The study involved colleagues from Technical University of Munich and Massachusetts Institute of Technology.

The remainder of Chapter 5 is organized as follows. In Sect. 5.2, a historical background of "New Space" paradigm is provided. In Sect. 5.3 the motivation and context for model-based concept framework development for suborbital human spaceflight missions are discussed. In Sect. 5.4 the specific objectives for this case study are mentioned. Application of the model-based concept framework to three alternative suborbital human spaceflight projects is demonstrated in Sect. 5.5. The formal analysis applied for alternative concepts is discussed in Sect. 5.6. The interconnections between solution diagrams and architectural decisions are explained in Sect. 5.7 Finally, in Sect. 5.8 the conclusions are outlined.

5.2 Historical Background

Space exploration is a natural need of mankind: from a very beginning of our history we were interested in expansion of our boundaries of knowledge. This was not just a curiosity, rather a logical consequence of breakthrough in science and technology that have led to new, more and more complex, systems. These systems and capabilities, provided by them, were used for searching for new territories for trading, expansion of influence, and enrichment of culture.

The twentieth century has become a fruitful in exploration and development of systems aimed at expanding our knowledge about air and space. In 1903 Wilbur and Orville Wright have successfully completed the test of pushing off the ground. The Wright brothers have put their names in history as the inventors of the first powered aircraft (Hallenberg 2004).

At the same year Konstantin Tsiolkovsky published his manuscript “Exploration of the World Space with Reaction Machines” (Tsiolkovsky 1903). This work, as well as his follow-up articles became the world’s first scientifically viable proposals to explore outer space with rockets. In 1926 Robert Goddard successfully launched the first liquid-fueled rocket (NASA Facts). During the WW2 Wernher von Braun invented Fau-2, the first ballistic rocket. After the war, the scientific developments of Germans were “divided” by the United States and the Soviet Union. Wernher von Braun moved to USA to continue his work on space systems, while many technical documents were delivered to USSR.

On October 4, 1957 the Soviet Union has successfully launched the first artificial Earth satellite. This event marked the start of the space age and US-USSR space race. On April 12, 1961 the Soviet Union has made the next step, when Yuri Gagarin became the first human to journey into outer space. In a response to these achievements, a month after Gagarin's flight President John F. Kennedy announced before a special session of Congress an ambitious goal of sending an American to the Moon (Kennedy 1961). This mission was completed on July 20, 1969, when Neil Armstrong stepped down on the Moon surface. In July 1975 two superpowers conducted Apollo-Soyuz Test Flight in order to symbolize the policy of détente.

The Space race was mainly politically motivated time when countries dramatically increased the States budgets for space exploration (as it were for the Moon program) in order to demonstrate the technological superiority. This period was productive for space exploration. Among its legacy are the Earth communications and weather satellites, as well as human's presence at the International Space Station.

A new century marked the emergence and implementation of new philosophy in space sector. An exponential growth of IT industry and the spirit of serial entrepreneurs have led to understanding that many of the space products and services can be commercially viable. As a result, a "New Space" paradigm has appeared in many subsectors of space industry. In Chapter 5 we apply the framework developed in this Thesis to one of such subsectors, a suborbital human spaceflight market.

5.3 Motivation and Context

The suborbital market is considered as a new and promising one, attracting a huge interest from business, scientific, and government organizations. In May 1996 the "X PRIZE" contest, later renamed as "Ansari X PRIZE", was announced. This was a starting point for emergence and growth of significant number of new projects in suborbital human spaceflight market. Twenty five companies have joined this contest, which had a clearly stated purpose: to offer $10M prize for the first non-governmental organization, which will "build a reliable, reusable, privately financed, manned spaceship capable of carrying three people to 100 kilometers above the Earth's surface twice within two weeks" (X PRIZE). The prize was awarded in 2004 to the Tier One project designed by Burt Rutan, using the experimental spaceplane SpaceShipOne. In our days this concept underlies the mission of Virgin Galactic system, which is part of the case study.

Peeters emphasizes the potential evolution of commercial personal spaceflights (Peeters 2010). He highlights two big markets for suborbital spaceflights – space tourism and point-to-point (P2P) space travel. Space tourism market aims at attracting wealth customers willing to explore new conditions like space environment and ready to pay high-ticket price in amount of $200-250k. For this market, based on simulation model presented by MacLeod (MacLeod 2008), one can conclude that the maximum return is expected in a capacity of 2000 passengers per year, while the maximum profits, corresponding to end of growth phase of product life cycle, is assumed between 5 to 7 years of operation. According to Peeters, P2P transportation would be a sustainable market. For example, the time saving for passenger, journeying from New York to Tokyo will be 11,5 hours: aircraft duration is 12 hours 50 minutes vs. space vehicle duration, estimated in 83 minutes. We can roughly estimate the market for space tourism during first 5-7 years of operations, which can reach around $400M per year. This is only one of the segments of suborbital human spaceflight systems. Peeters emphasized the analogy between suborbital market and aeronautical sector, concluding that the "real sustainable market will be point-to-point (P2P) regular space travel, whereby the relatively high ticket prices will be compensated by the considerable time savings, important for a select but tangible 'time-poor, cash-rich' target public" (Peeters 2010).

Davidian and Foust (2010) applied the Christensen's Disruption Theory (2003) to the incumbent and new entrant companies of the suborbital payload market. These companies were analyzed from the perspective of three types of innovations: sustaining innovations, low-end disruptive innovation, and new-market disruptive innovations. As the final conclusions, the authors made recommendations on suborbital market new company's strategy for each one of these scenarios. In another work Davidian and Conrad (2012) applied Porter's framework (Porter 2008) to suborbital market. The authors analyzed the market from Porter's Five forces: threat of new entrants, bargaining power of suppliers, bargaining power of buyers, threat of substitutes, and rivalry. Their final conclusion was that the suborbital market has pre-emerging status: as of today, there are no suborbital operators providing commercial operations, but "it is clear that multiple firms are in the research, testing and development stages of activity".

In Russia the interest to suborbital systems exists in different areas. For example, one of the companies, participated in the Ansari X PRIZE contest, was Russian company called Suborbital Corporation, which was developing Cosmopolis-XXI spaceplane. Another example of the company in this sector is a six-years old company called Cosmocourse, which is a resident of Skolkovo Foundation and is developing a reusable suborbital space system, comprising of reusable suborbital booster and reusable suborbital spacecraft.

When we analyze the existing suborbital projects we observe that their concepts are presented differently: the level of details, the used terminology, and the concept of operations are explained based on the individual preferences of specific design team. This creates an opportunity to propose a universal model-based *system concept representation framework* that would allow to *encode* the alternative concepts in a systematic way, and would support a *formal analysis*. The knowledge generated with support of such a framework is *reused* at later stages of product development process.

Aforementioned context is a great motivation to demonstrate applicability of proposed framework to suborbital human spaceflight systems. Mankind has a great opportunity to have a stable and profitable suborbital spaceflight market. This market may include space tourism, P2P business travel, and many brunches of any kind of research experiments that require microgravity conditions.

5.4 Specific Objectives

In previous section the motivation and context for development of model-based *system concept representation framework* for suborbital human spaceflight systems were provided. In this section we discuss the specific objectives that we plan to achieve by this case study. Among these objectives are:

- Identify the architectural decisions that form the suborbital human spaceflight concepts;

- *Encode* a conceptual information about multiple alternatives into the proposed framework. These alternatives are 3 projects of suborbital human spaceflight missions: Virgin Galactic system, which we will call Virgin Galactic; Blue Origin system, which we will call either Blue Origin or New Shepard; and XCOR system, which we will call either XCOR or Lynx. This implies encoding such entries of framework as stakeholders and their needs (proposition I), solution-neutral and solution-specific environments (propositions II and III, correspondingly), integrated concept (proposition IV), and concept of operations (proposition V);

- Demonstrate how the proposed approach could support a *formal analysis*, such as conceptual similarity assessment;

- Demonstrate how the model-based approach supports the interconnections between the architectural decisions and model-based solution diagrams.

5.5 Model-Based System Concept Representation Frameworks Development

This section has the following structure. In sub-section 5.5.1 we outline the core inputs of *system concept representation framework*. After this we discuss the importance of architectural decisions for suborbital human spaceflight missions. In sub-section 5.5.3 we start the process of filling in the entries of concept framework for suborbital systems for the first proposition, which concerns the stakeholders and their needs. Solution-neutral and solution-specific environments are discussed in sub-sections 5.5.4 and 5.5.5, respectively. The integrated concepts at the first level decomposition for each one of the projects – Virgin Galactic, Blue Origin, and XCOR – are provided in sub-section 5.5.6. In sub-section 5.5.7 we discuss the outcomes of the first level decomposition of alternative concepts. The integrated concepts at the second level decomposition for all three projects are discussed in sub-section 5.5.8. The outcomes of the second level decomposition of the integrated concepts are provided in sub-section 5.5.9. Sub-section 5.5.10 is dedicated to the methods of capturing the conceptual design information through DSM. Sub-section 5.5.11 discusses the fifth proposition of concept framework, a concept of operations.

5.5.1 System Concept Representation Framework Introduction

The *system concept representation framework*, presented in Chapter 3, consists of 28 entries and defines concept as mapping function to form (Crawley et al. 2015).

We begin a framework development with a clear identification of stakeholders and their needs (entries 1, 2 of framework). Once this information is defined, we can identify the entries of the second proposition, a solution-neutral environment. These entries – from 3 to 7 – can be considered as a high-level abstraction of a concept. It includes the solution-neutral's operand and process and their attributes. Once this information is defined, the system architect is ready for conceptual design stage. In proposed notation of concept framework, the conceptual design is a movement through imaginary barrier from problem statement (solution-neutral environment) to solution statement (solution-specific environment). The solution-specific's operand and process appear as a result of specialization process from solution-neutral's operand and process. The specialized entries narrow down the set of alternative solutions satisfying stakeholders' needs. However, one of the main differences between problem statement and solution statement is the presence of form in solution-specific environment. It includes both generic and specific forms that are used as instruments executing functions. Form, which is the "physical or informational embodiment of a system that exists or has the potential for stable, unconditional existence, for some period of time, and is instrumental in the execution of function" (Crawley et al. 2015), is a concept implementation. Above-mentioned information is contained in entries 8 to 16 of concept framework.

Identification of specific form is not enough to claim that we have a concept of a system. An integrated concept reveals the core internal elements of form, internal processes, and internal operands (with corresponding attributes), which explain how the concept is intended to operate (Menshenin and Crawley 2018a). Sometimes the essential elements of the system are hidden on lower levels of architecture and can be defined only after the decomposition of specific form. Such decomposition reveals the internal elements of form (entries 22), internal processes (entries 20), and internal operands (entries 17) with corresponding attributes (entries 23, 21, 18 and 19, respectively). These internal processes and elements of form are very well correlated with architectural decisions, which guide the system development and are presented in Table 5.1. Thus, the decomposition part of the integrated concept covers entries 17 to 23 of concept framework.

Another part of the integrated concept is the information about structure (entry 24) and interactions (entry 25). A *structure* is the "set of formal relationships among the elements of form of a system" (Crawley et al. 2015). In other words, in structure we store the information about what's connected to what from the formal point of view. If the *structure* deals with formal relationships of form, the *interactions* (entry 25) inform systems engineer about what's exchanged among the elements on a functional level. Together with decomposition part, the structure and interactions conclude the integrated concept (proposition IV) of concept framework.

The fifth proposition of concept framework is the concept of operations. It includes ConOps itself (entry 26), Operator (entry 27), and Context (entry 28). The rationale for including this information into framework was explained in Chapter 3. This information is important, as it produces a deep insight of operations, which are the sequence of activities that deliver the primary function. The concept of operations for each alternative concept of suborbital human spaceflight missions will be provided in current Chapter. The operator is usually a human operator (entry 27). The last, but not least, entry of concept framework is the *context* (entry 28), which is something what surrounds the system: spaceport, communication system, and customer support, as examples for suborbital missions. All these systems aggregate to the *whole product system*.

All these 28 entries are essential and intrinsic parts of the *system concept representation framework*. In Chapter 5 the model-based concepts for 3 alternative concepts (Virgin Galactic system, Blue Origin system, and XCOR system) of suborbital spaceflight mission are *encoded*, and the results of *formal analysis* are discussed.

5.5.2 Architectural Decisions for Suborbital Spaceflight Missions

The *architectural decisions* are the "subset of design decisions that are most impactful" (Crawley et al. 2015). The importance of architectural decisions is highlighted in software architecture. For instance, Kruchten (2008) mentions that "thin line in the sand that separates architectural decisions from all other design decisions, including the detailed ones captures in the code, must be made visible for all parties involved". Thus, it should be noted that the architectural decisions are important for any kind of system under development: either this is a hardware system, or software system. So, in case of development a concept for software architecture that is intended to serve suborbital human spaceflight missions, the system architect will also need to develop architectural decisions.

In order to define the architectural decisions, few processes and associated instruments were identified. These processes and instruments are the most critical elements that should be known for any system under development for suborbital human spaceflight missions.

The system architect should recognize the existence of few steps, which are intrinsic to any kind of suborbital concept. Any concept for suborbital spaceflight missions will have such processes as “launching”, “flying”, and “landing”, as these are fundamental processes required for successful implementation of the system. In addition to these processes, the architectural decisions capture such essential questions as number of modules and crew members (both – pilots and passengers).

It is important to note that for the purpose of consistency in the work, by “module” we imply the physical implementation of system’s parts. In such terminology, the Virgin Galactic concept has two modules: the mothership WhiteKnightTwo, and the spacecraft SpaceShipTwo. Alternative concept Blue Origin, has two modules: a propulsion module and a separated crew capsule. Another alternative concept – XCOR – has one module, which is a single stage rocketplane, called a Lynx.

From Table 5.1, which summarizes the architectural decisions for suborbital systems, we may see that the process “flying” is further decomposed into internal processes “lifting”, “guiding”, “increasing (energy of module)”, “decreasing (energy of module)”.

Table 5.1: Architectural decisions for suborbital spaceflight missions

Step		N	Parameter	Opt 1	Opt 2	Opt 3	Opt 4	Opt 5	Opt 6	Opt 7
General		1	N of Modules	1	2					
Launching		2	Type of Launch	Horizontal	Vertical					
		3	Place of Launch	Ground	Water					
		4	Instrument of launching	Landing gear	Rocket engine					
i = 1,2										
Module i	General (i)	5	N of pilots	0	1	2				
		6	N of passengers	0	1	2	3	4	5	6
	Flying (i)	7	Lifting	Wing	None					
		8	Guiding	Aerodynamic surface	Rocket engine	Thrusters	None			
		9	Increasing energy of module	Jet engine	Rocket engine	None				
		10	Decreasing energy of module	Aerodynamic decelerators	Rocket engine	Jet engine	Wings	None		
	Landing (i)	11	Type of landing	Horizontal	Vertical					
		12	Place of landing	Ground	Water					
		13	Instrument of landing	Landing gear	Rocket engine	Parachute				

One of the benefits of having the architectural decisions map is that it contains the information about the alternative instruments that might be used to satisfy a specific function. In the following sub-sections we will demonstrate how these architectural decisions complement the development of model-based concepts for suborbital human spaceflight missions.

5.5.3 Proposition I: Stakeholders and Stakeholders' Needs for Suborbital Human Spaceflight Systems

The stakeholders and their needs constitute the first proposition of concept framework, filling in entries 1 and 2. The stakeholders and their needs can be defined based on different sources: talking to representatives of business, government, and scientific organizations or communities, which are directly or indirectly involved in use of the system or are benefiting from this system; extensive literature review on a topic; and market analysis.

Davidian and Foust (2010) mentioned universities and government organizations as the traditional customers for the suborbital spaceflight market. These customers might have a broad spectrum of needs: physical and biological process in microgravity, observation and data collection of Earth and its atmosphere, and astronomical observation (Davidian and Foust 2010). Walter Peeters (2010) has also highlighted the interest to the market opportunities from such commercially oriented organizations as insurance companies and spaceports.

In a 10-years market forecast, prepared by Tauri Group (2014), it is stated that approximately 80% of demand in suborbital spaceflight business is related to commercial human spaceflight market.

This case study is focused on this particular market niche with the assumption that this is the most promising one. We consider such needs as space tourism, point-to-point transportation (high-speed passenger transportation), and individual trainings as the dominant needs that have a huge market potential. Table 5.2 summarizes the stakeholders and their needs based on extensive literature review, and highlights the primary stakeholders and their needs for this case study.

Table 5.2: Stakeholders of suborbital spaceflight missions and the needs of these stakeholders

Stakeholder(s)	Need(s)
Individual / Commercial human spaceflight providers	• **Have fun** • Get somewhere • Training
Government entities	• Biological and physical research in microgravity • Observation and data collection of Earth and its atmosphere • Astronomic observation • Hardware qualification and test
Commercial entities	• Television and film production • Media, advertising and sponsorship • Insurance • Spaceports or existing airports
Universities and schools	• Increase access to and awareness of space • Observation and data collection of Earth and its atmosphere • Astronomic observation

In section 3.3.1 (Chapter 3) we introduced our definition of stakeholders, following the approaches, given in Freeman's 1984 publication on Stakeholder management (Freeman 1984) and Crawley et al.'s book on System Architecture (Crawley et al. 2015). For the purpose of consistency, it is important to narrow down our definition of stakeholders, showing its applicability to suborbital spaceflight case study. First, by stakeholders we imply individuals, who can affect the system or be affected by the system. They can do this by means of purchasing or not purchasing our product and/or service, for instance. Secondly, stakeholders have some needs, which can be satisfied by our product and/or service.

In our case study we define "individuals" as stakeholders of suborbital spaceflight system. They have a need "have fun". These are entries 1 and 2 of concept framework, which are summarized in Table 5.3 and Figure 5.1 for the first proposition.

Table 5.3: Entries 1 and 2 of concept framework for suborbital concept

Proposition I: Stakeholders		
1	**Stakeholders**	**Individual**
2	Need	Have fun

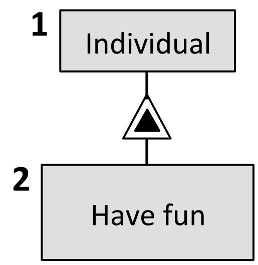


Figure 5.1: Stakeholders (I) proposition for suborbital concept

The model-based representation follows the OPM semantics that has been discussed earlier.

5.5.4 Proposition II: Solution-Neutral Operand and Process for Suborbital Human Spaceflight Missions

The information in solution-neutral environment for suborbital spaceflight system deals with the function of system without reference to the instrument of system (Suh 2001). The purpose of solution-neutral environment is to identify the operand of concept, which is "individual" (entry 3 in Table 5.4), and the process, which is "entertaining" (entry 6). The value attribute of the solution-neutral operand is "enjoyment level" (entry 4). The enjoyment level is the value-related attribute, because it is changed by the process. The other attribute is "number" (entry 5), it is not changed by the process, yet this information is important to know. The attribute of process is "safely" (entry 7). Thus, the functional intent of the system is to "entertain individual, safely". It is important to note that the solution-neutral function is broad by definition, which was discussed in section 3.4 in Chapter 3.

In Table 5.4 we summarize the solution-neutral's entries of concept, which reflect the solution-neutral's operand and process and corresponding attributes. In Figure 5.2 we demonstrate the model-based representation of solution-neutral environment for suborbital human spaceflight missions. Both, Table 5.4 and Figure 5.2, contain exactly the same information.

Table 5.4: Entries 3 to 7 of concept framework for suborbital concept

Proposition II: Solution-neutral environment (Problem statement)		
3	**Solution-neutral operand (SNO)**	**Individual**
4	SNO value attribute	Enjoyment level
5	SNO other attribute	Number
6	**Solution-neutral process (SNP)**	**Entertaining**
7	SNP attribute	Safely

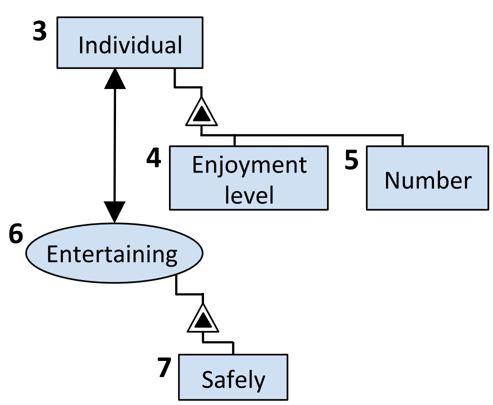


Figure 5.2: Solution-neutral (II) proposition for suborbital concept

The numbers reflect the enumeration of concept framework's entries, and are consistent among the tables and model-based views. One of the benefits of using the OPD is that the designers from different parts of the world can *encode* information about concept in the same way. Thus, by assigning a number to each entry of concept framework we may keep track of every entry and its inheritance.

5.5.5 Proposition III: Conceptual Design Phase for Suborbital Human Spaceflight Missions

Conceptual design allows the system architect to move from a highly abstracted solution-neutral environment to a solution-specific one. The mental diagram of conceptual design is shown in Figure 5.3, which demonstrates that the conceptual design is associated with the specialization process, and that it is the movement from abstracted (solution-neutral's) operand and process to a more concrete (solution-specific's) operand and process. One of the key elements presented in Figure 5.3 are the generic and specific form. Conceptual design enables to define them.

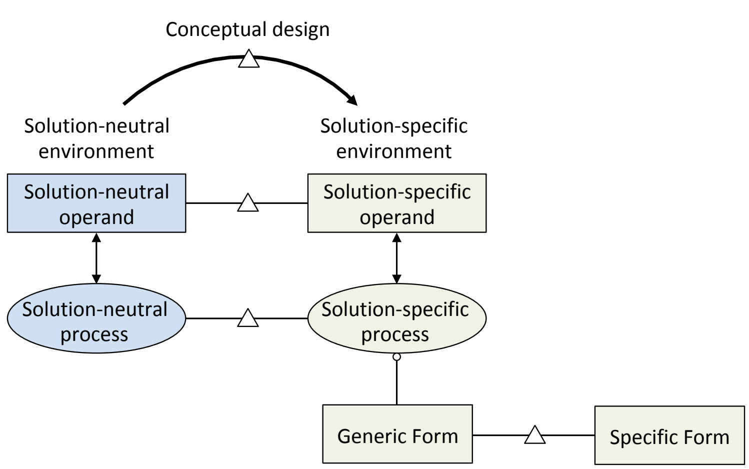


Figure 5.3: Conceptual design process

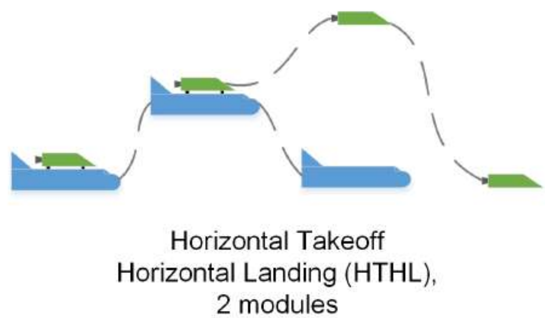
Applying the above-mentioned approach to our case study, we may find the following results. The solution-neutral operand "individual" specializes to the solution-specific operand "passenger" (see Table 5.5 and Figure 5.5). According to the first rule of concept framework development, formulated in section 3.5.3 in Chapter 3, there is a principle of inheritance of attributes. Thus, the attributes mentioned as entries 9, 10, 12 in Table 5.5 are inherited from solution-neutral's environment. The solution-neutral process "entertaining" specializes to the solution-specific process "flying" (entry 11), which has an attribute "safely" (entry 12) (inherited as well).

As it was discussed earlier, the main outcomes of conceptual design process is the ability to identify the form that executes the function. In Table 5.5 and Figure 5.5 we see that the generic form "suborbital vehicle" (entry 13) is assigned to the function "flying passenger". The attribute that is associated with this generic form is "cost" (entry 14).

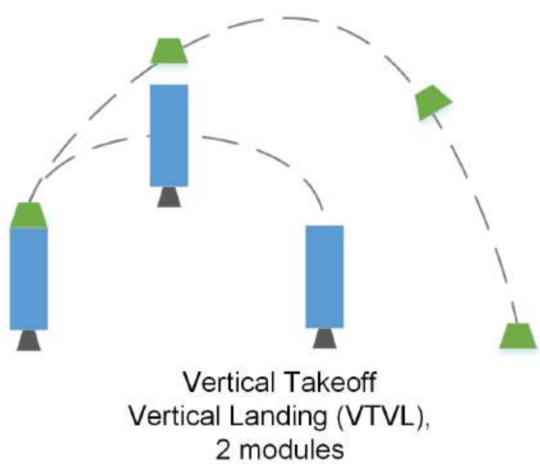
The generic form plays a role of funnel: on the one hand, it is the first appearance of form, thus this information is much more concrete than the information that is contained in solution-neutral environment. On the other hand, the generic form is still abstract comparing to the downstream information – a specific form.

Based on the analysis of 32 projects (see Appendix B), the majority of which appeared from X Prize's Ansari contest, we extracted the following conceptually different approaches to suborbital human spaceflight missions (See Figure 5.4) (Guerster and Crawley 2018). Three concepts were chosen for further study, in which we develop the model-based concepts for each project of suborbital mission.

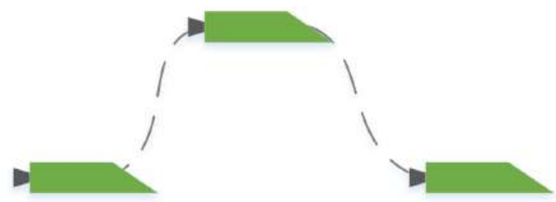
Thus, one can see that the first approach - HTHL with 2 modules, see Figure 5.4(a) – corresponds to Virgin Galactic system, the second approach – VTVL with 2 modules, see Figure 5.4(b) – corresponds to Blue Origin system, and the third approach – HTHL with 1 module, see Figure 5.4(c) – corresponds to XCOR system. In the next three sub-sections we will precisely consider the integrated concepts (proposition IV of concept framework) for each of these projects.



(a)



(b)



(c)

Figure 5.4:

(a). Virgin Galactic concept sketch; (b). Blue Origin concept sketch; (c). XCOR concept sketch

In Table 5.5 and Figure 5.5 all three alternative concepts are indicated as the entries 15 of concept framework. Note that the attribute "cost" (entry 16) is inherited from entry 14. In fact, each one of the alternative concepts has its own entry number – this allows us to compare the alternatives on the same level of granularity and to enable the computations. For example, Virgin Galactic is the entry 15A, Blue Origin is the entry 15B, and XCOR is the entry 15C.

Thus, the entries 8 to 16 of concept framework represent the solution-specific information and are demonstrated in Table 5.5 and Figure 5.5.

Table 5.5: Entries 8 to 16 of concept framework for suborbital concept

8	**Solution-specific operand (SSO)**	**Passenger**	
9	SSO value attribute	Enjoyment level	
10	SSO other attribute	Number	
11	**Solution-specific process (SSP)**	**Flying**	
12	SSP attribute	Safely	
13	**Generic Form**	**Suborbital vehicle**	
14	Generic Form attribute	Cost	
15	**Specific Form**	**15A**	**Virgin Galactic system**
		15B	**Blue Origin system**
		15C	**XCOR system**
16	Specific Form attribute	**16A**	**Cost**
		16B	**Cost**
		16C	**Cost**

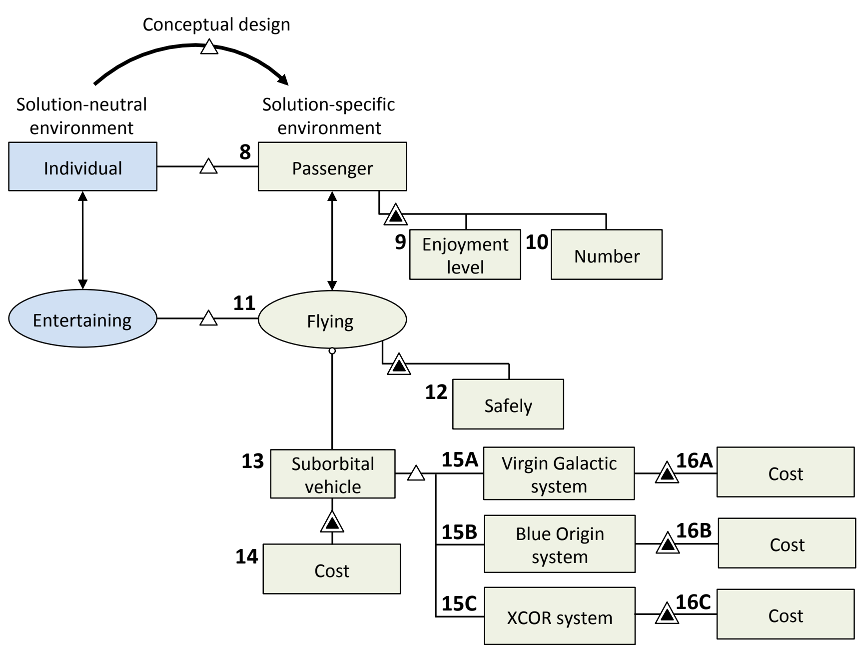


Figure 5.5: Solution-specific (III) proposition for suborbital concept

Up until now concept framework contains the stakeholders information (proposition I), solution-neutral information (proposition II), and solution-specific information (proposition III). Except for entries 15 and 16, all entries are the same. The real difference between alternatives starts to appear beginning with entries 15 and 16 and subsequent decomposition of specific forms into the integrated concepts. Note that the attribute "cost" is related to the production cost.

5.5.6 Proposition IV: Integrated Concepts for Alternative Suborbital Systems: First Level Decomposition

The integrated concept contains essential information about the decomposition of specific form into its internal elements of form (entries 22), internal processes (entries 20), and internal operands (entries 17) with corresponding attributes (entries 18, 19, 21, and 23). In this sub-section we demonstrate how to *encode* the first level decomposition information for each alternative: Virgin Galactic system, Blue Origin system, and XCOR system.

- Integrated concept for Virgin Galactic system: first level decomposition

At the first level decomposition of specific form "Virgin Galactic system" (entry 15A) is decomposed into internal elements of form "WhiteKnightTwo" (entry 22A1) and "SpaceShipTwo" (entry 22A2). Note that "A1" at the end of entry specifies the information for WhiteKnightTwo, while "A2" is related to the SpaceShipTwo.

At the level of integrated concept we see a growing role of model-based representation, as it is getting harder to read the information in the grid. From both – Table 5.6 and Figure 5.6 – we may see that the function of WhiteKnightTwo is "flying SpaceShipTwo" (entries 20A1 and 17A1), while the function of SpaceShipTwo is “flying passengers” (entries 20A2 and 17A2). “SpaceShipTwo” as an operand (entry 17A1) has an internal operand value attribute “energy” (entry 18A1), and internal operand other attribute “mass” (entry 19A1). The internal process “flying” has an attribute “safely” (entry 21A1). The internal element of form “WhiteKnightTwo” has an attribute “cost” (entry 23A1).

In turn, “SpaceShipTwo” as an internal element of form (entry 22A2) has an attribute “cost” (entry 23A2). The internal process “flying” has an attribute “safely” (entry 21A2). Internal operand “passengers” (entry 17A2) has the internal operand value attribute “enjoyment level” (entry 18A2) and internal operand other attribute “number” (entry 19A2).

We keep the same semantics and the level of granularity for both subsystems.

Table 5.6: Entries of integrated concept for Virgin Galactic in grid

		A1: WhiteKnightTwo	**A2: SpaceShipTwo**
17	**Internal Operands (IO)**	**SpaceShipTwo**	**Passengers**
18	IO value attribute	Energy	Enjoyment level
19	IO other attribute	Mass	Number
20	**Internal Processes (IP)**	**Flying**	**Flying**
21	IP attribute	Safely	Safely
22	**Internal Elements of Form (IEoF)**	**WhiteKnightTwo**	**SpaceShipTwo**
23	IEoF attribute	Cost	Cost
24	Structure	SS2 attached below WK2	
25	Interactions	WK2 transfers force to SS2	

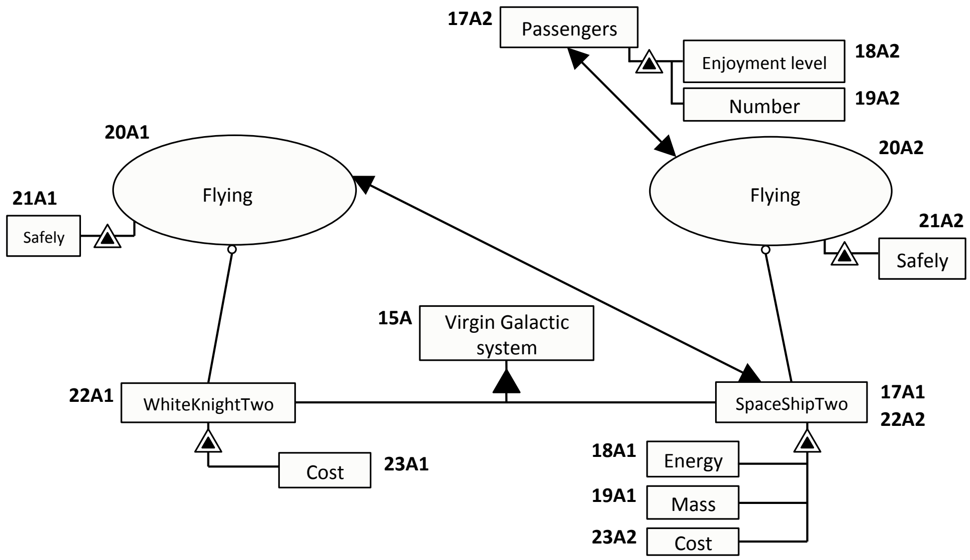


Figure 5.6: Integrated concept (IV) proposition for Virgin Galactic concept

In Figure 5.7 we demonstrate the model-based representation of structure and interactions information for Virgin Galactic system. From upper side of Figure, we see the information about formal allocation of elements to each other: SpaceShipTwo is "attached below" WhiteKnightTwo. From the lower side of Figure, we notice what's exchanged among the elements on functional level: WhiteKnightTwo "transfers force" to SpaceShipTwo.

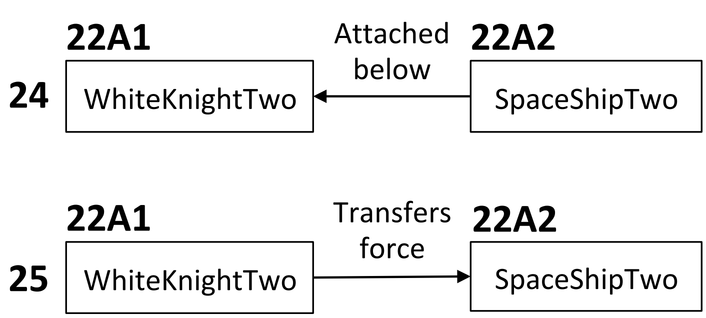


Figure 5.7: Structure and Interactions for Virgin Galactic concept

- Integrated concept for Blue Origin system: first level decomposition

Table 5.7 and Figures 5.8 and 5.9 demonstrate the first level decomposition of specific form “Blue Origin system” (entry 15B). Here we can see that the specific form is decomposed into two internal elements of form – propulsion module (entry 22B1) and capsule (entry 22B2). Their attributes are costs (entries 23B1 and 23B2, respectively). The internal function of propulsion module is “carrying capsule” (entries 20B1 and 17B1), while the internal function of capsule is “flying passengers” (entries 20B2 and 17B2, correspondingly). Capsule as an operand has value attribute “energy” (entry 18B1) and other attribute “mass” (entry 19B1). The operand “passengers” has the value attribute “enjoyment level” (entry 18B2) and “number” (entry 19B2). Internal process “carrying” has an attribute “safely” (entry 21B1), and internal process “flying” has also an attribute “safely” (entry 21B2).

Table 5.7: Entries of integrated concept for Blue Origin in grid

		B1: Propulsion module	**B2: Capsule**
17	**Internal Operands (IO)**	**Capsule**	**Passengers**
18	IO value attribute	Energy	Enjoyment level
19	IO other attribute	Mass	Number
20	**Internal Processes (IP)**	**Carrying**	**Flying**
21	IP attribute	Safely	Safely
22	**Internal Elements of Form (IEoF)**	**Propulsion module**	**Capsule**
23	IEoF attribute	Cost	Cost
24	Structure	Capsule attached above Propulsion module	
25	Interactions	Propulsion module transfers force to Capsule	

The identical information is contained in the model-based representation of the first level decomposition of Blue Origin system, presented in Figure 5.8.

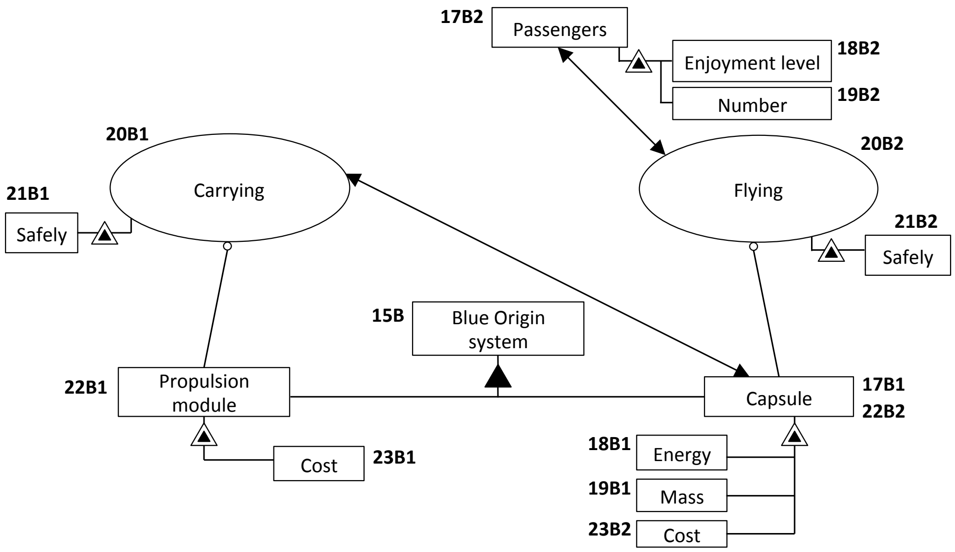


Figure 5.8: Integrated concept (IV) proposition for Blue Origin concept

The structure and interactions information is presented in Figure 5.9. From this Figure we see that the formal allocation of elements to each other is that the capsule is "attached above" the propulsion module. The functional interaction is that propulsion module "transfers force" to capsule.

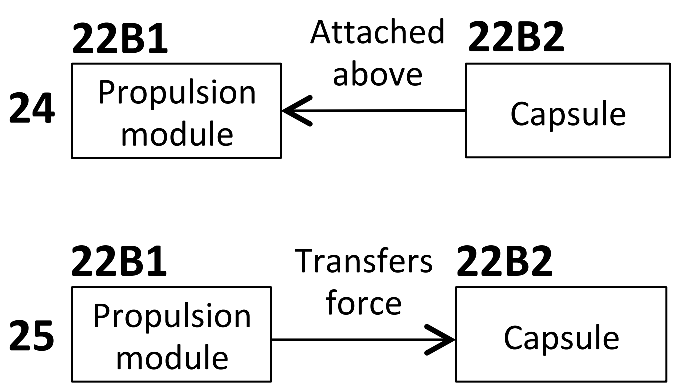


Figure 5.9: Structure and Interactions for Blue Origin concept

- Integrated concept for XCOR system: first level decomposition

XCOR system doesn't decompose into internal elements of form on the same level of granularity as it is for the first two cases. This is due to the fact that XCOR system has one module, which is XCOR itself. Nevertheless, in order to be consistent with previous two examples we will demonstrate the first level decomposition of XCOR, which is actually the same information as it is contained in the solution-specific environment.

The internal element of form XCOR (entry 22C) has an attribute "cost" (entry 23C). The internal process is "flying" (entry 20C) with an attribute "safely" (entry 21C). The operand is "passenger" (entry 17C). Its value attribute is "enjoyment level" (entry 18C), and other attribute is "number" (entry 19C).

Table 5.8: Entries of integrated concept for XCOR in grid

		C: XCOR
17	**Internal Operands (IO)**	**Passengers**
18	IO value attribute	Enjoyment level
19	IO other attribute	Number
20	**Internal Processes (IP)**	**Flying**
21	IP attribute	Safely
22	**Internal Elements of Form (IEoF)**	**XCOR**
23	IEoF attribute	Cost

The model-based representation of the same information is indicated in Figure 5.10 that shows the core information about system and its function. Note that for the above-mentioned reasons here we do not represent the structure and interactions information.

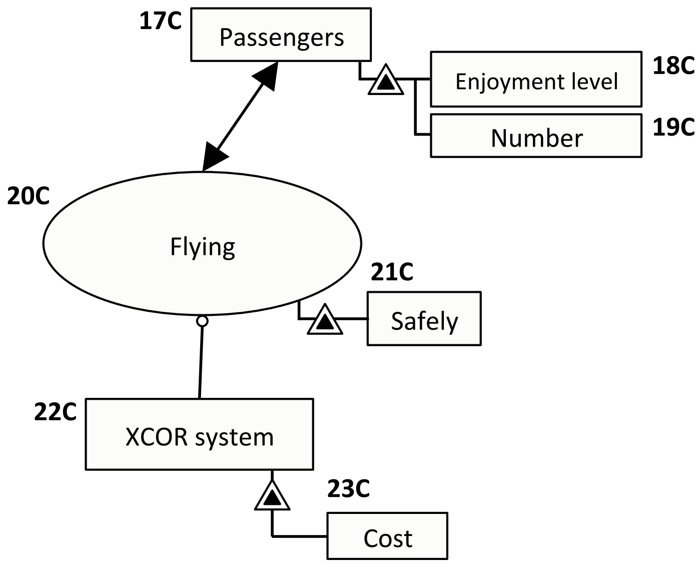


Figure 5.10: Integrated concept (IV) proposition for XCOR concept

5.5.7 Outcomes of the First Level Decomposition of Alternative Concepts

In previous Chapter we discussed that the level of granularity plays an important role in systems decomposition. We supported this assumption based on the claims of patents, illustrating that the most essential information about patents might be hidden in second level decomposition and stored in claims.

We may see a similar result through the analysis of three alternatives presented in sub-section 5.5.6 of Chapter 5. For example, comparing the conceptual difference between Virgin Galactic system and Blue Origin systems (see Figure 5.11) we can note that in both cases the first module is different (WhiteKnightTwo in case of Virgin Galactic and Propulsion module in case of Blue Origin), as well as the internal processes (“flying” in case of WhiteKnightTwo vs. “carrying” in case of Propulsion module). These processes act on different operands (SpaceShipTwo and Capsule, respectively).

15A: Virgin Galactic		
	A1: WhiteKnightTwo	**A2: SpaceShipTwo**
17	**SpaceShipTwo**	**Passengers**
18	Energy	Enjoyment level
19	Mass	Number
20	**Flying**	**Flying**
21	Safely	Safely
22	**WhiteKnightTwo**	**SpaceShipTwo**
23	Cost	Cost

15B: Blue Origin		
	B1: Propulsion module	**B2: Capsule**
17	**Capsule**	**Passengers**
18	Energy	Enjoyment level
19	Mass	Number
20	**Carrying**	**Flying**
21	Safely	Safely
22	**Propulsion module**	**Capsule**
23	Cost	Cost

Figure 5.11: Comparison of conceptual difference between Virgin Galactic (left) and Blue Origin (right) concepts

Concerning the second module we see that as the first level decomposition the only conceptual difference between two alternatives is the modules themselves (SpaceShipTwo in case of Virgin Galactic and Capsule in case of Blue Origin). The internal functions are the same: “flying passengers”. Both “passengers” and “flying” entries have the identical attributes (entries 18, 19, and 21, respectively).

The comparison of Virgin Galactic system (15A) and XCOR system (15C) reveals a different result – see Figure 5.12. The core difference is the number of modules that are used to satisfy the stakeholders’ needs. In case of Virgin Galactic we have two modules: WhiteKnightTwo (entry 22A1) and SpaceShipTwo (entry 22A2). In case of XCOR there is only one module, which is XCOR itself (entry 22C). Consequently, we may see that XCOR itself performs all internal functions that are executed by two internal elements of form in case of Virgin Galactic system.

15A: Virgin Galactic		
	A1: WhiteKnightTwo	**A2: SpaceShipTwo**
17	**SpaceShipTwo**	**Passengers**
18	Energy	Enjoyment level
19	Mass	Number
20	**Flying**	**Flying**
21	Safely	Safely
22	**WhiteKnightTwo**	**SpaceShipTwo**
23	Cost	Cost

15C: XCOR	
	C1: XCOR
17	**Passengers**
18	Enjoyment level
19	Number
20	**Flying**
21	Safely
22	**XCOR**
23	Cost

Figure 5.12: Comparison of conceptual difference between Virgin Galactic (left) and XCOR (right) concepts

A similar result can be seen for the comparison of Blue Origin system and XCOR system (see Figure 5.13). The reason is the same as it is for the previous example. The XCOR's primary function is "flying passengers", which is only identical for the second module of Blue Origin system.

15B: Blue Origin		
	B1: Propulsion module	**B2: Capsule**
17	**Capsule**	**Passengers**
18	Energy	Enjoyment level
19	Mass	Number
20	**Carrying**	**Flying**
21	Safely	Safely
22	**Propulsion module**	**Capsule**
23	Cost	Cost

15C: XCOR	
	C1: XCOR
17	**Passengers**
18	Enjoyment level
19	Number
20	**Flying**
21	Safely
22	**XCOR**
23	Cost

Figure 5.13: Comparison of conceptual difference between Blue Origin (left) and XCOR (right) concepts

Thus, the model-based concept framework can be used to capture the conceptual difference between alternative concepts. At the same time, we would like to have a tool that might be used by system architect to keep track of conceptual similarity between alternatives. Such tool for the first level decomposition is presented in Figure 5.14. Here we use Design Structure Matrix (DSM) to keep track of important information about concepts' internal instruments of form (DSM) and internal processes (DMM). Additionally, the matrix has the information about structure and interactions (indicated in DSM at the intersection of corresponding internal elements of form).

Each number mentioned in matrix of Figure 5.14 corresponds to above-mentioned number assigned to entries of the concept framework.

Full DMM				13					11				
				Suborbital vehicle					Flying				
				15A		15B		15C	11		11		11
				Virgin Galactic		Blue Origin		XCOR	Flying		Flying		Flying
				22A1	22A2	22B1	22B2	22C	20A1	20A2	20B1	20B2	20C
				WK2	SS2	Propulsion module	Capsule	XCOR	Flying	Flying	Carrying	Flying	Flying
13	Suborbital vehicle	15A Virgin Galactic	22A1 WK2		Attached below Receives force				V				
			22A2 SS2	Attached above Transfers force						V			
		15B Blue Origin	22B1 Propulsion module				Attached above Receives force				V		
			22B2 Capsule			Attached below Transfers force						V	
		15C XCOR	22C XCOR										V

DSM | DMM

Figure 5.14: DSM/DMM for integrated concepts of suborbital concepts

From DSM part of the Figure 5.14 we see both specialization relationships among entries (for example, entry 13 "suborbital vehicle" specializes into entries 15A "Virgin Galactic", 15B "Blue Origin", and 15C "XCOR") and decomposition relationships (for instance, entry 15A “Virgin Galactic” decomposes into the internal elements of form entry 22A1 “WhiteKnightTwo” and entry 22A2 “SpaceShipTwo”). This information is correlated with the OPDs presented in Figures 5.6, 5.8, and 5.10.

Another important information *encoded* into DSM part of the matrix of Figure 5.14 is the information about structure and interactions among internal elements of form. Structure is mentioned in text in black at the intersection of corresponding elements, while the interactions are indicated in colored text there. This data is correlated with OPDs presented in Figures 5.7 and 5.9.

The DMM part of the Figure 5.14 informs us about allocation of internal elements of form to internal processes (this is marked by “V” signs). DMM is also used to facilitate a *formal analysis*, such as a conceptual similarity assessment. In example of Figure 5.14 this assessment is indicated in colored yellow cells. Colored cells in DMM part mean that among two alternatives under comparison the internal element of form is used for different internal processes. For example, in our case study the yellow color means that the first modules of Virgin Galactic (WhiteKnightTwo) and Blue Origin (Propulsion module) have different internal processes – “flying” and “carrying”.

Indeed, we may conclude that the decomposition into the first level is not enough to perform a full conceptual similarity assessment, because the number of elements is small. Thus, the next level of decomposition is required to explore more information about a concept, keeping a design process on conceptual level. Such decomposition is explained in details at the next sub-section.

5.5.8 Integrated Concepts for Alternative Suborbital Systems: Second Level Decomposition

In sub-section 5.5.8 we further decompose Virgin Galactic, Blue Origin, and XCOR concepts into the second level decomposition. Similarly to sub-section 5.5.6 we *encode* these concepts in a model-based approach. In this sub-section we also demonstrate the architectural decisions tables for each alternative.

- Integrated concept for Virgin Galactic concept: second level decomposition

Virgin Galactic concept is decomposed into two sub-systems: WhiteKnightTwo and SpaceShipTwo. The architectural decisions for the first module, a WhiteKnightTwo, are presented in Figure 5.15, and the architectural decisions for the second module, a SpaceShipTwo, are presented in Figure 5.16. These architectural decisions contain information about the form to process allocation. For example, from Figure 5.15 we may notice that the process "lifting" is performed by the form "wing" for the first module of Virgin Galactic concept.

Step		N	Parameter	Opt 1	Opt 2	Opt 3	Opt 4	Opt 5	Opt 6	Opt 7
General		1	N of Modules	1	2					
Launching		2	Type of Launch	Horizontal	Vertical					
		3	Place of Launch	Ground	Water	At altitude				
		4	Instrument of launching	Landing gear	Rocket engine					
i = 1,2										
Module 1	General (1)	5	N of pilots	0	1	2				
		6	N of passengers	0	1	2	3	4	5	6
	Flying (1)	7	Lifting	Wing	None					
		8	Guiding	Aerodynamic surface	Rocket engine	Thrusters	None			
		9	Increasing energy of module	Jet engine	Rocket engine	None				
		10	Decreasing energy of module	Aerodynamic decelerators	Rocket engine	Jet engine	Wings	None		
	Landing (1)	11	Type of landing	Horizontal	Vertical					
		12	Place of landing	Ground	Water					
		13	Instrument of landing	Landing gear	Rocket engine	Parachute				

Figure 5.15: Architectural decisions for the first module (WhiteKnightTwo) of Virgin Galactic concept

Step		N	Parameter	Opt 1	Opt 2	Opt 3	Opt 4	Opt 5	Opt 6	Opt 7
General		14	N of Modules	1	2					
Launching		15	Type of Launch	Horizontal	Vertical					
		16	Place of Launch	Ground	Water	At altitude				
		17	Instrument of launching	Landing gear	Rocket engine					
i = 1,2										
Module 2	General (2)	18	N of pilots	0	1	2				
		19	N of passengers	0	1	2	3	4	5	6
	Flying (2)	20	Lifting	Wing	None					
		21	Guiding	Aerodynamic surface	Rocket engine	Thrusters	None			
		22	Increasing energy of module	Jet engine	Rocket engine	None				
		23	Decreasing energy of module	Aerodynamic decelerators	Rocket engine	Jet engine	Wings	None		
	Landing (2)	24	Type of landing	Horizontal	Vertical					
		25	Place of landing	Ground	Water					
		26	Instrument of landing	Landing gear	Rocket engine	Parachute				

Figure 5.16: Architectural decisions for the second module (SpaceShipTwo) of Virgin Galactic concept

It is essential that system architecture principles remain the same regardless the level of granularity. As such, from Table 5.6 and Figure 5.6 we may notice that at the second level we will decompose the WhiteKnightTwo and SpaceShipTwo into their own internal entries.

At the second level decomposition the number of elements is increasing, so the risk of losing control over the concept's entries enumeration is increasing as well. In sub-section 3.6.3 we explained the principle of entries enumeration. We first assign numbers ".1, .2, …" to internal processes (repeating processes having the same internal elements of form are assigned with the same numbers). After this we sequentially assign the numbers ".1, .2, …" to internal elements of form (repeating internal elements of form have the same numbers). After this we sequentially assign the numbers ".1, .2, …" to the internal operands (repeating internal operands have the same numbers). The same principles apply for the attributes.

The second level decomposition of Virgin Galactic's first module "WhiteKnightTwo" is demonstrated in Figure 5.17.

17A1	**Internal Operands (IO)**	**.1**	**.1**	**.1**	**.1**	**.1**	**.1**	**.2**
		WK2	**WK2**	**WK2**	**WK2**	**WK2**	**WK2**	**SS2**
18A1	IO value attribute	**.1**	**.2**	**.3**	**.4**	**.4**	**.5**	**.6**
					Energy	Energy		
19A1	IO other attribute	**.1**	**.2**	**.3**	**.4**	**.6**	**.7**	**.8**
20A1	**Internal Processes (IP)**	**.1**	**.2**	**.3**	**.4**	**.5**	**.6**	**.7**
		Launching	**Lifting**	**Guiding**	**Increasing**	**Decreasing**	**Landing**	**Flying**
21A1	IP attribute	**.1**	**.2**	**.3**	**.4**	**.6**	**.7**	**.8**
		Horizontally, On ground					Horizontally, On ground	
22A1	**Internal Elements of Form (IEoF)**	**.1**	**.2**	**.3**	**.4**	**.4**	**.1**	**.5**
		Landing gear	**Wings**	**Aerodynamic surfaces**	**Jet engine**	**Jet engine**	**Landing gear**	**Pilots**
23A1	IEoF attribute	**.1**	**.2**	**.3**	**.4**	**.5**	**.1**	**.5**

Figure 5.17: Second level decomposition of Virgin Galactic's first module (WhiteKnightTwo)

From Figure 5.17 we may notice an important aspect of assigning the numbers to columns. These numbers reflect the entries of the concept framework. If the specific entry is the same, we keep the same number in different columns, as it is for the entry

17A1.1 (WhiteKnightTwo). At the same time, there might be multiple attributes of this entry – that is why we see entries 18A1.1, 18A1.2, 18A1.3 at the attributes space.

The second level decomposition of the Virgin Galactic's second module "SpaceShipTwo" is presented in Figure 5.18.

17A2	**Internal Operands (IO)**	**.1**	**.1**	**.1**	**.1**	**.1**	**.1**	**.1**	**.2**
		SS2	**SS2**	**SS2**	**SS2**	**SS2**	**SS2**	**SS2**	**Passengers**
18A2	IO value attribute	.1	.2	.3	.4	.5	.5	.6	.7
						Energy	Energy		
19A2	IO other attribute	.1	.2	.3	.4	.5	.6	.7	.8
20A2	**Internal Processes (IP)**	**.1**	**.2**	**.3**	**.3**	**.4**	**.5**	**.6**	**.7**
		Separating	**Lifting**	**Guiding**	**Guiding**	**Increasing**	**Decreasing**	**Landing**	**Flying**
21A2	IP attribute	.1	.2	.3	.4	.5	.6	.7	.8
		Horizontally On ground						Horizontally On ground	
22A2	**Internal Elements of Form (IEoF)**	**.1**	**.2**	**.3**	**.4**	**.5**	**.5**	**.6**	**.7**
		WK2	**Wings**	**Aerodynamic surfaces**	**Thrusters**	**Rocket engine**	**Rocket engine**	**Landing gear**	**Pilots**
23A2	IEoF attribute	.1	.2	.3	.4	.5	.6	.7	.8

Figure 5.18: Second level decomposition of Virgin Galactic's second module (SpaceShipTwo)

Note that some of the cells in Figures 5.17 and 5.18 are blank due to the principles of inheritance of attributes: the entries of these cells are contained in Table 5.6.

The table representations for the second level decomposition reveal some drawbacks. Although they are logically correct and consistent (see Figures 5.17 and 5.18), the increased number of elements and processes make it hard to read and analyze. Thus, we *encode* the second level granularity for Virgin Galactic concept in model-based environment and present it in Figure 5.19. Such representation also has some advantages, as it contains data for both sub-systems keeping the same level of information in one diagram.

At the model of Figure 5.19 we see Virgin Galactic concept (entry 15A) from solution-specific environment that is decomposed into two sub-systems: WhiteKnightTwo (entry 22A1) and SpaceShipTwo (entry 22A2). Each one of these sub-systems is further decomposed into internal elements of form (entries 22A1.1, etc./22A2.1, etc.) that execute the internal processes (entries 20A1.1, etc./20A2.1, etc.) acting on internal operands (entries 17A1.1, etc./17A2.1, etc.). All this information in details is contained in Figure 5.19.

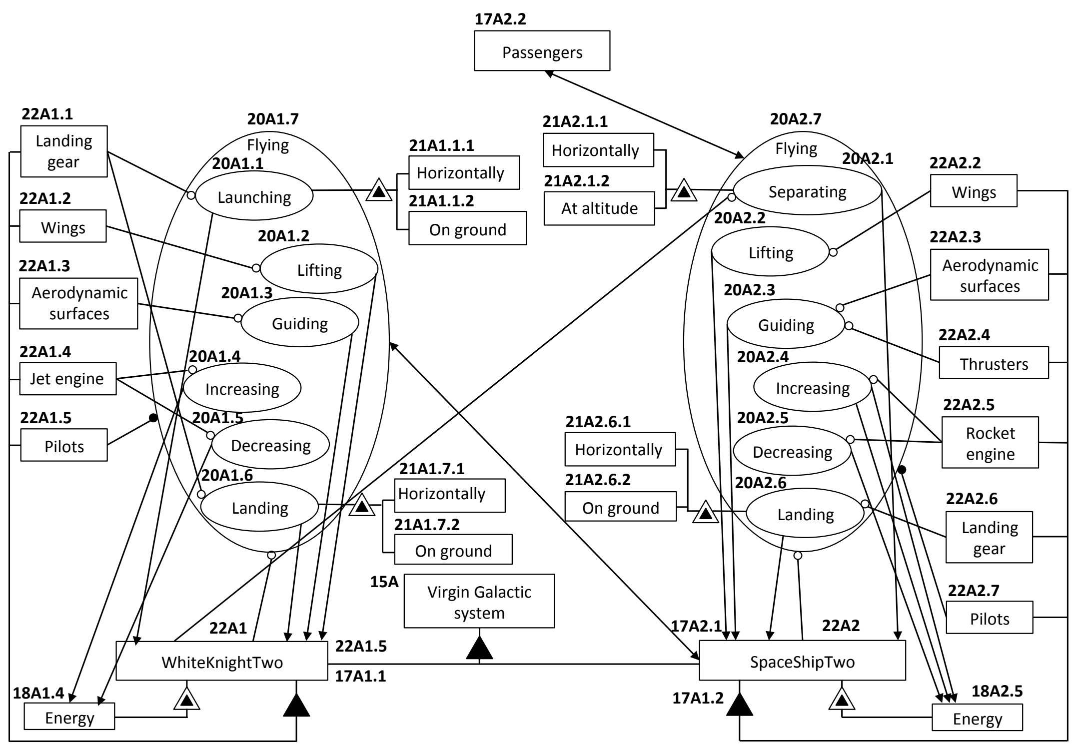


Figure 5.19: Model-based representation of the second level decomposition for Virgin Galactic concept

As such, from Figure 5.19 we may notice that sub-system WhiteKnightTwo (entry 22A1) is decomposed into five internal elements of form: landing gear (entry 22A1.1), wings (entry 22A1.2), aerodynamic surfaces (entry 22A1.3), jet engine (entry 22A1.4), and pilots (entry 22A1.5). Landing gear is used for two internal functions: launching WhiteKnightTwo and landing WhiteKnightTwo; wings are used for lifting WhiteKnightTwo; aerodynamic surfaces are used for guiding WhiteKnightTwo; jet engine is used for two functions: increasing energy of WhiteKnightTwo and decreasing energy of WhiteKnightTwo. Finally, pilots are flying SpaceShipTwo. Launching is performed horizontally on ground (entries 21A1.1.1 and 21A1.1.2, respectively). Landing is also performed horizontally on ground (entries 21A.7.1 and 21A1.7.2, correspondingly).

Sub-system SpaceShipTwo (entry 22A2) is decomposed into six internal elements of form: wings (entry 22A2.2), aerodynamic surfaces (entry 22A2.3), thrusters (entry 22A2.4), rocket engine (entry 22A2.5), landing gear (entry 22A2.6), and pilots (entry 22A2.7). Wings are used for lifting SpaceShipTwo; aerodynamic surfaces and thrusters are both used for guiding SpaceShipTwo; rocket engine is used for two internal functions: increasing energy of SpaceShipTwo and decreasing energy of SpaceShipTwo; landing gear is used for landing SpaceShipTwo; and pilots are flying passengers. Landing is performed horizontally on ground (entries 21A2.6.1 and 21A2.6.2, correspondingly).

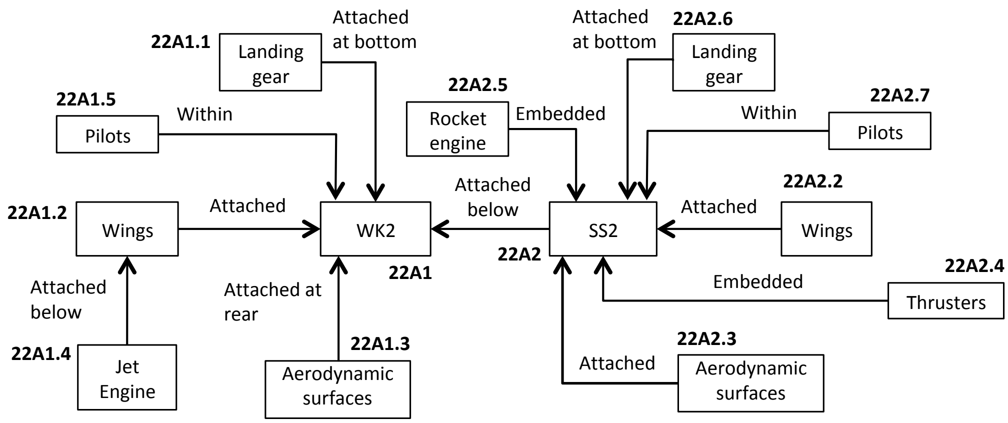
Structure and interactions are presented in Table 5.9 and Figure 5.20. In these Table and Figure formal and functional interactions among internal elements of form are explained.

Structure reflects formal arrangement of elements among each other, while interactions reflect functional arrangement. Table 5.9 is the table representation of Virgin Galactic's structure and interactions information, while Figure 5.20(a) is the graphical format of the same information about formal arrangement of internal elements of Virgin Galactic. Figure 5.20(b) represents functional interactions among internal elements of this suborbital concept.

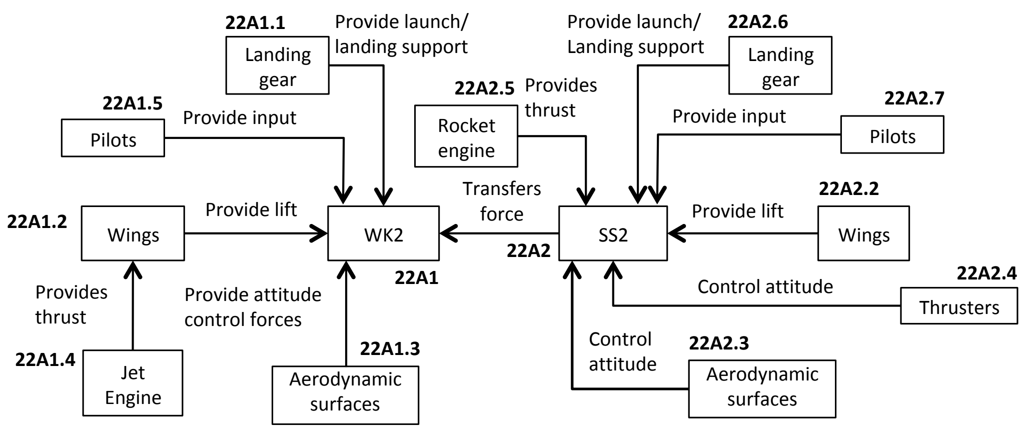
Table 5.9: Structure and Interactions for Virgin Galactic concept (table format)

24	Structure	Landing gear **Attached at bottom** WK2 Wings **Attached** WK2 Aerodynamic surfaces **Attached at rear** WK2 Jet engine **Attached below** Wings Pilots **Within** WK2	SS2 **Attached below** WK2 Wings **Attached** SS2 Aerodynamic surfaces **Attached** SS2 Thrusters **Embedded** SS2 Rocket engine **Embedded** SS2 Landing gear **Attached at bottom** SS2 Pilots **Within** SS2
25	Interactions	Landing gear **Provide launch/landing support** WK2 Wings **Provide lift** WK2 Aerodynamic surfaces **Provide attitude control forces** WK2 Jet engine **Provides thrust** Wings Pilots **Provide input** WK2	Wings **Provide lift** SS2 Aerodynamic surfaces **Control attitude** SS2 Thrusters **Control attitude** SS2 Rocket engine **Provides thrust** SS2 Landing gear **Provide launch/landing support** SS2 Pilots **Provide input** SS2

A graphical representation of structure in OPD – see Figure 5.20(a) – allows a system architect to present structure in a very strict notation. Thus, the formal arrangement of such elements of Virgin Galactic system as SpaceShipTwo and WhiteKnightTwo at the first level of decomposition, and landing gear, wings, aerodynamic surfaces, jet engine, pilots, thrusters, rocket engine at the second level of decomposition can be extracted from this diagram.



(a)



(b)

Figure 5.20: Structure (a) and interactions (b) for internal elements of form for Virgin Galactic concept at the second level of decomposition

All this information is clearly visible from the model-based diagram that contains the information about concept of interest on different levels of granularity. The ability to perform the enumeration of concept entries leads to further *formal analysis* capability.

• Integrated concept for Blue Origin concept: second level decomposition

Similarly to previous example, Blue Origin is decomposed into two sub-systems, or modules. The architectural decisions for the first module (propulsion module) of Blue Origin system are presented in Figure 5.21, while the architectural decisions for the second module (capsule) are presented in Figure 5.22.

Step		N	Parameter	Opt 1	Opt 2	Opt 3	Opt 4	Opt 5	Opt 6	Opt 7
General		1	N of Modules	1	2					
Launching		2	Type of Launch	Horizontal	Vertical					
		3	Place of Launch	Ground	Water	At altitude				
		4	Instrument of launching	Landing gear	Rocket engine					
i = 1,2										
Module 1	General (1)	5	N of pilots	0	1	2				
		6	N of passengers	0	1	2	3	4	5	6
	Flying (1)	7	Lifting	Wing	None					
		8	Guiding	Aerodynamic surface	Rocket engine	Thrusters	None			
		9	Increasing energy of module	Jet engine	Rocket engine	None				
		10	Decreasing energy of module	Aerodynamic decelerators	Rocket engine	Jet engine	Wings	None		
	Landing (1)	11	Type of landing	Horizontal	Vertical					
		12	Place of landing	Ground	Water					
		13	Instrument of landing	Landing gear	Rocket engine	Parachute				

Figure 5.21: Architectural decisions for the first module (Propulsion module) of Blue Origin concept

Step		N	Parameter	Opt 1	Opt 2	Opt 3	Opt 4	Opt 5	Opt 6	Opt 7
General		14	N of Modules	1	2					
Launching		15	Type of Launch	Horizontal	Vertical					
		16	Place of Launch	Ground	Water	At altitude				
		17	Instrument of launching	Landing gear	Rocket engine					
i = 1,2										
Module 2	General (2)	18	N of pilots	0	1	2				
		19	N of passengers	0	1	2	3	4	5	6
	Flying (2)	20	Lifting	Wing	None					
		21	Guiding	Aerodynamic surface	Rocket engine	Thrusters	None			
		22	Increasing energy of module	Jet engine	Rocket engine	None				
		23	Decreasing energy of module	Aerodynamic decelerators	Rocket engine	Jet engine	Wings	None		
	Landing (2)	24	Type of landing	Horizontal	Vertical					
		25	Place of landing	Ground	Water					
		26	Instrument of landing	Landing gear	Rocket engine	Parachute				

Figure 5.22: Architectural decisions for the second module (capsule) of Blue Origin concept

One of the hypotheses we are trying to test by the second level decomposition is that at this level of granularity we may see a core difference between alternative solutions (Virgin Galactic, Blue Origin, and XCOR). At the previous sub-section, where we explored the difference among the options at the first level decomposition, we didn't find many conceptual distinctions among Virgin Galactic and Blue Origin concepts. Later in this sub-section we will explore the conceptual similarity between these concepts at the second level decomposition.

17B1	Internal Operands (IO)	.1	.1	.1	.1	.1	.1	.2
		Propulsion module	Propulsion module	Propulsion module	Propulsion module	Propulsion module	Propulsion module	Capsule
18B1	IO value attribute	.1	.2	.3	.3	.4	.5	.6
				Energy	Energy			
19B1	IO other attribute	.1	.2	.3	.4	.5	.6	.7
20B1	Internal Processes (IP)	.1	.2	.3	.4	.5	.5	.6
		Launching	Guiding	Increasing	Decreasing	Landing	Landing	Carrying
21B1	IP attribute	.1	.2	.3	.4	.5	.5	.6
		Vertically, On ground				Vertically, On ground	Vertically, On ground	
22B1	Internal Elements of Form (IEoF)	.1	.2	.1	.1	.1	.3	.4
		Rocket engine	Aerodynamic surfaces	Rocket engine	Rocket engine	Rocket engine	Landing gear	Propulsion module
23B1	IEoF attribute	.1	.2	.3	.4	.5	.6	.7

Figure 5.23: Second level decomposition of Blue Origin's first module (propulsion module)

At the second level decomposition a conceptual similarity between competing alternatives becomes clearer. Decomposition of specific form into internal elements of form allows us to assign these elements to internal processes. Thus, we have information about internal functions that are executed by internal elements of form, and we may see that the same internal functions are performed by different internal elements of form. The information about decomposition of Blue Origin's first and second modules is presented in Figures 5.23 and 5.24, respectively. Some of the cells in Figures 5.23 and 5.24 are blank due to the principles of inheritance of attributes: the entries of these cells are contained in Table 5.7.

17B2	**Internal Operands (IO)**	**.1**	**.1**	**.1**	**.1**	**.2**
		Capsule	**Capsule**	**Capsule**	**Capsule**	**Passengers**
18B2	IO value attribute	**.1**	**.2**	**.3**	**.4**	**.5**
				Energy		
19B2	IO other attribute	**.1**	**.2**	**.4**	**.5**	**.6**
20B2	**Internal Processes (IP)**	**.1**	**.2**	**.3**	**.4**	**.5**
		Separating	**Guiding**	**Decreasing**	**Landing**	**Flying**
21B2	IP attribute	**.1**	**.2**	**.3**	**.4**	**.5**
		Vertically At altitude			Vertically On ground	
22B2	**Internal Elements of Form (IEoF)**	**.1**	**.2**	**.3**	**.4**	**.5**
		Propulsion module	**Thrusters**	**Aerodynamic decelerators**	**Parachute**	**Capsule**
23B2	IEoF attribute	**.1**	**.2**	**.3**	**.4**	**.5**

Figure 5.24: Second level decomposition of Blue Origin's second module (capsule)

The model-based concept for Blue Origin is presented in Figure 5.25 that contains the information about both decomposed forms: propulsion module (entry 22B1) and capsule (entry 22B2). From this diagram we may also see that the function of Propulsion module is "carrying capsule", while the function of capsule is "flying passengers". The entire system is launched vertically (entry 21B1.1.1) from ground (entry 21B1.1.2). The first module is later landing vertically (entry 21B1.5.1) on ground (entry 21B1.5.2). The second module lands vertically (entry 21B2.4.1) on ground (entry 21B2.4.2) as well.

The model-based representation has the same ontology and semantics for both sub-systems, thus we may notice the core information about both sub-systems on the same level of granularity.

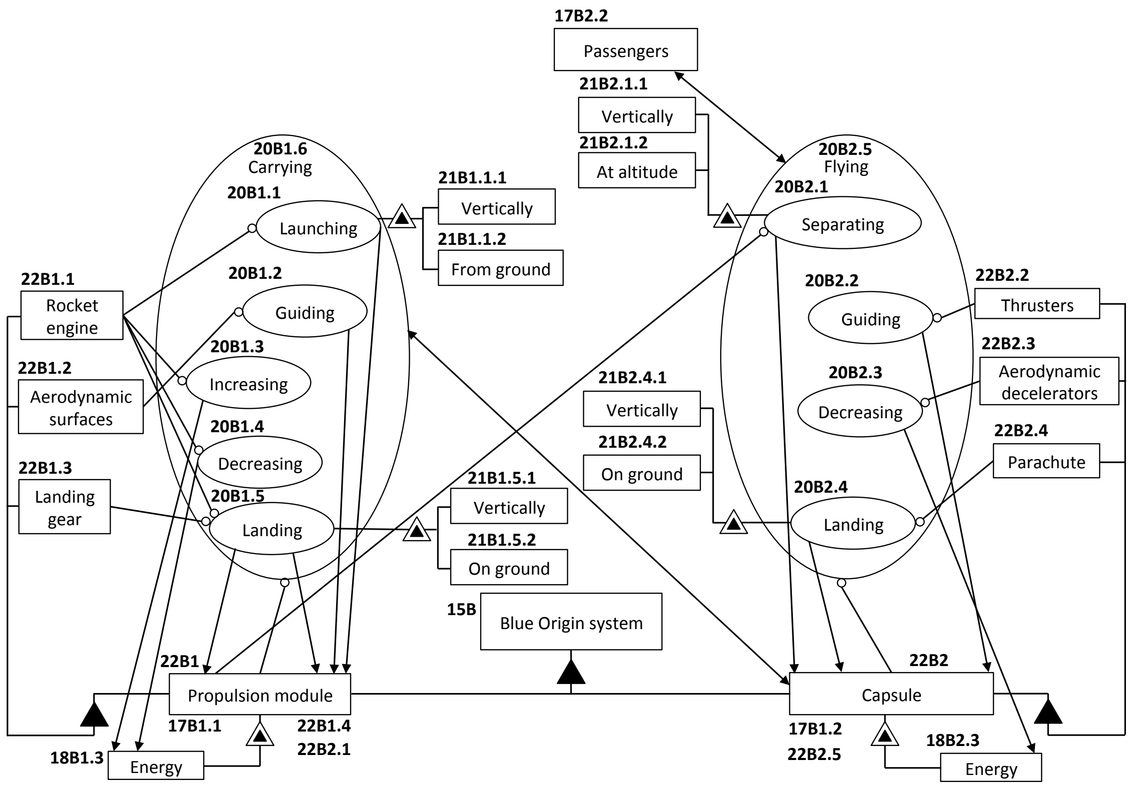


Figure 5.25: Model-based representation of second level decomposition for Blue Origin concept

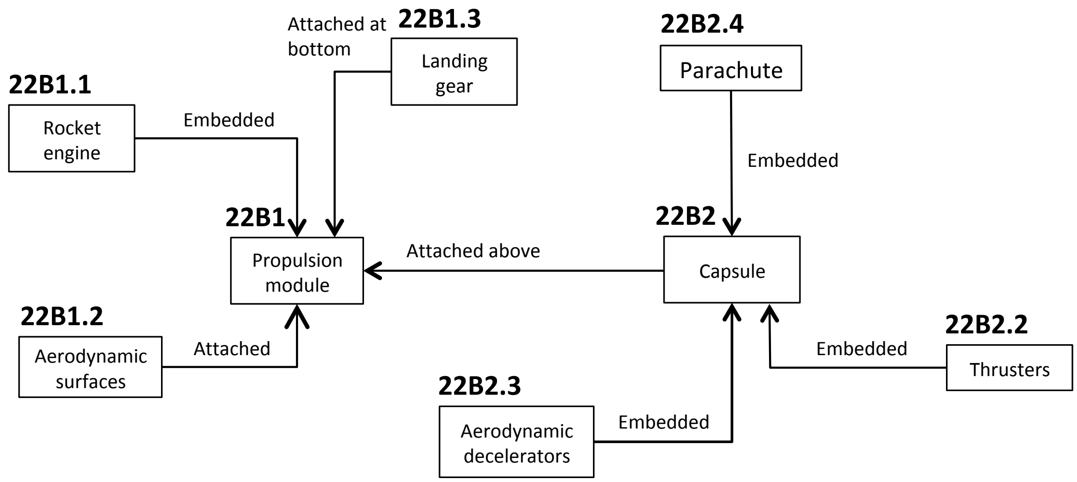
From Figure 5.25 we may see that sub-system propulsion module is decomposed into three internal elements of form: rocket engine (entry 22B1.1), aerodynamic surfaces (entry 22B1.2), and landing gear (entry 22B1.3). Rocket engine is used for multiple internal functions: it is launching propulsion module, increasing energy of propulsion module, decreasing energy of propulsion module, and landing propulsion module; aerodynamic surfaces are guiding propulsion module; landing gear is landing propulsion module. From the left-hand side of diagram we may notice that launching is performed vertically from ground (entries 21B1.1.1 and 21B1.1.2), and landing is performed vertically on ground (entries 21B1.5.1 and 21B1.5.2).

The second sub-system, a capsule, is also decomposed into three internal elements of form: thrusters (entry 22B2.2), aerodynamic decelerators (entry 22B2.3), and parachute (entry 22B2.4). Thrusters are used for guiding capsule; aerodynamic decelerators are decreasing energy of capsule; and parachute lands capsule. Landing is performed vertically on ground (entries 21B2.4.1 and 21B2.4.2).

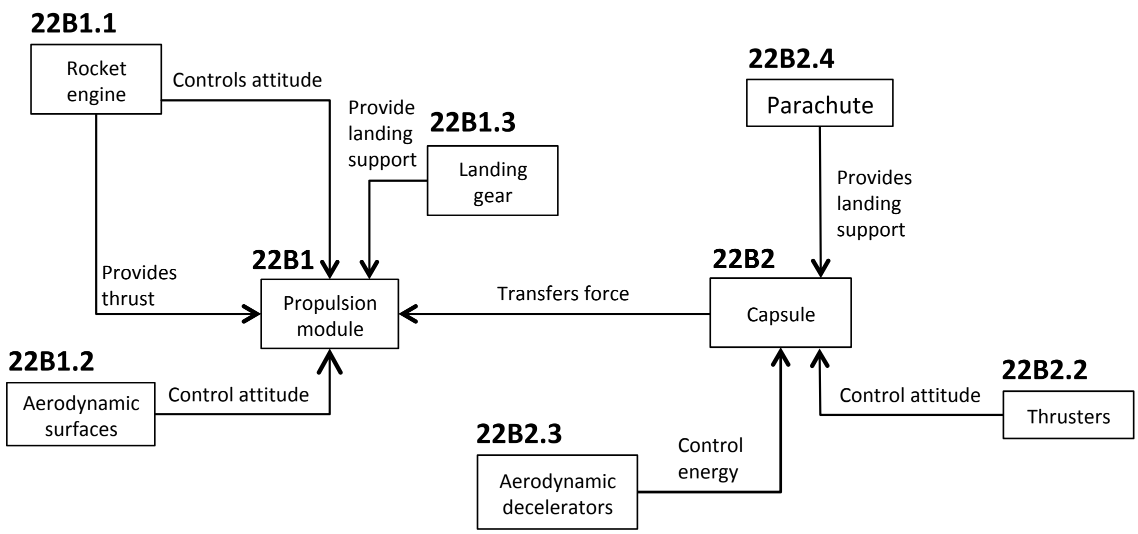
Structure and interactions are presented in Table 5.10 and Figure 5.26. The formal relationships of such elements of Blue Origin as Propulsion module and Capsule at the first level of decomposition of specific form, and rocket engine, aerodynamic surfaces, landing gear, thrusters, aerodynamic decelerators, and parachute at the second level decomposition are demonstrated in Table 5.10 (table view) and Figure 5.26 (graphical representation).

Table 5.10: Structure and Interactions for Blue Origin concept (table format)

24	Structure	Rocket engine **Embedded** Propulsion module Aerodynamic surfaces **Attached** Propulsion module Landing gear **Attached at bottom** Propulsion module	Capsule **Attached above** Propulsion module Thrusters **Embedded** Capsule Aerodynamic decelerators **Embedded** Capsule Parachute **Embedded** Capsule
25	Interactions	Rocket engine **Controls attitude** Propulsion module Rocket engine **Provides thrust** Propulsion module Aerodynamic surfaces **Control attitude** Propulsion module Landing gear **Provide landing support** Propulsion module	Capsule **Transfers force** Propulsion module Thrusters **Control attitude** Capsule Aerodynamic decelerators **Control energy** Capsule Parachute **Provides landing support** Capsule



(a)



(b)

Figure 5.26: Structure (a) and interactions (b) for internal elements of form for Blue Origin concept at the second level of decomposition

- Integrated concept for XCOR concept: second level decomposition

The architectural decisions for XCOR concept are presented in Figure 5.27 that summarize the information about the most impactful decisions that shape XCOR design.

Step		N	Parameter	Opt 1	Opt 2	Opt 3	Opt 4	Opt 5	Opt 6	Opt 7
General		1	N of Modules	1	2					
Launching		2	Type of Launch	Horizontal	Vertical					
		3	Place of Launch	Ground	Water	At altitude				
		4	Instrument of launching	Landing gear	Rocket engine					
i = 1,2										
Module 1	General (1)	5	N of pilots	0	1	2				
		6	N of passengers	0	1	2	3	4	5	6
	Flying (1)	7	Lifting	Wing	None					
		8	Guiding	Aerodynamic surface	Rocket engine	Thrusters	None			
		9	Increasing energy of module	Jet engine	Rocket engine	None				
		10	Decreasing energy of module	Aerodynamic decelerators	Rocket engine	Jet engine	Wings	None		
	Landing (1)	11	Type of landing	Horizontal	Vertical					
		12	Place of landing	Ground	Water					
		13	Instrument of landing	Landing gear	Rocket engine	Parachute				

Figure 5.27: Architectural decisions for XCOR concept

At the first level decomposition we noticed a key difference between XCOR and other concepts: XCOR doesn't decompose into sub-systems at the first level of decomposition, as it has one module. Thus, we may see the decomposition at the second level, which is presented at this sub-section.

Figure 5.28 contains the information about five internal elements of form, internal processes and internal operands on which these processes act.

17C	**Internal Operands (IO)**	**.1**	**.1**	**.1**	**.1**	**.1**	**.1**	**.2**
		XCOR	**XCOR**	**XCOR**	**XCOR**	**XCOR**	**XCOR**	**Passengers**
18C	IO value attribute	**.1**	**.2**	**.3**	**.4**	**.4**	**.5**	**.6**
					Energy	Energy		
19C	IO other attribute	**.1**	**.2**	**.3**	**.4**	**.5**	**.6**	**.7**
20C	**Internal Processes (IP)**	**.1**	**.2**	**.3**	**.4**	**.5**	**.6**	**.7**
		Launching	**Lifting**	**Guiding**	**Increasing**	**Decreasing**	**Landing**	**Flying**
21C	IP attribute	**.1**	**.2**	**.3**	**.4**	**.5**	**.6**	**.7**
		Horizontally On ground					Horizontally On ground	
22C	**Internal Elements of Form (IEoF)**	**.1**	**.2**	**.3**	**.4**	**.4**	**.1**	**.5**
		Landing gear	**Wings**	**Aerodynamic surfaces**	**Rocket engine**	**Rocket engine**	**Landing gear**	**Pilots**
23C	IEoF attribute	**.1**	**.2**	**.3**	**.4**	**.5**	**.6**	**.7**

Figure 5.28: Second level decomposition for XCOR concept

The model-based representation of the same information is demonstrated in Figure 5.29. The numbers that are assigned to internal operands, processes and forms correspond to the ones presented in Figure 5.28.

Note that some of the cells in Figure 5.28 are blank due to the principles of inheritance of attributes: the entries of these cells are contained in Table 5.8.

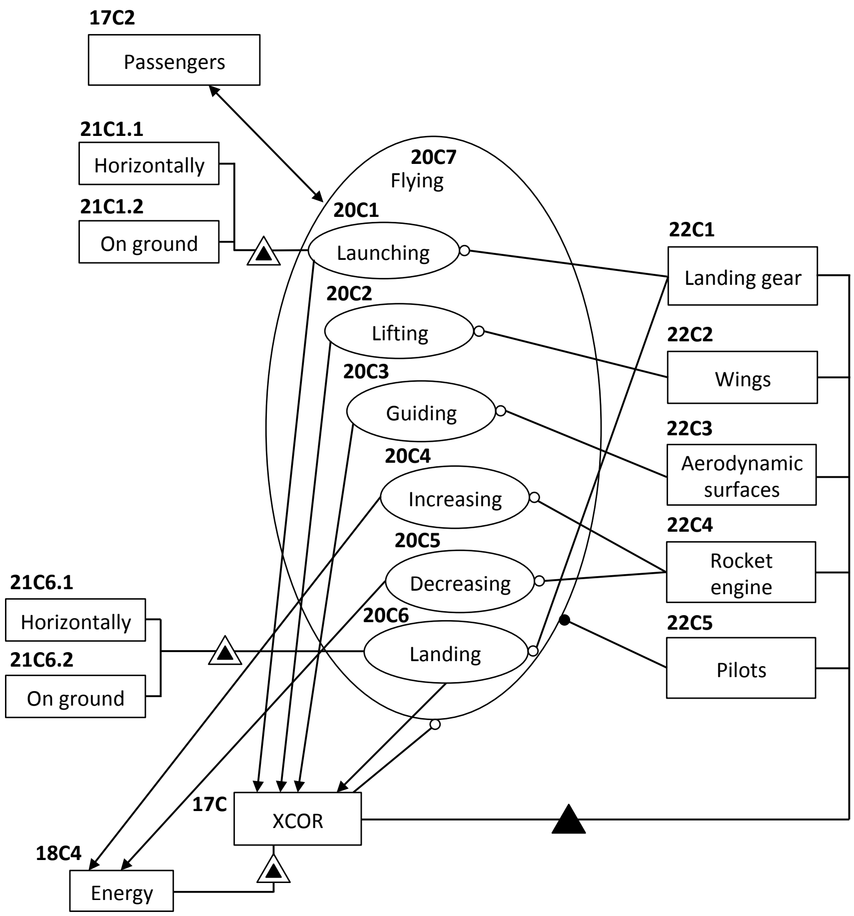


Figure 5.29: Model-based representation of XCOR concept

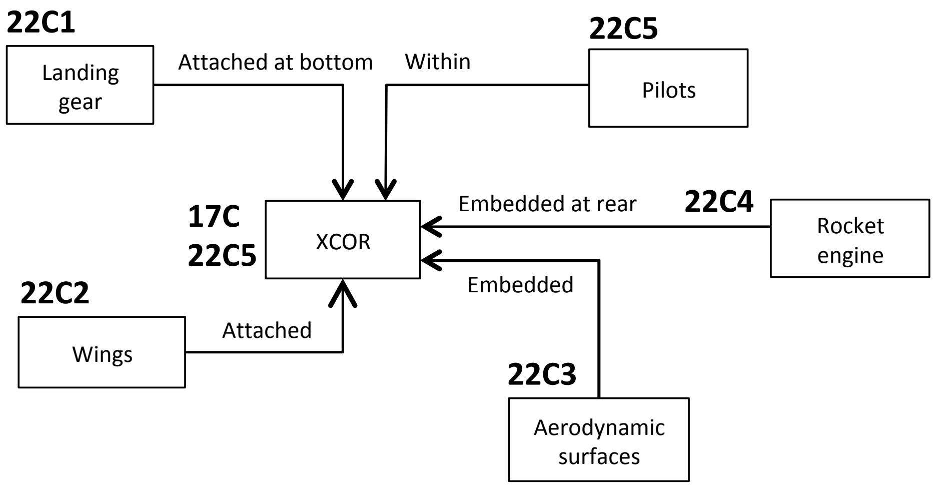
The model-based view presented in Figure 5.29 informs us that XCOR is decomposed into five sub-systems: landing gear (entry 22C1), wings (entry 22C2), aerodynamic surfaces (entry 22C3), rocket engine (entry 22C4), and pilots (entry 22C5). Landing gear is used for two internal functions: launching XCOR and landing XCOR; wings are used for lifting XCOR; aerodynamic surfaces are guiding XCOR; and rocket engine is used for two internal functions: increasing energy of XCOR and decreasing energy of XCOR. Pilots are flying passengers. This system is launched horizontally on ground (entries 21C1.1 and 21C1.2) and is landed horizontally on ground (entries 21C6.1 and 21C6.2).

Structure and interactions for XCOR concept are presented in Table 5.11 and Figure 5.30. Note that XCOR is one module system, which implies that there is one level of decomposition of specific form.

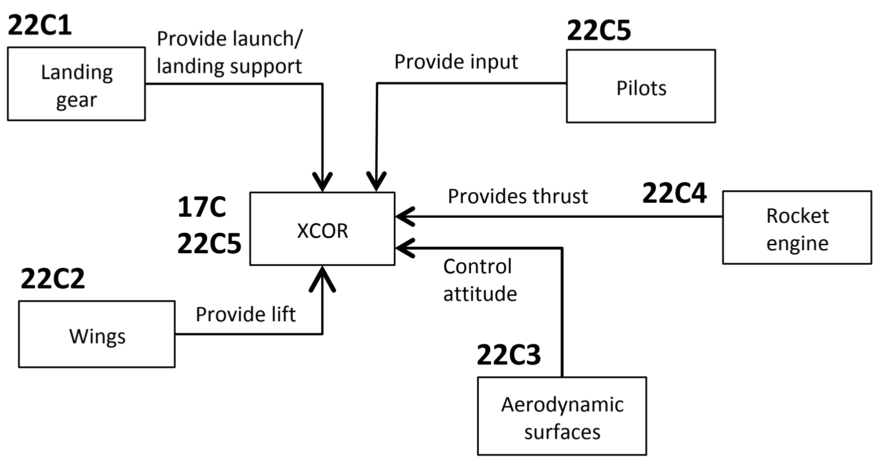
Table 5.11 is the table representation of XCOR's structure and interactions information, while Figure 5.30, is the graphical format of the same information about formal arrangement and functional interactions among internal elements of form.

Table 5.11: Structure and Interactions for XCOR concept (table format)

24	Structure	Landing gear **Attached at bottom** XCOR Wings **Attached** XCOR Aerodynamic surfaces **Embedded** into XCOR Rocket engine **Embedded at rear** XCOR Pilots **Within** XCOR
25	Interactions	Landing gear **Provide launch/landing support** XCOR Wings **Provide lift** to XCOR Aerodynamic surfaces **Control attitude** XCOR Rocket engine **Provides thrust** to XCOR Pilots **Provide input** XCOR



(a)



(b)

Figure 5.30: Structure (a) and interactions (b) for internal elements of form for XCOR concept at the second level of decomposition

5.5.9 Outcomes of the Second Level Decomposition of Alternative Concepts

In sub-section 5.5.9 we compare the pairs of concepts' modules based on their functions. For example, function of Virgin Galactic's first module (WhiteKnightTwo) is "flying SpaceShipTwo". The analog of this module for Blue Origin concept is

Propulsion module, whose function is “carrying capsule” (see Figure 5.31). Figure 5.32 is a table representation of comparison of these two modules.

Figure 5.31: The first modules of Virgin Galactic (left) and Blue Origin (right) concepts

The upper table of Figure 5.32 contains the information about the internal elements of form and internal processes that are related to WhiteKnightTwo module of Virgin Galactic. The lower table of Figure 5.32 refers to Propulsion module of Blue Origin concept. If we compare them column-by-column, we will see the differences in assigning internal element of form to internal process for each alternative. From these tables we may see that, for instance, WhiteKnightTwo has a process “launching horizontally”, while Propulsion module – “launching vertically”. We also see that in some cases there is no counterpart of some internal process, as it is, for example, for “lifting” process of WhiteKnigtTwo. This is due to the fact that Propulsion module (rocket) doesn’t have a lifting process. Overall representation of Figure 5.32 contains an important information about which exactly internal element of form is responsible for which exactly internal function.

17	WK2	WK2	WK2	WK2	WK2	WK2			SS2
20	Launching Horizontally	Lifting	Guiding	Increasing energy	Decreasing energy	Landing horizontally			Carrying
22	Landing gear	Wings	Aerodynamic surfaces	Jet engine	Jet engine	Landing gear			Pilots

17	Propulsion module		Propulsion module	Propulsion module	Propulsion module	Propulsion module	Propulsion module	Propulsion module	Capsule
20	Launching vertically		Guiding	Increasing energy	Decreasing energy	Landing vertically	Landing vertically	Landing vertically	Carrying
22	Rocket engine		Aerodynamic surfaces	Rocket engine	Rocket engine	Rocket engine	Landing gear	Rocket engine	Propulsi on module

Figure 5.32: Comparison of first modules of Virgin Galactic and Blue Origin concepts

The second module of Virgin Galactic concept is SpaceShipTwo with internal function “flying passengers”, while the second module of Blue Origin concept is Capsule whose internal function is also “flying passengers”. Both modules are presented in Figure 5.33. The comparison of the second modules of Virgin Galactic and Blue Origin concepts is demonstrated in Figure 5.34.

Figure 5.33: The second modules of Virgin Galactic (left) and Blue Origin (right) concepts

We may see a bigger number of differences between Virgin Galactic and Blue Origin at the comparison of second modules than we saw during the comparison of first modules. As such, Virgin Galactic's second module SpaceShipTwo has internal process "lifting", which does not have a counterpart for the second module of Blue Origin concept – a Capsule. Also, there is an "increasing energy" internal process for SpaceShipTwo, and no such process for Capsule.

17	SS2	SS2	SS2	SS2	SS2	SS2	SS2	Passengers
20	Separating horizontally	Lifting	Guiding	Guiding	Increasing energy	Decreasing energy	Landing horizontally	Flying
22	WK2	Wings	Aerodynamic surfaces	Thrusters	Rocket engine	Rocket engine	Landing gear	Pilots

17	Capsule		Capsule		Capsule	Capsule	Passengers
20	Separating vertically		Guiding		Decreasing energy	Landing vertically	Flying
22	Propulsion module		Thrusters		Aerodynamic decelerators	Parachute	Capsule

Figure 5.34: Comparison of second modules of Virgin Galactic and Blue Origin concepts

In Figure 5.35 the first modules of Virgin Galactic and XCOR concepts are presented. Note that XCOR has one module, but we will count it as the first module while comparing with WhiteKnightTwo, and the second module during the comparison with SpaceShipTwo.

Figure 5.35: The first modules of Virgin Galactic (left) and XCOR (right) concepts

Figure 5.36 contains the details on comparison of the first modules of Virgin Galactic and XCOR concepts. As such we may notice that internal processes are almost identical, while internal instruments of form are different for processes “increasing energy” and “decreasing energy”.

17	WK2	WK2	WK2	WK2	WK2	WK2	SS2
20	Launching Horizontally	Lifting	Guiding	Increasing energy	Decreasing energy	Landing horizontally	Carrying
22	Landing gear	Wings	Aerodynamic surfaces	Jet engine	Jet engine	Landing gear	Pilots

17	XCOR	XCOR	XCOR	XCOR	XCOR	XCOR	Passengers
20	Launching horizontally	Lifting	Guiding	Increasing energy	Decreasing energy	Landing horizontally	Flying
22	Landing gear	Wings	Aerodynamic surfaces	Rocket engine	Rocket engine	Landing gear	XCOR

Figure 5.36: Comparison of first modules of Virgin Galactic and XCOR concepts

Comparison of second modules (see Figure 5.37) is demonstrated in Figure 5.38. SpaceShipTwo has two internal elements of form that are performing internal function “guiding SpaceShipTwo” – aerodynamic surfaces and thrusters. In turn, XCOR has one internal element of form that perform such a function – aerodynamic surfaces.

Figure 5.37: The second modules of Virgin Galactic (left) and XCOR (right) concepts

17	SS2	SS2	SS2	SS2	SS2	SS2	SS2	Passengers
20	Separating horizontally	Lifting	Guiding	Guiding	Increasing energy	Decreasing energy	Landing horizontally	Flying
22	WK2	Wings	Aerodynamic surfaces	Thrusters	Rocket engine	Rocket engine	Landing gear	Pilots

17	XCOR	XCOR	XCOR	XCOR	XCOR	XCOR	Passengers
20	Launching horizontally	Lifting	Guiding	Increasing energy	Decreasing energy	Landing horizontally	Flying
22	Landing gear	Wings	Aerodynamic surfaces	Rocket engine	Rocket engine	Landing gear	XCOR

Figure 5.38: Comparison of second modules of Virgin Galactic and XCOR concepts

In Figure 5.39 the first modules for Blue Origin and XCOR concepts are presented with the same assumption as it was discussed for previous example.

Figure 5.39: The first modules of Blue Origin (left) and XCOR (right) concepts

Figure 5.40 demonstrates the comparison of both modules. We may see that such internal process as “lifting” as it appears for XCOR cannot be found for propulsion module.

17	Propulsion module		Propulsion module	Propulsion module	Propulsion module	Propulsion module	Propulsion module	Capsule
20	Launching vertically		Guiding	Increasing energy	Decreasing energy	Landing vertically	Landing vertically	Carrying
22	Rocket engine		Aerodynamic surfaces	Rocket engine	Rocket engine	Rocket engine	Rocket engine	Propulsion module

17	XCOR	XCOR	XCOR	XCOR	XCOR	XCOR	Passengers
20	Launching horizontally	Lifting	Guiding	Increasing energy	Decreasing energy	Landing horizontally	Flying
22	Landing gear	Wings	Aerodynamic surfaces	Rocket engine	Rocket engine	Landing gear	XCOR

Figure 5.40: Comparison of first modules of Blue Origin and XCOR concepts

Figure 5.41 represents the second modules for Blue Origin and XCOR concepts.

Figure 5.41: The second modules of Blue Origin (left) and XCOR (right) concepts

From Figure 5.42 we see that such internal processes as “lifting” and “increasing energy” as they appear for XCOR do not have a counterpart for capsule module of Blue Origin concepts.

17	Capsule		Capsule		Capsule	Capsule	Passengers
20	Separating vertically		Guiding		Decreasing energy	Landing vertically	Flying
22	Propulsion module		Thrusters		Aerodynamic decelerators	Parachute	Capsule

17	XCOR	XCOR	XCOR	XCOR	XCOR	XCOR	Passengers
20	Launching horizontally	Lifting	Guiding	Increasing energy	Decreasing energy	Landing horizontally	Flying
22	Landing gear	Wings	Aerodynamic surfaces	Rocket engine	Rocket engine	Landing gear	XCOR

Figure 5.42: Comparison of second modules of Blue Origin and XCOR concepts

5.5.10 Quantifying a Conceptual Similarity Through the DSM-Based Approach

In our work (Menshenin and Crawley 2018a) we proposed a new way to capture both types of relationships – decomposition and specialization – in DSM-based methods. These proposals can be effectively used for conceptual design phase of suborbital human spaceflight missions. In Figure 5.43 these types of relationships are captured in OPM notation – black triangle for decomposition, and unfilled triangle for specialization relationship.

Figure 5.43 is the full DMM matrix that contains a core design information about 3 alternatives (Virgin Galactic, Blue Origin, and XCOR concepts) on different levels of granularity. In particular this matrix facilitates a conceptual similarity assessment between competing alternatives (DSM matrix, left hand side of Figure 5.43). The role of DSM part in Figure 5.43 is the following. Comparing the different modules, for example, WhiteKnightTwo and SpaceShipTwo, the system architect could highlight the identical internal elements of form that are used by both concepts – wings, aerodynamic surfaces, landing gear, and pilots (indicated by the "·" symbol at the intersection of corresponding cells in Figure 5.43). For example, at the intersection of first column (WK2) and third row (Propulsion module) we may see that these two modules have 2 common internal elements of form: aerodynamic surfaces and landing gear.

This matrix also has the information about the integrated concept (DMM matrix, right hand side of the Figure 5.43). The DMM part of Figure 5.43 has two important meanings. First, it indicates which internal element of form is used for which internal process – for instance, landing gear of WhiteKnightTwo is used for launching and landing processes (indicated by the "V" symbol at the intersection of corresponding cells in Figure 5.43). Secondly, the DMM part of Figure 5.43 has cells highlighted by colors – either yellow or red. The yellow color implies that in two modules under comparison – for example, WhiteKnightTwo of Virgin Galactic and Propulsion module of Blue Origin – the internal elements perform the different processes ("carrying" and "flying", respectively).

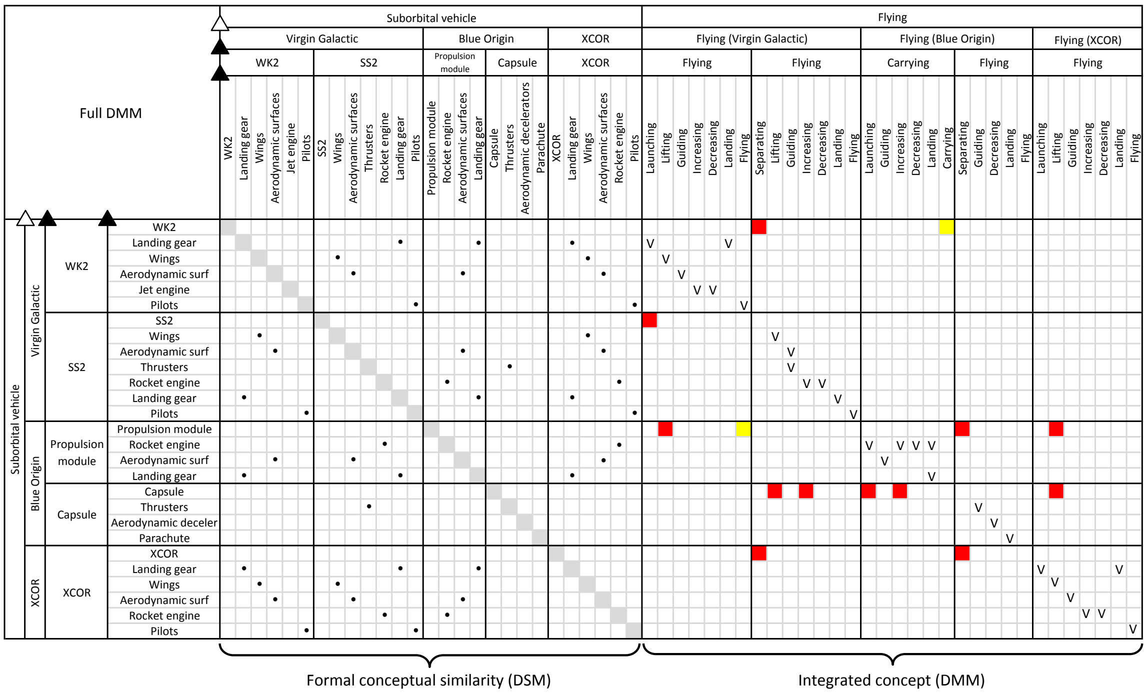


Figure 5.43: Full DMM matrix with a core conceptual design information

Red color means that in two concepts under comparison the only one has some specific internal process, while the second concept doesn't have it. The example is that the SpaceShipTwo has the internal process "lifting", while the Capsule does not have it.

Another utility of matrix demonstrated in Figure 5.43 is that it quantifies elements' criticality. For example, for Blue Origin's propulsion module it is shown that the rocket engine participates in the largest number of connections with internal processes. Thus, this sub-system has a significant importance for the concept.

Structure and interactions for the first alternative, Virgin Galactic, is demonstrated in Figure 5.44. From this Figure we see that landing gear is attached at bottom of WhiteKnightTwo (structure), or that the jet engine provides thrust to wings (interactions). At the bottom left part of Figure 5.44 we may notice the information about formal conceptual similarity of the two sub-systems – WhiteKnightTwo and SpaceShipTwo. For example, both of them have wings, aerodynamic surfaces, landing gear, and pilots as the instruments.

Structure and Interactions for Virgin Galactic project		22A1	22A1.1	22A1.2	22A1.3	22A1.4	22A1.5	22A2	22A2.2	22A2.3	22A2.4	22A2.5	22A2.6	22A2.7
		WK2	Landing gear	Wings	Aerodynamic surfaces	Jet engine	Pilots	SS2	Wings	Aerodynamic surfaces	Thrusters	Rocket engine	Landing gear	Pilots
22A1	WK2		Attached at bottom Provide launch/ landing support	Attached Provide lift	Attached at rear Provide attitude control forces		Within Provide input	Attached below Transfers forces						
22A1.1	Landing gear												•	
22A1.2	Wings					Attached below Provides thrust			•					
22A1.3	Aerodynamic surfaces									•				
22A1.4	Jet engine													
22A1.5	Pilots													•
22A2	SS2								Attached Provide lift	Attached Control attitude	Embedded Control attitude	Embedded Provides thrust	Attached at bottom Provide launch/ landing support	Within Provide input
22A2.2	Wings			•										
22A2.3	Aerodynamic surfaces				•									
22A2.4	Thrusters													
22A2.5	Rocket engine													
22A2.6	Landing gear		•											
22A2.7	Pilots						•							

Figure 5.44: Structure, interactions, and conceptual similarity for Virgin Galactic concept

The same idea is present at Figure 5.45 for Blue Origin concept. As opposed to previous concept, the Blue Origin's sub-systems are completely different: they do not share any identical elements of form. The information about structure and interactions is contained at the intersections of corresponding instruments.

Structure and Interactions for Blue Origin project		22B1 Propulsion module	22B1.1 Rocket engine	22B1.2 Aerodynamic surfaces	22B1.3 Landing gear	22B2 Capsule	22B2.2 Thrusters	22B2.3 Aerodynamic decelerators	22B2.4 Parachute
22B1	Propulsion module		Embedded Controls attitude; Provides thrust	Attached Control attitude	Attached at bottom Provide landing support	Attached above Transfers force			
22B1.1	Rocket engine								
22B1.2	Aerodynamic surfaces								
22B1.3	Landing gear								
22B2	Capsule						Embedded Control attitude	Embedded Control energy	Embedded Provide landing support
22B2.2	Thrusters								
22B2.3	Aerodynamic decelerators								
22B2.4	Parachute								

Figure 5.45: Structure, interactions, and conceptual similarity for Blue Origin concept

In Figure 5.46 we present structure and interactions for XCOR concept. We may notice that wings are attached to XCOR (structure), while rocket engine provides thrust to XCOR (interactions).

Structure and Interactions for XCOR project		17C XCOR	22C1 Landing gear	22C2 Wings	22C3 Aerodynamic surfaces	22C4 Rocket engine	22C5 Pilots
17C	XCOR		Attached at bottom Provide launch/landing support	Attached Provide lift	Embedded Control attitude	Embedded at rear Provides thrust	Within Provide input
22C1	Landing gear						
22C2	Wings						
22C3	Aerodynamic surfaces						
22C4	Rocket engine						
22C5	Pilots						

Figure 5.46: Structure, interactions, and conceptual similarity for XCOR concept

Thus, DSM representations, combined with our methods to include the specialization and decomposition relationships are powerful tools to *encode* the conceptual design information and to *measure* a conceptual similarity between competing alternatives, as well as among sub-systems of specific concept.

5.5.11 Proposition V: Concept of Operations

The fifth and last proposition of model-based *system concept representation framework* is the concept of operations. This proposition includes ConOps itself (entry 26), operator (entry 27), and context (entry 28).

As it was discussed in section 3.7.2, the whole product system contains the information about system of interest and context, which can be represented by accompanying systems. System boundary separates system we are developing from other systems that should be taken into account.

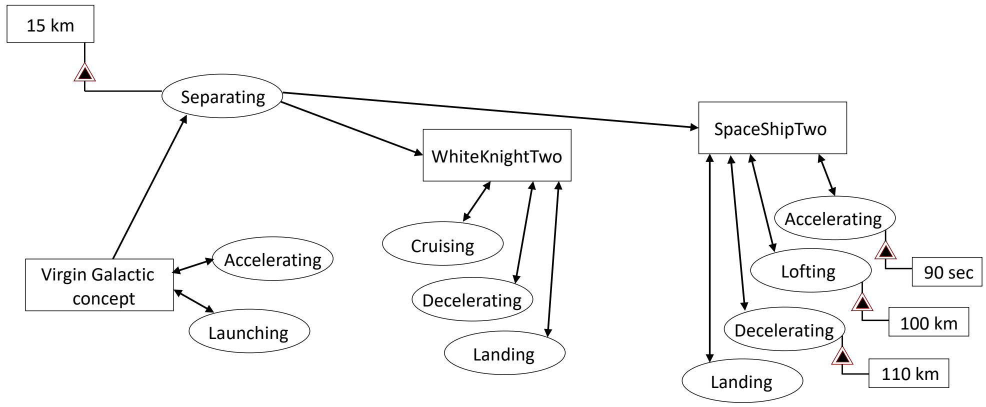
ConOps for Virgin Galactic concept is presented in Figure 5.47(a) and Figure 5.47(b). The latter is the Object-Process Diagram representation of ConOps, while the former is the conventional demonstration of operations. Context is demonstrated on Figure 5.47(c). From it we see that inside the system boundary we have the Virgin Galactic system itself, while accompanying systems are communication system, spaceport Mojave, and customer support.

Figure 5.47 demonstrates the sequence of activities that lead to the delivery of the primary function.

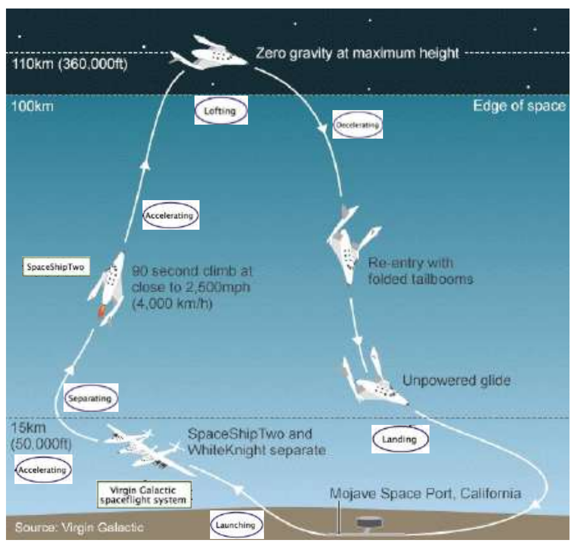
Having these two conceptual representations in hands the system architect should be able to understand how the system is intended to operate. Context – Figure 5.47(c) – under which the system operates, is supporting the system architect with identification of accompanying systems.

Thus, based on concept framework, system architect is able to capture the following sequence of activities for concept of operations of Virgin Galactic concept:

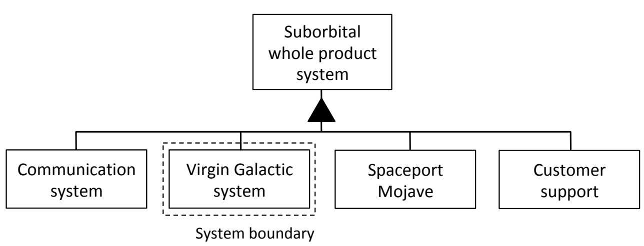
- It is launched from Spaceport Mojave, which is one of the accompanying systems (the process is “launching”, and the specific form is “Virgin Galactic concept” in diagram of Figure 5.47(a));
- After launching the system is “accelerating” up until the altitude of 15 km, on which “SpaceShipTwo” separates from “WhiteKnightTwo”;
- After separation the “SpaceShipTwo” is “accelerating” during 90 seconds reaching a velocity of 4000 km/h;
- After acceleration during 90 seconds and at about altitude of 100 km “SpaceShipTwo” is “lofting”, reaching the altitude of 110 km, and delivering the primary function, mentioned in the solution-neutral environment of concept framework: entertaining passengers;
- After several minutes of weightlessness, “SpaceShipTwo” is “decelerating” to re-enter the atmosphere;
- And finally, “SpaceShipTwo” is “landing” by means of unpowered glide in Spaceport Mojave.



(a)



(b)



(c)

Figure 5.47: ConOps in OPD view (a), ConOps in a scheme (b), and Context (c) for Virgin Galactic concept

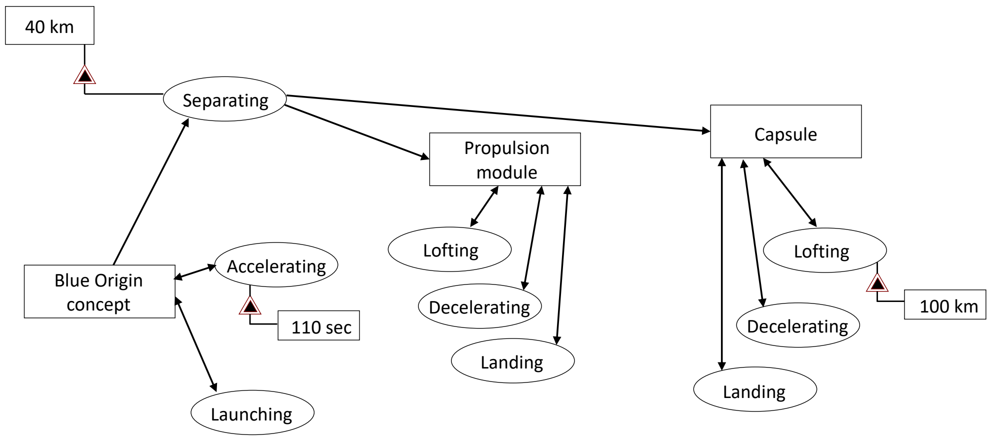
In Figures 5.48(a) and 5.48(b) ConOps for Blue Origin concept is presented (in OPD view and conventional diagram), and Figure 5.48(c) contains the information about the whole product system for Blue Origin concept.

Figure 5.48 demonstrates the representations of concept of operations for Blue Origin concept. By analogy with example in previous sub-section, this information reveals the sequence of activities that leads to the delivery of primary function of system.

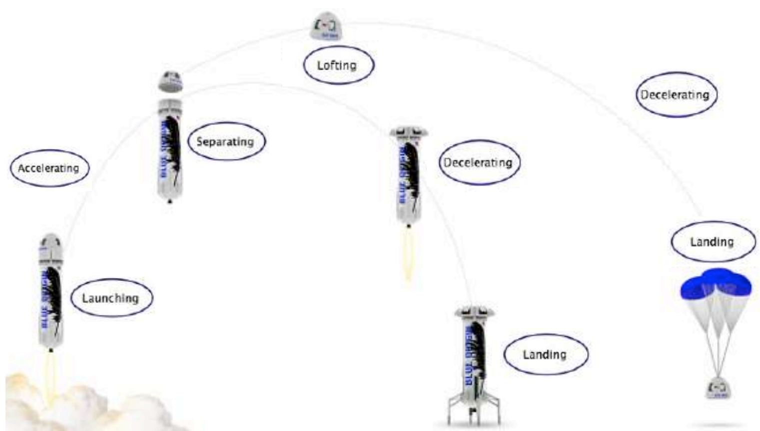
In case of New Shepard concept the Corn Ranch spaceport is chosen for “launching”, which is shown in Figure 5.48(c).

Figure 5.48(a) and 5.48(b) capture the following sequence of activities for concept of operations of Blue Origin concept:

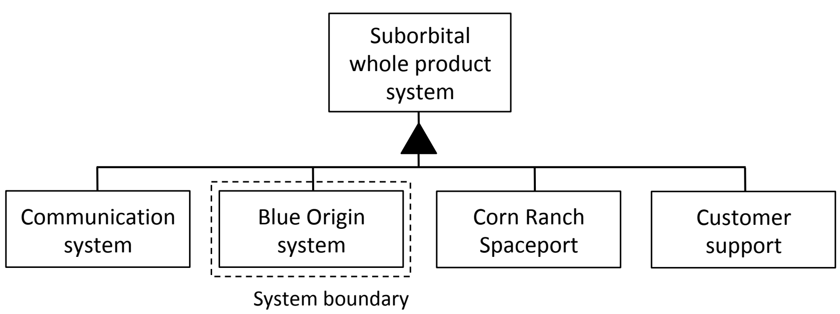
- It is launched from Corn Ranch spaceport (the process is “launching”, and specific form is “Blue Origin concept”);
- After launch the system is “accelerating” during 110 seconds, which is attribute of the process;
- At the altitude of 40 km “capsule” separates from “propulsion module”;
- After separation “capsule” is reaching the altitude of 100 km and is “lofting”, by that delivering the primary function, mentioned in solution-neutral environment of concept framework: entertaining passengers;
- After several minutes of weightlessness “Capsule” is “decelerating”;
- And finally, “capsule” is “landing” on ground.



(a)



(b)



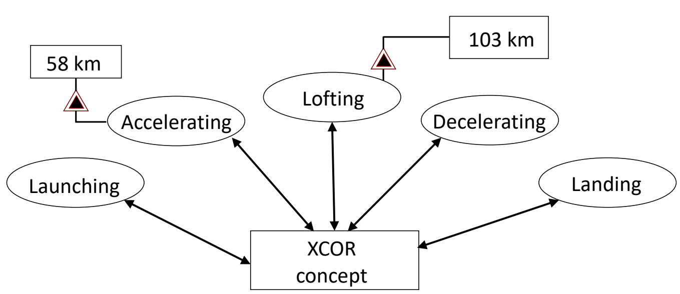
(c)

Figure 5.48: ConOps in OPD view (a), ConOps in a scheme (b), and Context (c) for Blue Origin concept

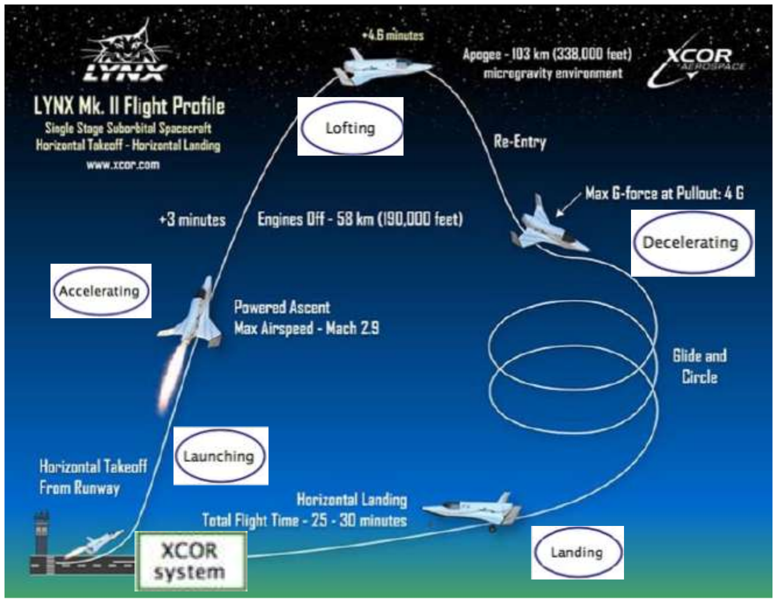
Finally, we present the information about ConOps and Context for XCOR concept – see Figure 5.49(a), 5.49(b), and 5.49(c). These figures describe the concept of operations and context of XCOR concept. These three figures reveal the representations of sequence of activities for XCOR concept.

Spaceport Mojave was chosen for "launching" of XCOR concept that is shown in Figure 5.49(c). Concept of operations captures the following main activities, leading to delivery of primary function:

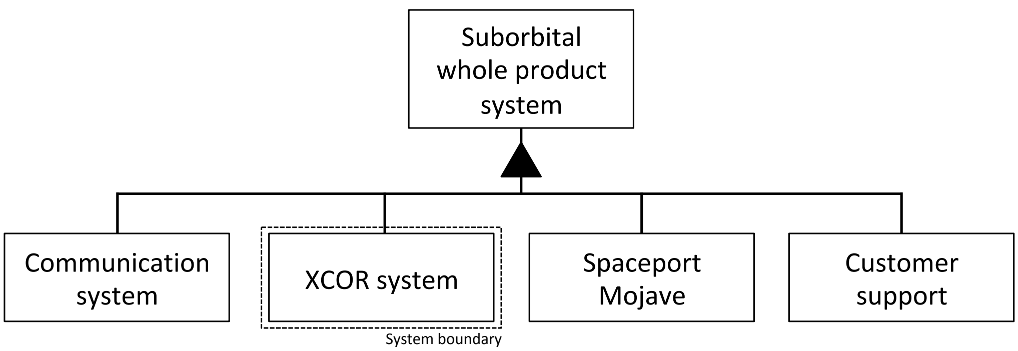
- XCOR is launched from Mojave spaceport (the process is "launching", and the specific form is "XCOR concept");
- After launch the concept is "accelerating" until the altitude of 58 km;
- At the altitude of 58 km the engines are switched off, the system is reaching the altitude of 103 km and is "lofting", by that delivering the primary function, mentioned in the solution-neutral environment of concept framework: entertaining passengers;
- After several minutes of weightlessness "XCOR" concept is "decelerating";
- And finally, "XCOR" concept is "landing" on ground.



(a)



(b)



(c)

Figure 5.49: ConOps in OPD view (a), ConOps in a scheme (b), and Context (c) for XCOR concept

Thus, we see how precise and useful the information contained in the model-based concepts for all three alternative concepts of suborbital missions that are considered in this case study.

5.6 Conceptual Similarity Between Alternative Concepts

Conceptual similarity can also be measured based on architectural decisions, demonstrated in Table 5.1. Figures 5.15 and 5.16, 5.21 and 5.22, 5.27 are summarized architectural decisions for Virgin Galactic, Blue Origin, and XCOR, respectively.

Using these Figures, the system architect and project team can precisely define the decomposed instruments and the functions that these instruments perform. This information is required for system integration and manufacturing purposes. There are also the details that are necessary not only for technical side, but also for economical one. For instance, a different number of passenger seats leads to a various business plans of the companies. On the one hand, 6 seats as it is in case of Virgin Galactic concept versus 1 seat as it is in case of XCOR concept generate more revenue for the Richard Branson's company. On the other hand, if the customer wants to get a personal experience, he or she might prefer XCOR concept, even for a higher ticket price.

There can also be a psychological factor. In our days the number of air travelers reach dozens of millions per year. A conventional airplane has a horizontal takeoff horizontal landing mode of operation. So, the majority of potential suborbital spaceflight customers would feel themselves in a well-known environment in case of HTHL concept of suborbital vehicle. If such a vehicle has a vertical takeoff vertical landing, this is more like experience of being an astronaut.

As systems engineering practitioners, we should take all these factors – not only technical ones – into account while designing suborbital spaceflight systems.

It should be noted that although the term “concept distance” doesn’t appear in a wide spread in scientific literature, there is a common idea of “architecture distance”, which is used by system engineers and system architects. We will first make a brief introduction to idea of “architecture distance”. After that we will demonstrate, which parameters have been chosen for measuring the “concept distance” for suborbital human spaceflight missions.

Smaling and de Weck proposed the *delta DSM* for technology infusion assessment (Smaling and de Weck 2007). According to authors, “the degree of invasiveness of different system architecture is related to amount of design change required to accommodate the new technology”. Thus, in order to define the difference between two architectures, the authors proposed to use a component-based change Design-Structure Matrix (DSM).

Nakamura and Basili (2005) proposed an approach for architectural change in software architecture. According to this approach, “two endpoints of a major change are taken as reference points, and intermediate connectivity changes are examined relative to the endpoints”. The authors proposed to use a graph to define a distance measure between software structures. This distance measure is used to define a metric, which “models the architecture change as a transition between two endpoints”.

For the purpose of identification alternative concepts for suborbital human spaceflight mission the architectural decisions, shown in Table 5.1, have been analyzed and the following of them were chosen with corresponding parameters:

Parameter	Virgin Galactic	Blue Origin	XCOR
Number of modules	2	2	1
Type of launch	Horizontal	Vertical	Horizontal
Place of launch	Ground	Ground	Ground
Number of pilots	4	0	1
Number of passengers	6	6	1
Type of landing	Horizontal	Vertical	Horizontal
Place of landing	Ground	Ground	Ground

There are number of parameters that lead to conceptual difference between Virgin Galactic, Blue Origin and XCOR concepts. Virgin Galactic and Blue Origin comprise of two modules, while XCOR has one module. Meanwhile, Virgin Galactic and XCOR have horizontal takeoff, horizontal landing (HTHL) concept of operations, while Blue Origin is a vertical takeoff, vertical landing (VTVL) one (see Figure 5.4).

It should be clearly defined what we imply by “conceptual distance”. Since the concept definition deals with mapping function to form, the conceptual distance is close to this idea. A conceptual distance is a difference in an instrument that performs the same function, or in a parameter that represents the architectural decision. In other words, this is a reflection of the fact that different internal operands, corresponding to internal functions, are performing the same high-level function.

5.7 Interconnections Between Solution Diagrams and Architectural Decisions

The power of solution diagrams, presented in Figures 5.19 (Virgin Galactic), 5.25 (Blue Origin), and 5.29 (XCOR) is that they contain the architectural decisions, shown in Figures 5.15 and 5.16 (Virgin Galactic), 5.21 and 5.22 (Blue Origin), and 5.27 (XCOR).

In order to demonstrate the interwoven character of these representations Figures 5.50 (Virgin Galactic), 5.51 (Blue Origin), and 5.52 (XCOR) are shown in this Chapter.

Consider Figures 5.50-5.52 dedicated to connections between solution diagram and architectural decisions for Virgin Galactic, Blue Origin, and XCOR concepts. The system architect might see that almost all the decisions, mentioned in Figures 5.15-16 are shown in Figure 5.50.

Moreover, the allocation of these decisions in the diagrams follows some pattern. The decision on number of modules is located in lower part of the diagram – in the place where the system architect might find the aggregation/decomposition symbol in OPM notation. Such internal processes as “lifting”, “guiding”, “increasing energy of module”, “decreasing energy of module” that are all aggregated into the process “flying”, are present in both sides of the Figure 5.50. One set of internal processes is aggregated into the process “flying” (WhiteKnightTwo), while the other one is aggregated into the process “flying” (SpaceShipTwo). In turn, the internal elements of form are assigned to the internal processes and present at left hand side and right-hand side of Figure, respectively. The respective attributes that inform us about the place and type of launch or landing are demonstrated closer to the center of Figure. Finally, the architectural decision related to the operand is mentioned at the top of Figure.

This pattern on architectural decisions is relevant to all three projects that are under exploration in this Chapter.

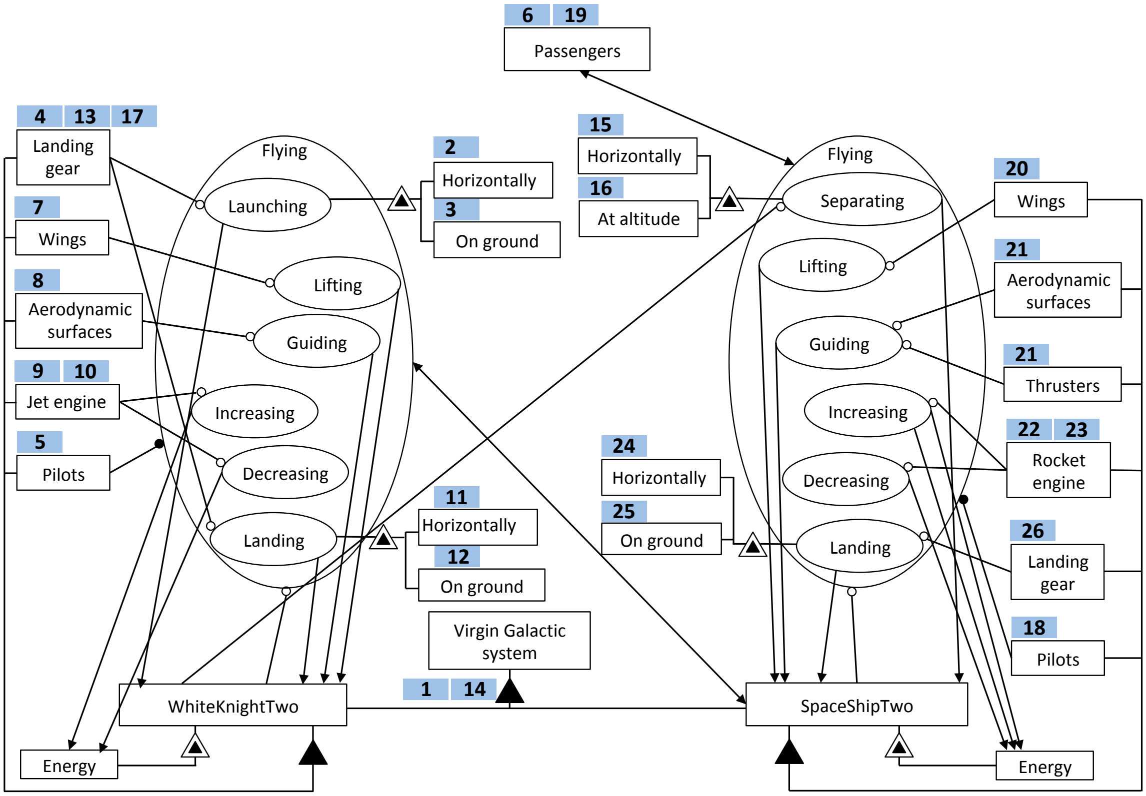


Figure 5.50: Architectural decisions in solution diagram for Virgin Galactic

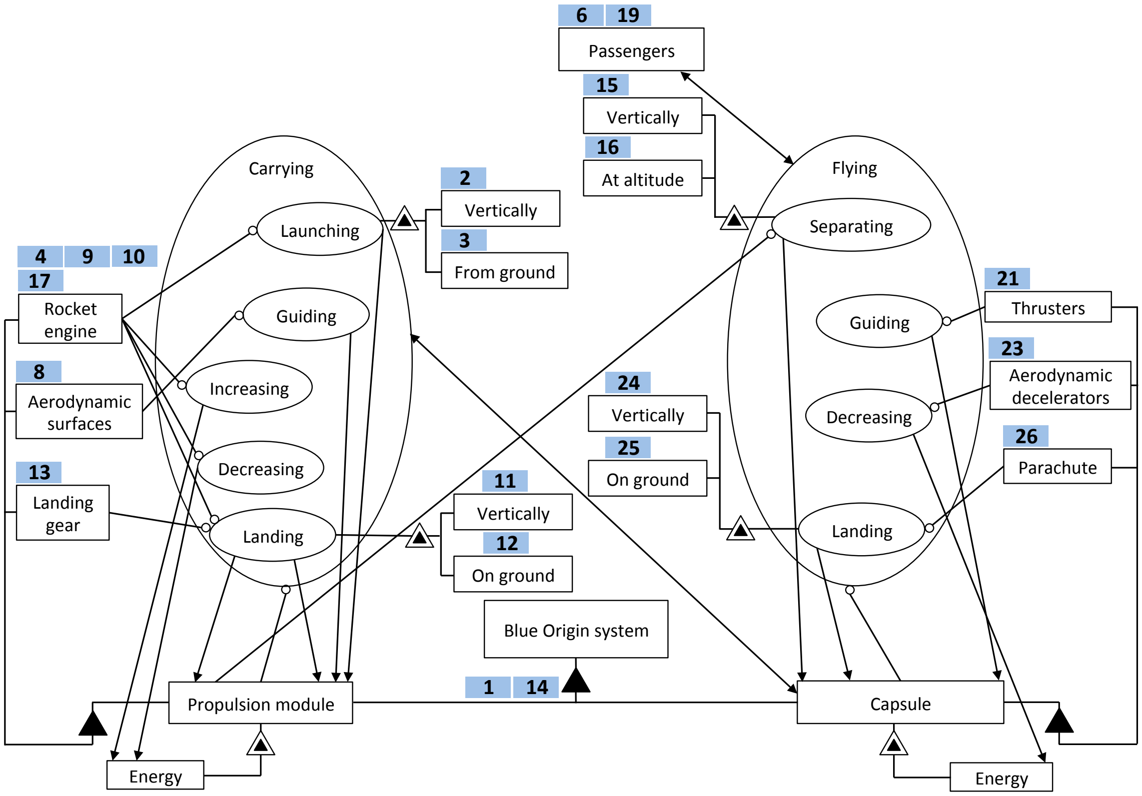


Figure 5.51: Architectural decisions in solution diagram for Blue Origin

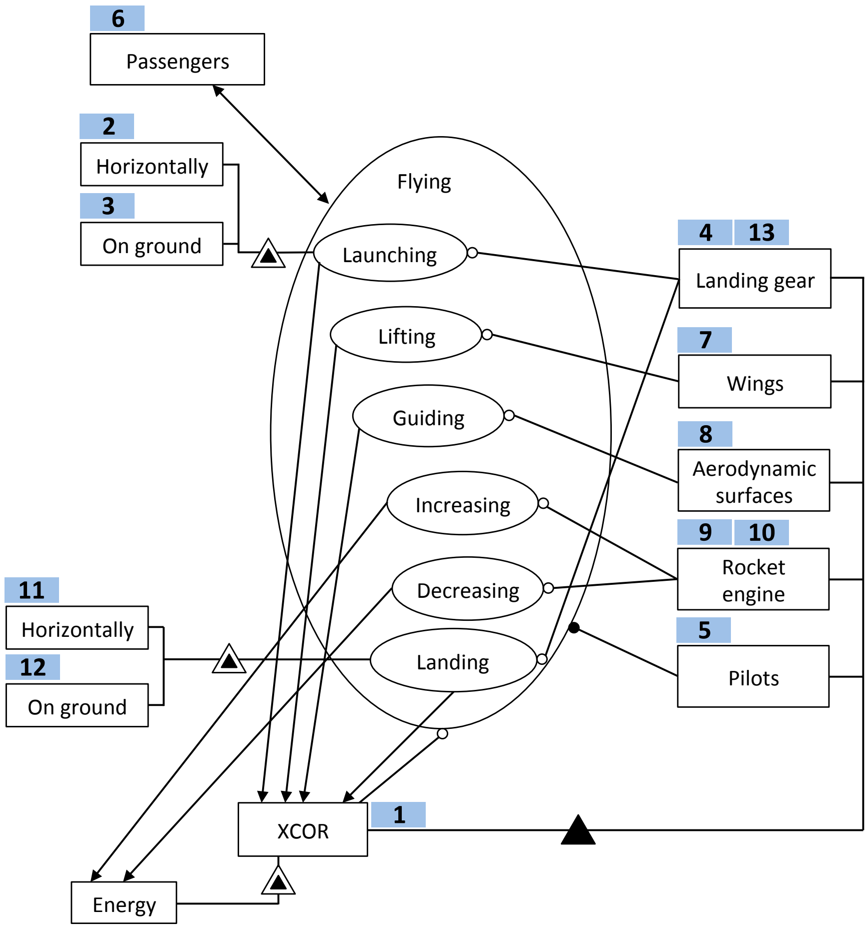


Figure 5.52: Architectural decisions in solution diagram for XCOR

These Figures demonstrate the importance of solution diagram. It contains not only the information about from-function allocation, but also the key architectural decisions that guide system design.

5.8 Suborbital Human Spaceflight Case Study Summary and Conclusions

In Chapter 5 we demonstrated the utility of the proposed model-based *system concept representation framework* for conceptual design of suborbital human spaceflight concepts, such as Virgin Galactic, Blue Origin, and XCOR. In particular:

- The architectural decisions for suborbital human spaceflight missions have been developed and presented in Table 5.1. These architectural decisions are applicable to any system, which is intended to operate in suborbital human spaceflight environment. In Chapter 5 we applied these architectural decisions to three above-mentioned alternative concepts. We presented the chosen architectural decisions for each of these projects in Figures 5.15-16, 5.21-22, 5.27, respectively;
- We *encoded* a conceptual information about all 3 alternative concepts into proposed model-based frameworks comprised of 28 entries. These entries are spread among different parts of framework: stakeholders and their needs; solution-neutral environment (problem statement); solution-specific environment (solution statement); integrated concepts at different levels of granularity, including structure and interactions; concept of operations, and context;
- We demonstrated how the proposed approach supports a *formal analysis*, such as a conceptual similarity assessment. In particular, we have shown that the DSM-based approach quantifies the number of internal elements of form, or the number of internal processes that are the same among two alternative concepts under comparison;
- The interconnections between concept framework's solution diagrams and architectural decisions are demonstrated, which allows to consider the methodology

as a united framework to develop any suborbital human spaceflight concept.

Thus, in Chapter 5 we demonstrated the utility of proposed model-based *system concept representation framework* for suborbital human spaceflight missions.

Chapter 6. Case Study II: Space Communication Mission

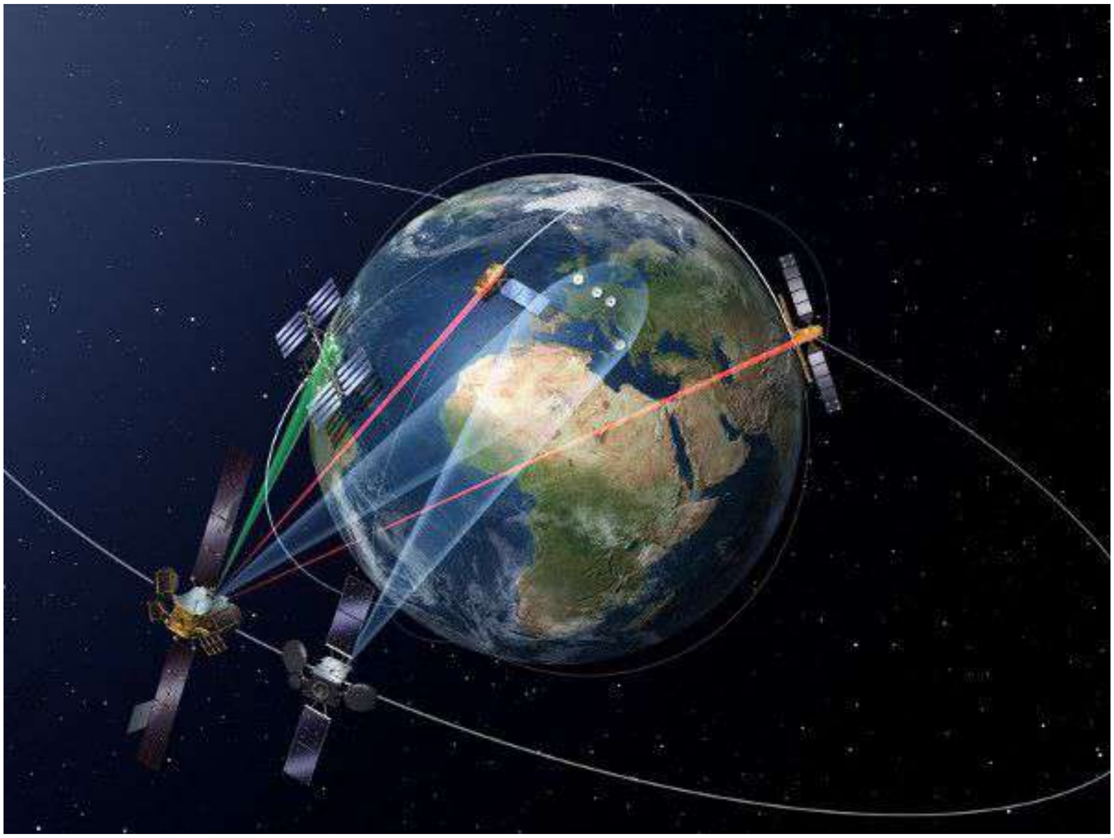


Image: esa.int

6.1 Introduction

The objective of Chapter 6 is to demonstrate the utility of proposed framework for the space communication (spacecom) systems. In this Chapter we will explore the systems aiming at relaying information to and from non-geostationary satellites, spacecraft, other vehicles, and fixed Earth stations that otherwise are not able to permanently transmit/receive data. These systems are represented by the Tracking and Data Relay Satellite (TDRS) system, European Data Relay System (EDRS), as well as the Near Earth Network (NEN) system.

Applying the model-based *system concept representation framework* to space communication systems we would demonstrate a practical utility of concepts' *encoding* and how this process supports a *formal analysis*, such as concept similarity assessment.

In Chapter 6 we focus on model-based representations of three conceptually different spacecom systems, namely, TDRS system, EDRS system, and NEN system. These concepts are *encoded* in a systematic way – starting from stakeholders and their needs and ending with the concept of operations of the systems. These concept models are connected with architectural decisions that are developed by systems engineer. The model-based concept frameworks for space communication systems demonstrated in Chapter 6 are built based on the methodology, explained in Chapter 3. The Chapter 6 corresponds to the Descriptive Study II, according to the DRM framework that is used in this Thesis.

In Chapter 6 we also propose a new method to document the information about specialization and decomposition relationships through the DSM-based approaches (Menshenin and Crawley 2018a). This method is utilized in current Chapter to demonstrate the *formal analysis*, such as conceptual similarity assessment.

The case study presented in Chapter 6 was developed in the System Architecture Lab at MIT in 2016. The study involved colleagues from the Massachusetts Institute of Technology.

The remainder of Chapter 6 is organized as follows. In Sect. 6.2, a historical background of space communication systems is provided. In Sect. 6.3 the motivation and context for model-based concept framework development for spacecom missions are discussed. In Sect. 6.4 the specific objectives for this case study are mentioned. Application of the model-based concept framework to three alternative space communication projects is demonstrated in Sect. 6.5. The formal analysis applied for alternative concepts is discussed in Sect. 6.6. The interconnections between solution diagrams and architectural decisions are explained in Sect. 6.7 Finally, in Sect. 6.8 the conclusions are outlined.

6.2 Historical Background

Thousands satellites are orbiting the Earth planet today. These artificial objects have different purposes and operational capabilities, but what unities many of them is a necessity to send data to the Earth and to receive data from the Earth. Making space communications reliable and at any time operable regardless the orbit of the satellite is a very demanding and important task of any space mission.

The tracking and data relay satellites are a response to this challenge. They allow transmitting data and receiving data from different platforms – such as satellites, unmanned vehicles, aircraft, the International Space Station, and many others. This data transmission is supported by different mechanisms – the geostationary satellites, and the network of ground stations.

The first example of the system built upon the geostationary satellites is the NASA's Tracking and Data Relay Satellite (TDRS) system. TDRS program was launched in 1970's, and since then thirteen TDRS satellites were launched to geostationary orbit, for which they are all designed and built. The first generation of

TDRS included seven satellites (TDRS-1, TDRS-2, TDRS-3, TDRS-4, TDRS-5, TDRS-6, TDRS-7). These satellites were launched in the time period from 1983 to 1995. Note that during the production phase each satellite had the name with a letter, for example, “TDRS-A”. Once on-orbit it was assigned the number – for instance, “TDRS-1”. Due to this TDRS-2 had never appeared on the map, as it was destroyed as TDRS-B during the launch. In this case study we will name each satellite with the number at the end. The second generation TDRS satellites were launched in 2000-2002 (TDRS-8, TDRS-9, TDRS-10), which are still on orbit. And finally, the third generation TDRS satellites (TDRS-11, TDRS-12, TDRS-13) were launched in 2013, 2014, and 2017, respectively. Another intrinsic part of the TDRS system is the TDRS ground stations – the first is the White Sands Ground Terminal (WSGT) in New Mexico, and the second is the Guam Remote Ground Terminal (GRGT) in Guam. GRGT is covering the Zone of Exclusion over the Indian Ocean. Note that in this dissertation we only focus on TDRS-A satellite.

The second example of space-based communication system is the European Data Relay System (EDRS). It comprises of two geostationary placed satellites – EDRS-A (launched in 2016) and EDRS-C (expected to be launched in the second quarter of 2019). Once these two satellites are on orbit, the system will support the coverage for LEO satellites over Europe, the Americas, the Middle East, Africa, Asia, and the Poles. Two more satellites to be launched after 2020 would allow the system to provide a global coverage of the Earth. Similarly to the NASA’s TDRS system, the European complex would have three ground stations – in case of EDRS they are located in Germany, Belgium, and UK.

The third alternative that is explored in this case study is the Near Earth Network (NEN), which is the network of 15 ground stations, 4 of which are NASA's ones, 9 are in commercial usage, and 2 are the ground stations of partner agencies. These ground stations are demonstrated in Figure 6.1.

Figure 6.1: The ground stations of the Near Earth Network (image: NASA)

The model-based *system concept representation framework* would allow the system architect to *encode* each one of these alternative concepts keeping the same level of information granularity and modeling capabilities. In turn, this would create the means for a *formal analysis* and quantitative measure for conceptual difference between the alternative concepts. Another advantage is the ability to engage the concurrent engineering design environment on early phases of the design process. And finally, based on the proposed approach the system engineer could integrate all the knowledge to create the future space communication mission, such as the Space Communication and Navigation (SCaN) program (NASA; Sanchez et al. 2014).

6.3 Motivation and Context

The space segment, which we will call TDRSS (TDRS system) in this study, was established to replace NASA's network of ground stations. The objective of the system is to provide tracking and data relay services to customer missions. The system comprises of a constellation of geosynchronous satellites and associated ground stations (NASA 2017).

A constellation of geostationary satellites includes TDRS satellites, launched starting from 1983 to 2017. Over the last 3 decades three generation of tracking and data relay satellites were developed. Table 6.1 summarizes the status of each satellite that was built within the TDRS program and which generation it belongs to.

Table 6.1: Status of TDRS fleet

Name	Generation	Launch year	Status
TDRS-1	First generation	1983	Decommissioned
TDRS-2	First generation	1986	Destroyed
TDRS-3	First generation	1988	In storage
TDRS-4	First generation	1989	Decommissioned
TDRS-5	First generation	1991	In storage
TDRS-6	First generation	1993	Active
TDRS-7	First generation	1995	Active
TDRS-8	Second generation	2000	Active
TDRS-9	Second generation	2002	Active
TDRS-10	Second generation	2002	Active
TDRS-11	Third generation	2013	Active
TDRS-12	Third generation	2014	Active
TDRS-13	Third generation	2017	Active

The ground segment of TDRS system includes two stations (New Mexico and Guam) that provide the global coverage of the Earth.

Comparing to NASA's history of tracking and data relay satellites development, ESA has a shorter history. The first two satellites, named EDRS-A and EDRS-C, were launched in 2016 and 2019, respectively. The European program is a result of public-private partnership between ESA and Airbus Defense and Space. The European Commission is EDRS's anchor customer through its Sentinel-1 and Sentinel-2 missions (ESA 2016). In Figure 6.2 the EDRS-A satellite is demonstrated.

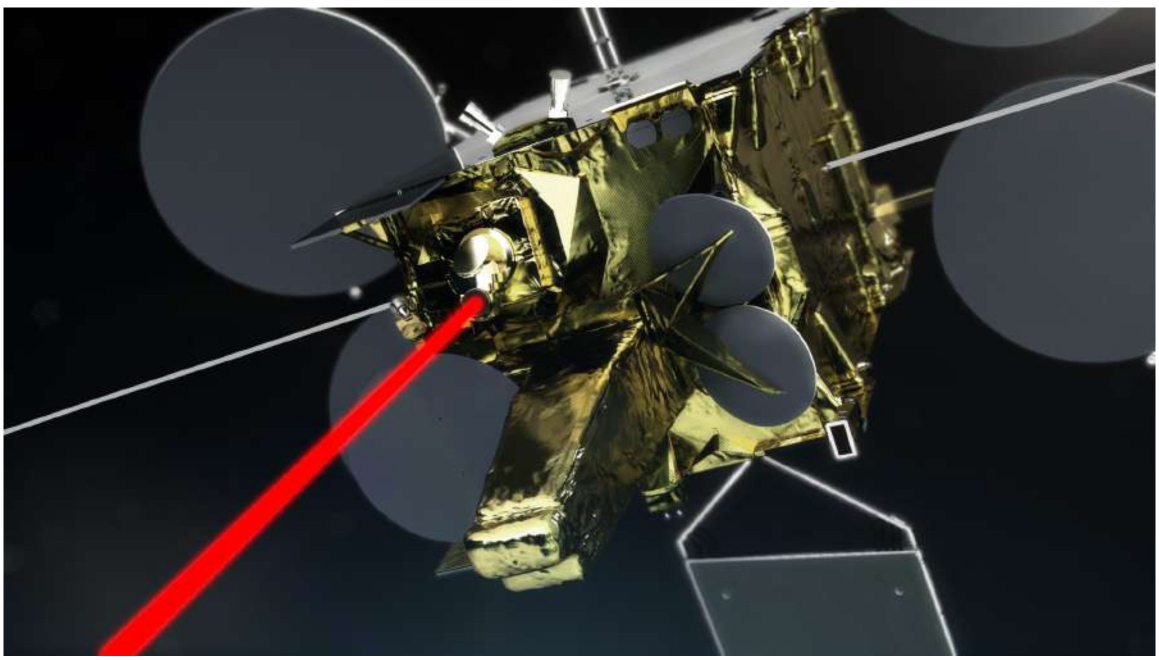


Figure 6.2: ESA's EDRS-A satellite launched in 2016

ESA has a variety of ground stations that could be used for data transmitting and receiving. This network of ground stations includes the ones located in Germany, Belgium, and UK.

NASA's Near Earth Network (NEN) provides communications services to space assets by means of ground stations located around the world (NASA 2016) – the set of ground stations are mentioned in Figure 6.1.

The chosen space communication concepts are presented differently: the level of details, the used terminology, and the concept of operations are explained based on the individual preferences of specific design team. In our work we are proposing the universal model-based *system concept representation framework* that would allow to *encode* the alternative concepts in a systematic way, and would support a *formal analysis*. The knowledge generated with support of such a framework is *reused* at later stages of product development process.

Aforementioned context is a great motivation to demonstrate applicability of proposed framework to space communication systems.

6.4 Specific Objectives

We discussed the motivation for inclusion of the model-based *system concept representation framework* for space communication missions. This sub-section is dedicated to the specific objectives that we aim at achieving by doing this case study. These objectives are:

- Identify the architectural decisions that form the space communication mission concepts;
- *Encode* a conceptual information about the alternative space communication missions into the proposed framework. These alternatives are represented by the TDRS

system; EDRS system; and Near Earth Network system. We will demonstrate such entries as stakeholders and their needs (proposition I), solution-neutral and solution-specific environments (propositions II and III, respectively), the integrated concept (proposition IV), and the concept of operations (proposition V) for each alternative of this case study;

• Demonstrate how the proposed approach could support a *formal analysis*, such as conceptual similarity assessment;

• Demonstrate how the model-based approach supports the interconnections between the architectural decisions and model-based solution diagrams.

6.5 Model-Based System Concept Representation Frameworks Development

This section is dedicated to demonstration of the sys*tem concept representation framework* and its entries for all three alternative concepts. In sub-section 6.5.1 we briefly outline the key entries of the *system concept representation framework* and their allocation to five propositions. In sub-section 6.5.2 we illustrate the architectural decisions for space communication missions. The first proposition related to the stakeholders and their needs is filled in in sub-section 6.5.3. The abstract solution-neutral information for space communication missions is provided in sub-section 6.5.4, while the alternative concepts are outlined at the solution-specific sub-section 6.5.5. The integrated concepts for TDRSS, EDRS, and NEN concepts are mentioned in sub-section 6.5.6. The outcomes of the first level decomposition of alternative concepts are summarized in sub-section 6.5.7. The integrated concepts at the second level decomposition for all three space communication projects are discussed in sub-section 6.5.8. The outcomes of the second level decomposition of the integrated concepts are provided in sub-section 6.5.9. At the next sub-section we provide the second level decomposition for all three alternative concepts. Sub-section 6.5.10 is dedicated to the methods of capturing the conceptual design information through DSM. Sub-section 6.5.11 discusses the fifth proposition of concept framework, a concept of operations, for the space communication missions.

6.5.1 System Concept Representation Framework Introduction

As it was discussed in Chapter 3 and sub-section 5.5.1, the system concept representation framework consists of 28 entries spread among 5 propositions: stakeholders (I), solution-neutral environment (II), solution-specific environment (III), integrated concept (IV), and concept of operations (V). In this Section we will present the *system concept representation frameworks* built for each alternative concept of the chosen space communication missions – TDRS system, EDRS system, and NEN system.

6.5.2 Architectural Decisions for Space Communication Missions

In the previous case study, we have explained the rationale and importance of the *architectural decisions* (Crawley et al. 2015) and how do they affect the system concept development. In this sub-section we demonstrate the architectural decisions for space communication systems.

The elicitation of architectural decisions is based on the definition of the key processes and associated forms that are used as the instruments performing the functions. These processes and forms are the most critical entries that should be taken into account during the analysis or development of space communication systems.

During the development of the architectural decisions table we need to focus on the processes that should be present regardless the chosen concept. Among these processes are “relay-to-ground linking”, “relay-to-user linking”, “intersatellite linking”, “user-to-ground linking”, “TT&C transmitting”, “high-rate transmitting”, “low-rate transmitting”, “multiple-user transmitting”, and “storing the information”. Each one of these processes is included in Table 6.2 that contains the list of architectural decisions.

Once the list of processes is established we may focus on the forms that could serve as the instruments for each process: “RG link”, “RU link”, “IS link”, “UG link”, “TT&C transmitter”, “High-rate transmitter”, “Low-rate transmitter”, “Multiple-access link”, and “Storage device”.

After that we could proceed with different options that may or may not be present for a specific process. This information is mentioned under the columns “Opt 1”, and “Opt 2” (yes/no) in Table 6.2.

Table 6.2: Architectural decisions for spacecom missions

N	Process	Form	Opt 1	Opt 2
1	Relay-to-Ground Linking	RG Link	Yes	No
2	Relay-to-User Linking	RU Link	Yes	No
3	Intersatellite Linking	IS Link	Yes	No
4	User-to-Ground Linking	UG Link	Yes	No
5	TT&C transmitting	TT&C transmitter	Yes	No
6	High-rate transmitting	High-rate transmitter	Yes	No
7	Low-rate transmitting	Low-rate transmitter	Yes	No
8	Multiple-user transmitting	Multiple access link	Yes	No
9	Storing the information	Storage device	Yes	No

As we have shown in Chapter 5 for a suborbital human spaceflight case study, and as we will show later in Chapter 6 for current case study, the architectural decisions table is important, because it contains the essential information about form-function allocation. The model-based *system concept representation framework* complements this table and puts it in a rigorous ontology.

6.5.3 Proposition I: Stakeholders and Stakeholders' Needs for Space Communication Systems

The stakeholders and their needs constitute the first entries of concept framework. In order to define them we have taken the list presented in (Sanchez et al. 2014) and (Sanchez et al. 2013). Among the stakeholders of TDRSS such organizations as NASA, the U.S. Geological Survey (USGS), the National Oceanic and Atmospheric Administration (NOAA), NSF Antarctic Program, International Partners are identified.

We have discussed the different approaches on who should be considered as stakeholders in sub-section 3.3.1, stating that in our study by stakeholder we imply individuals or organizations that can affect the system or be affected by the system. They can do this by means of purchasing or not purchasing our product and/or service, for instance. Secondly, stakeholders have some needs, which can be satisfied by our product and/or service.

For the purpose of our study we will name the stakeholders as "operators of spacecraft" (entry 1) and the need as "get data" (entry 2). Note that this information is presented in an abstract way highlighting a nature of often fuzzy stated needs. The information about stakeholders and their needs is summarized in Table 6.3 and Figure 6.3.

Table 6.3: Entries 1 and 2 of concept framework for spacecom concept

Proposition I: Stakeholders		
1	**Stakeholders**	**Operators of spacecraft**
2	Need	Get data

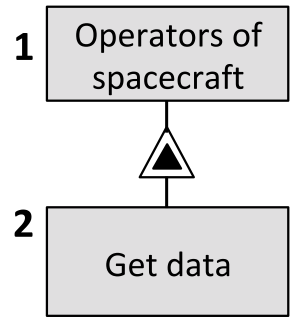


Figure 6.3: Stakeholders (I) proposition for spacecom concept

6.5.4 Proposition II: Solution-Neutral Operand and Process for Space Communication Missions

The solution-neutral environment is fundamentally about the definition of solution-neutral operand (entry 3) and its attributes (entries 4 and 5), as well as the solution-neutral process (entry 6) with attribute (entry 7). The core idea that lies behind this proposition is to define the functional intent that would later allow us to specify the abstract information into more concrete to define the alternative concepts.

In space communication case study we define "information" as the solution-neutral operand that has the value-related attribute "location". The other attribute would be "volume". The location is the value-related attribute, because the solution-neutral process "changing" changes the location of information from, for example, satellite to ground station, or vice versa. The attribute of the solution-neutral process would be "data rate". This information is demonstrated in Table 6.4 and Figure 6.4.

Table 6.4: Entries 3 to 7 of concept framework for spacecom concept

Proposition II: Solution-neutral environment (Problem statement)		
3	**Solution-neutral operand (SNO)**	**Information**
4	SNO value attribute	Location
5	SNO other attribute	Volume
6	**Solution-neutral process (SNP)**	**Changing**
7	SNP attribute	Data rate

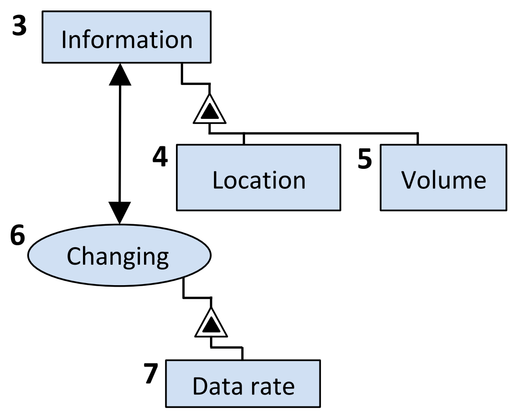


Figure 6.4: Solution-neutral (II) proposition for spacecom concept

It is important that from both representation – of Table 6.4 and Figure 6.4 – we get exactly the same information. One of the benefits of using the OPD is that the designers from different parts of the world can *encode* information about concept in the same way. Thus, by assigning a number to each entry of concept framework we may keep track of every entry and its inheritance. Note that from OPD the one can see the absence of the instrument that executes the function "changing information's location". This is due to the fact that we are in the solution-neutral domain, and in order to assign the instrument to function we should first specialize both the solution-neutral operand

and the solution-neutral process. This will be done at the next sub-section dedicated to the conceptual design phase.

6.5.5 Proposition III: Conceptual Design Phase for Space Communication Missions

The specialization link plays an important role in the conceptual design phase. There is a clear difference between, for example, specialization and decomposition links.

Consider an example of the process "moving" (with the implicit instrument "vehicle") – see Figure 6.5(a) – that is provided in the paper of Deubzar and Lindemann (Deubzer and Lindemann 2009). In contrast, Figure 6.5(b) shows the specialization of "moving" into three alternative processes – flying (pushing down on air), floating (pushing down on water) and rolling (pushing down on solid ground). The figures 6.5(a) and 6.5(b) clearly demonstrate the difference between decomposition, realized by dividing process "moving" into smaller sub-processes "storing (energy)", "converting (energy)", "using (energy)", and specialization, realized by relating general process "moving" to such types of that process as "flying", "floating", and "rolling". We see that decomposition and specialization convey different information and both types of information are important and should be considered during the conceptual design phase.

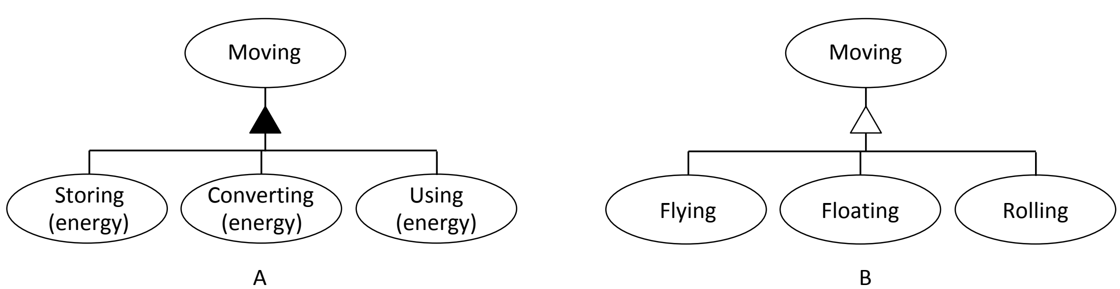


Figure 6.5: Decomposition (a) and specialization (b) relationships of the process *moving*

The specialization link is widely used during the conceptual design phase – when we specialize the solution-neutral operand (entry 3) to the solution-specific operand (entry 8), or the solution-neutral process (entry 6) to the solution-specific process (entry 11). In turn, the decomposition is commonly used at the level of integrated concept to be discussed at the next sub-section – there we will decompose the chosen alternative into its constituents.

The specialization and decomposition types of relationships are discussed in details in (Menshenin and Crawley 2018a).

Applying the conceptual design process to the space communication case study, we may see that the solution-neutral operand "information" specializes to the solution-specific operand "EM signal" (see Table 6.5 and Figure 6.6). According to the first rule of the concept framework development, formulated in section 3.5.3 in Chapter 3, there is a principle of inheritance of attributes. Thus, the attributes mentioned as entries 9, 10, 12 in Table 6.5 are inherited from the solution-neutral's environment. Additionally, some of the new attributes appearing, such as "frequency". The solution-neutral process "changing" specializes to the solution-specific process "transmitting" (entry 11), which has an attribute "secure" (entry 12).

Table 6.5: Entries 8 to 16 of concept framework for spacecom concept

8	**Solution-specific operand (SSO)**	**EM signal**	
9	SSO value attribute	Location	
10	SSO other attribute	Frequency	
11	**Solution-specific process (SSP)**	**Transmitting**	
12	SSP attribute	Secure	
13	**Generic Form**	**Data relay system**	
14	Generic Form attribute	Cost	
15	**Specific Form**	**15A**	**TDRS system**
		15B	**EDRS system**
		15C	**NEN system**
16	Specific Form attribute	**16A**	**Cost**
		16B	**Cost**
		16C	**Cost**

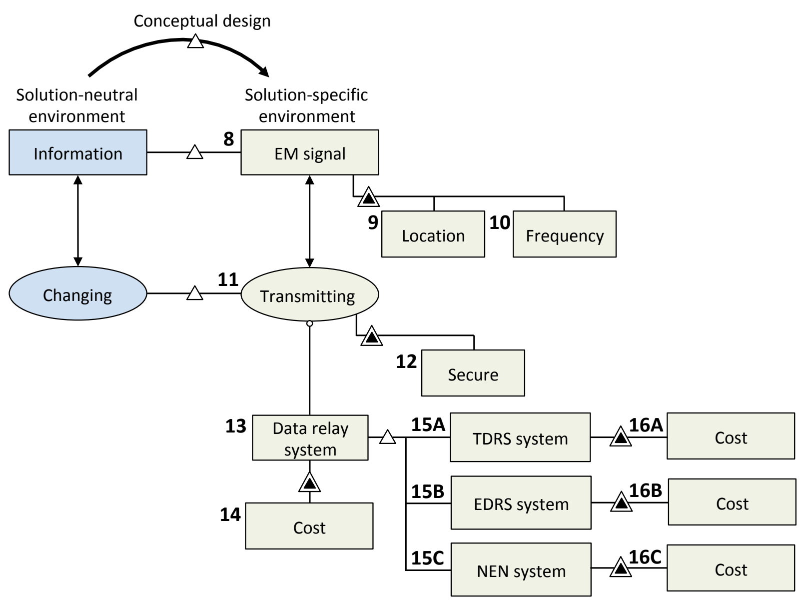


Figure 6.6: Solution-specific (III) proposition for spacecom concept

The main outcome of the conceptual design process is the identification of the generic form (entry 13) and the specific form (entry 15) and their attributes. In Table 6.5 and Figure 6.6 we define "Data relay system" as the generic form that is specialized in alternative concepts, such as "TDRS system" (entry 15A), "EDRS system" (entry 15B), and “NEN system” (entry 15C). Each one of these forms has the attribute “cost” (entries 14 and 16).

We have chosen these three alternative concepts, as they should represent two similar projects (TDRSS and EDRS) built upon the geostationary satellites and ground stations, and one completely different one (NEN), as it is based on the network of ground stations.

Up until now concept framework contains the stakeholders information (proposition I), solution-neutral information (proposition II), and solution-specific information (proposition III). The entries 1 to 14 are relevant to any concept of the space communication missions that we discuss in this Chapter. The real difference between alternatives starts to appear beginning with entries 15 and 16 and subsequent decomposition of specific forms into the integrated concepts.

6.5.6 Proposition IV: Integrated Concepts for Alternative Space Communication Systems: First Level Decomposition

The integrated concept *encodes* the essential information about the decomposition of specific form into its internal elements of form (entries 22), internal processes (entries 20), and internal operands (entries 17) with corresponding attributes (entries 18, 19, 21, and 23). In this sub-section we demonstrate the integrated concepts for the first level decomposition of space communication projects: TDRS system, EDRS system, and NEN system.

- Integrated concept for TDRSS: first level decomposition

At the first level decomposition of specific form "TDRSS" (entry 15A) is decomposed into internal elements of form "TDRS" (entry 22A1) and "ground station" (entry 22A2). Note that "A1" at the end of entry specifies the information for TDRS, while "A2" is related to the ground station.

The level of integrated concept reveals an important role of model-based representation. From Table 6.6 and Figure 6.7 we see that the function of TDRS is "transmitting EM signal" (entries 20A1 and 17A1), the same is the function of ground station – "transmitting EM signal" (entries 20A2 and 17A2). EM signal has such value attribute and other attribute as "location" and "frequency" (entries 18A1/18A2 and 19A1/19A2, respectively). The internal processes "transmitting" have the attributes "secure" (entries 21A1/21A2, respectively). TDRS and ground station have the attribute "cost" (entries 23A1/23A2).

The information presented in Table 6.6 and Figure 6.7 has the same meaning and represents the core information about the TDRS concept decomposition and assigning the internal elements of form to the internal functions.

Table 6.6: Entries of integrated concept for TDRSS in grid

		A1: Satellite fleet	A2: Ground station
17	**Internal Operands (IO)**	**EM signal**	**EM signal**
18	IO value attribute	Location	Location
19	IO other attribute	Frequency	Frequency
20	**Internal Processes (IP)**	**Transmitting**	**Transmitting**
21	IP attribute	Secure	Secure
22	**Internal Elements of Form (IEoF)**	**TDRS**	**Ground station**
23	IEoF attribute	Cost	Cost
24	Structure	TDRS connected remotely Ground station	
25	Interactions	TDRS transmits/receives signal Ground station	

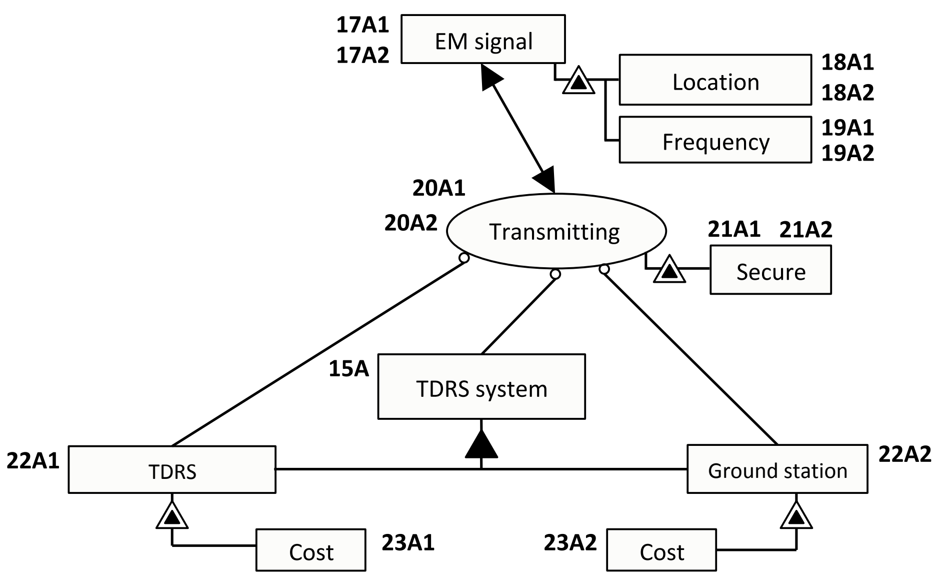


Figure 6.7: Integrated concept (IV) proposition for TDRSS concept

In Figure 6.8 we demonstrate the model-based representation of structure and interactions information for the TDRS system. From upper side of the Figure we see the information about formal allocation of elements to each other: ground station is "connected remotely" to TDRS. From the lower side of the Figure we notice what's exchanged among the elements on the functional level: ground station "transmits/receives signal" to/from TDRS.

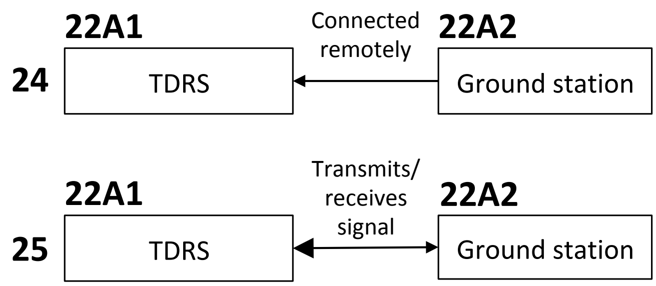


Figure 6.8: Structure and Interactions for TDRSS system

- Integrated concept for EDRS system: first level decomposition

Table 6.7 and Figures 6.9 and 6.10 demonstrate the first level decomposition of the specific form "EDRS system" (entry 15B). At the first level decomposition the EDRS system is not much different from the TDRSS. As such we may see that the specific form "EDRS system" is decomposed into two internal elements of form – EDRS (entry 22B1) and ground station (entry 22B2). Their attributes are costs (entries 23B1 and 23B2, respectively). The internal function of EDRS is "transmitting EM signal" (entries 20B1 and 17B1), and the same internal function of ground station is "transmitting EM signal" (entries 20B2 and 17B2, respectively). The operand "EM signal" has value attribute and other attribute "location" and "frequency" (entries 18B1/18B2 and 19B1/19B2, respectively). The attribute of internal process "transmitting" is "secure" (entries 21B1/21B2).

Table 6.7: Entries of integrated concept for EDRS in grid

		B1: Satellite fleet	B2: Ground station
17	**Internal Operands (IO)**	**EM signal**	**EM signal**
18	IO value attribute	Location	Location
19	IO other attribute	Frequency	Frequency
20	**Internal Processes (IP)**	**Transmitting**	**Transmitting**
21	IP attribute	Secure	Secure
22	**Internal Elements of Form (IEoF)**	**EDRS**	**Ground station**
23	IEoF attribute	Cost	Cost
24	Structure	EDRS connected remotely Ground station	
25	Interactions	EDRS transmits/receives signal Ground station	

The same information is contained in the model-based representation of the first level decomposition of the EDRS system, presented in Figure 6.9.

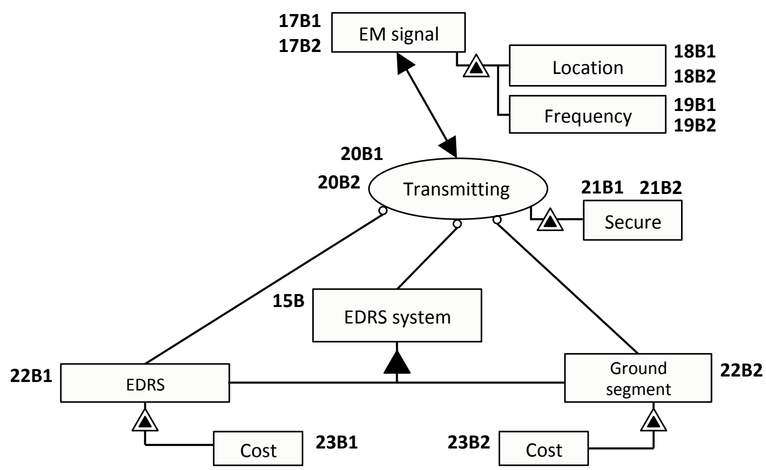


Figure 6.9: Integrated concept (IV) proposition for EDRS concept

The structure and interactions information are presented in Figure 6.10. From this Figure we see that the formal allocation of elements to each other is that the ground station is “connected remotely” to EDRS. The functional interaction is that ground station “transmits/receives signal” to/from EDRS.

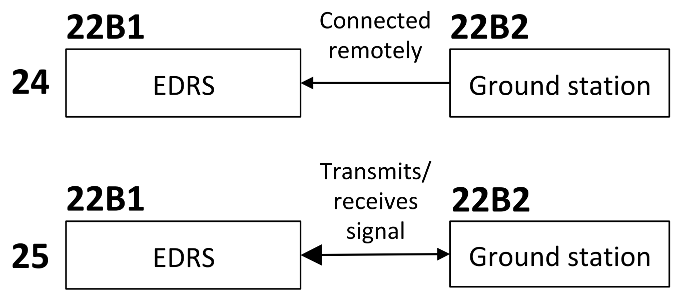


Figure 6.10: Structure and Interactions for EDRS system

- Integrated concept for NEN system: first level decomposition

NEN system decomposes into internal element of form “ground station” on the same level of granularity as it is for the first two cases – TDRSS and EDRS system. They have two decomposed elements. Nevertheless, in order to be consistent with previous two examples we will demonstrate the first level decomposition of NEN.

The internal element of form ground station (entry 22C) has an attribute “cost” (entry 23C). The internal process is “transmitting” (entry 20C) with an attribute “secure” (entry 21C). The operand is “EM signal” (entry 17C). Its value attribute is “location” (entry 18C), and other attribute is “frequency” (entry 19C).

Table 6.8: Entries of integrated concept for NEN in grid

		C: Near Earth Network
17	**Internal Operands (IO)**	**EM signal**
18	IO value attribute	Location
19	IO other attribute	Frequency
20	**Internal Processes (IP)**	**Transmitting**
21	IP attribute	Secure
22	**Internal Elements of Form (IEoF)**	**Ground station**
23	IEoF attribute	Cost

The model-based representation of the same information is indicated in Figure 6.11 that shows the core information about the NEN system and its function. Note that for the above-mentioned reasons here we do not represent the structure and interactions information.

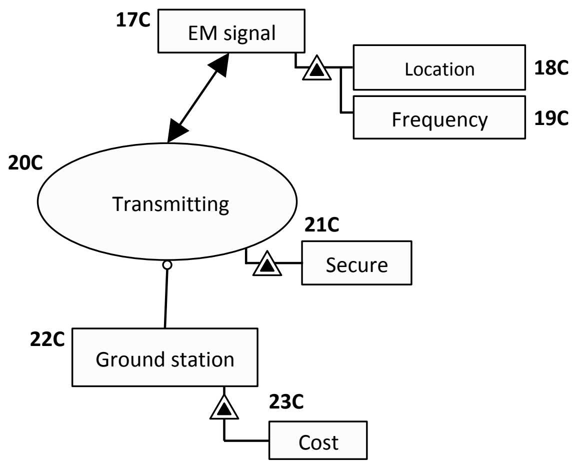


Figure 6.11: Integrated concept (IV) proposition for NEN concept

6.5.7 Outcomes of the First Level Decomposition of Alternative Concepts

In Chapter 4 we have discussed the importance of capturing an appropriate level of granularity in concept's decomposition. We supported that claim by illustrating the example of patents analysis.

A similar result could be found during the analysis of the first level decomposition of the space communication missions. Comparing the conceptual difference between the TDRS system and EDRS system (see Figure 6.12) we can note that in both cases the internal functions are the same – "transmitting EM signal". The only thing that is different is the in-space system – “TDRS” in case of TDRSS and “EDRS” in case of EDRSS.

15A: TDRSS		
	A1: TDRS	**A2: Ground station**
17	**EM signal**	**EM signal**
18	Location	Location
19	Frequency	Frequency
20	**Transmitting**	**Transmitting**
21	Secure	Secure
22	**TDRS**	**Ground station**
23	Cost	Cost

15B: EDRSS		
	B1: EDRS	**B2: Ground station**
17	**EM signal**	**EM signal**
18	Location	Location
19	Frequency	Frequency
20	**Transmitting**	**Transmitting**
21	Secure	Secure
22	**EDRS**	**Ground station**
23	Cost	Cost

Figure 6.12: Comparison of conceptual difference between the TDRS (left) and EDRS (right) systems

Concerning the second sub-system we see that at the first level decomposition they are the same – “ground station”. The internal functions are also the same: “transmitting EM signal”.

The comparison of TDRS system (15A) and NEN system (15C) reveals a different result – see Figure 6.13. The core difference is the number of modules that are used to satisfy the stakeholders’ needs. In case of TDRS we have two sub-systems: TDRS (entry 15A1) and ground station (entry 15A2). In case of NEN it is decomposed into the ground station (entry 15C).

15A: TDRSS		
	A1: TDRS	**A2: Ground station**
17	**EM signal**	**EM signal**
18	Location	Location
19	Frequency	Frequency
20	**Transmitting**	**Transmitting**
21	Secure	Secure
22	**TDRS**	**Ground station**
23	Cost	Cost

15C: NEN	
	C: NEN
17	**EM signal**
18	Location
19	Frequency
20	**Transmitting**
21	Secure
22	**Ground station**
23	Cost

Figure 6.13: Comparison of conceptual difference between the TDRSS (left) and NEN (right) systems

A similar result can be seen for the comparison of EDRS system and NEN system (see Figure 6.14). The reason is the same as it is for the previous example.

15B: EDRSS		
	B1: TDRS	**B2: Ground station**
17	**EM signal**	**EM signal**
18	Location	Location
19	Frequency	Frequency
20	**Transmitting**	**Transmitting**
21	Secure	Secure
22	**EDRS**	**Ground station**
23	Cost	Cost

15C: NEN	
	C: NEN
17	**EM signal**
18	Location
19	Frequency
20	**Transmitting**
21	Secure
22	**Ground station**
23	Cost

Figure 6.14: Comparison of conceptual difference between the EDRSS (left) and NEN (right) systems

By analogy with previous case study, we present the DSM-based methods to manage the information about the integrated concepts (see Figure 6.15).

The DSM part of Figure 6.15 contains the specialization relationships among the entries (for instance, entry 13 “Data relay system” specializes into entries 15A “TDRS system”, 15B “EDRS system”, and 15C “NEN system”) and the decomposition relationships (for example, entry 15A “TDRS system” decomposes into the internal elements of form entry 22A1 “TDRS” and entry 22A2 “Ground station”). This information is correlated with the OPDs presented in Figures 6.7, 6.9, 6.11.

Another important information *encoded* into DSM part of the matrix is the information about the structure and interactions. For example, Figure 6.15 informs the system engineer that the entry 22A2 “ground station” is connected remotely to entry 22A1 “TDRS” (structure), or that entry 22A1 “TDRS” transmits/receives signal to entry 22A2 “Ground station” (interactions). This data is correlated with the one mentioned in Figures 6.8 and 6.10.

Full DMM													
				13 Data relay system					**11** Transmitting				
				15A TDRS system		**15B** EDRS system		**15C** NEN system	**11** Transmitting		**11** Transmitting		**11** Transmitting
				22A1 TDRS	**22A2** Ground station	**22B1** EDRS	**22B2** Ground station	**22C** Ground station	**20A1** Transmitting	**20A2** Transmitting	**20B1** Transmitting	**20B2** Transmitting	**20C** Transmitting
13 Data relay system	**15A** TDRS system	**22A1**	TDRS		Connected remotely Transmits/ receives signal				V				
		22A2	Ground station	Connected remotely Transmits/ receives signal						V			
	15B EDRS system	**22B1**	EDRS				Connected remotely Transmits/ receives signal				V		
		22B2	Ground station			Connected remotely Transmits/ receives signal						V	
	15C NEN system	**22C**	Ground station										V

DSM DMM

Figure 6.15: DSM/DMM for the integrated concepts of spacecom concepts

The DMM part of the Figure 6.15 contains the information about the allocation of the internal elements of form to the internal processes – signs “V” at the intersection of corresponding cells inform us about this.

Summarizing the results of the first level decomposition we can conclude that at this level of granularity the real difference is related to the number of sub-systems that are responsible for the internal functions. We expect to extract a deeper knowledge about the conceptual difference among alternative concepts at the second level decomposition analysis.

6.5.8 Integrated Concepts for Alternative Space Communication Systems: Second Level Decomposition

In this sub-section we will decompose the TDRSS, EDRS, and NEN concepts into the second level decomposition. We will also present the architectural decisions tables for each alternative.

- Integrated concept for TDRSS: second level decomposition

The TDRSS is decomposed into two sub-systems: tracking and data relay satellites and ground station. The architectural decisions for the first sub-system – TDRS – are presented in Figure 6.16, while the architectural decisions for the second sub-system – ground stations – are presented in Figure 6.17. These architectural decisions inform the system architect about form to process allocation. Note that in this Thesis we only focus on TDRS-A satellite.

For example, from Figure 6.16 we may see that the form “RG link” provides the process “Relay-to-Ground linking” for the first sub-system of TDRS system. Note that the actual name of the “RG link” may have different names depending on the concept under consideration. For example, in case of TDRS satellite this is relay-to-ground antenna.

N	Process	Form	Opt 1	Opt 2
1	Relay-to-Ground Linking	RG Link	Yes	No
2	Relay-to-User Linking	RU Link	Yes	No
3	Intersatellite Linking	IS Link	Yes	No
4	User-to-Ground Linking	UG Link	Yes	No
5	TT&C transmitting	TT&C transmitter	Yes	No
6	High-rate transmitting	High-rate transmitter	Yes	No
7	Low-rate transmitting	Low-rate transmitter	Yes	No
8	Multiple-user transmitting	Multiple access link	Yes	No
9	Storing the information	Storage device	Yes	No

Figure 6.16: Architectural decisions for the first sub-system (TDRS) of TDRS system

N	Process	Form	Opt 1	Opt 2
1	Relay-to-Ground Linking	RG Link	Yes	No
2	Relay-to-User Linking	RU Link	Yes	No
3	Intersatellite Linking	IS Link	Yes	No
4	User-to-Ground Linking	UG Link	Yes	No
5	TT&C transmitting	TT&C transmitter	Yes	No
6	High-rate transmitting	High-rate transmitter	Yes	No
7	Low-rate transmitting	Low-rate transmitter	Yes	No
8	Multiple-user transmitting	Multiple access link	Yes	No
9	Storing the information	Storage device	Yes	No

Figure 6.17: Architectural decisions for the second sub-system (ground station) of TDRS system

As we discussed in the previous Chapter, the system architecture principles remain the same regardless the level of granularity. Thus, from Table 6.6 and Figure 6.7 we may notice that at the second level we will decompose the TDRS and ground station into their own internal elements of form and will assign them to the internal processes that act on the internal operands.

In Figure 6.18 we present the second level decomposition of the TDRSS' first sub-system "TDRS".

17A1	**Internal Operands (IO)**	**.1**	**.1**	**.1**	**.1**	**.1**	**.1**	**.1**	**.1**	**.1**
		EM signal	**EM signal**	**EM signal**	**EM signal**	**EM signal**	**EM signal**	**EM signal**	**EM signal**	**EM signal**
18A1	IO value attribute	**.1**	**.2**	**.3**	**.4**	**.5**	**.6**	**.7**	**.8**	**.9**
		Location	Location	Location	Location	Location	Location	Location	Location	Location
19A1	IO other attribute	**.1**	**.2**	**.3**	**.4**	**.5**	**.6**	**.7**	**.8**	**.9**
		Frequency	Frequency	Frequency	Frequency	Frequency	Frequency	Frequency	Frequency	Frequency
20A1	**Internal Processes (IP)**	**.1**	**.2**	**.3**	**.4**	**.5**	**.5**	**.6**	**.6**	**.6**
		High-rate transmitting	**Multiple-user transmitting**	**Relay-to-Ground transmitting**	**TT&C transmitting**	**Relay-to-User transmitting**	**Relay-to-User transmitting**	**Low-rate transmitting**	**Low-rate transmitting**	**Low-rate transmitting**
21A1	IP attribute	**.1**	**.2**	**.3**	**.4**	**.5**	**.6**	**.7**	**.8**	**.9**
		Secure	Secure	Secure	Secure	Secure	Secure	Secure	Secure	Secure
22A1	**Internal Elements of Form (IEoF)**	**.1**	**.2**	**.3**	**.4**	**.1**	**.2**	**.1**	**.2**	**.4**
		Single access antenna	**Multiple-access antenna**	**Relay-to-Ground antenna**	**Omni antenna**	**Single access antenna**	**Multiple-access antenna**	**Single access antenna**	**Multiple-access antenna**	**Omni antenna**
23A1	IEoF attribute	**.1**	**.2**	**.3**	**.4**	**.5**	**.6**	**.7**	**.8**	**.9**
		Cost	Cost	Cost	Cost	Cost	Cost	Cost	Cost	Cost

Figure 6.18: Second level decomposition of the TDRSS's first sub-system (TDRS)

From Figure 6.18 we see which internal element of form is used for which internal process, and on which internal operand it acts. We also assign the specific number to each entry. Thus, the above-mentioned figure informs us that, for example, "single access antenna" (entry 22A1.1) is used for "high-rate transmitting" (entry 20A1.1), "Relay-to-User transmitting" (entry 20A1.5), and "low-rate transmitting" (entry 20A1.6) of the "EM signal" (entry 17A1.1).

The second level decomposition of the TDRSS' second sub-system "ground station" is presented in Figure 6.19. From this Figure we may see that the internal element of form "ground-to-relay antenna" (entry 22A2.1) performs the internal process "ground-to-relay transmitting" (entry 20A2.1) to internal operand "EM signal" (entry 17A2.1), internal process "TT&C transmitting" (entry 20A2.2) to the same internal operand "EM signal" (entry 17A2.1), as well as the internal process "low-rate transmitting" (entry 20A2.3) to internal operand "EM signal" (entry 17A2.1).

17A2	**Internal Operands (IO)**	**.1**	**.1**	**.1**
		EM signal	**EM signal**	**EM signal**
18A2	IO value attribute	.1	.2	.3
		Location	Location	Location
19A2	IO other attribute	.1	.2	.3
		Frequency	Frequency	Frequency
20A2	**Internal Processes (IP)**	**.1**	**.2**	**.3**
		Ground-to-Relay transmitting	**TT&C transmitting**	**Low-rate transmitting**
21A2	IP attribute	.1	.2	.3
		Secure	Secure	Secure
22A2	**Internal Elements of Form (IEoF)**	**.1**	**.1**	**.1**
		Ground-to-Relay antenna	**Ground-to-Relay antenna**	**Ground-to-Relay antenna**
23A2	IEoF attribute	.1	.2	.3
		Cost	Cost	Cost

Figure 6.19: Second level decomposition of the TDRSS' second sub-system (ground station)

The attributes in the cells of Figures 6.18 and 6.19 are inherited from the solution-specific environment.

In the previous Chapter we discussed the drawback of having the table representations as the number of the concept framework's entries increasing. In a response to this challenge, we demonstrate the model-based representation of the same information as it is indicated in Figures 6.18 and 6.19. Figure 6.20 is the model-based representation of the second level decomposition for the TDRS system.

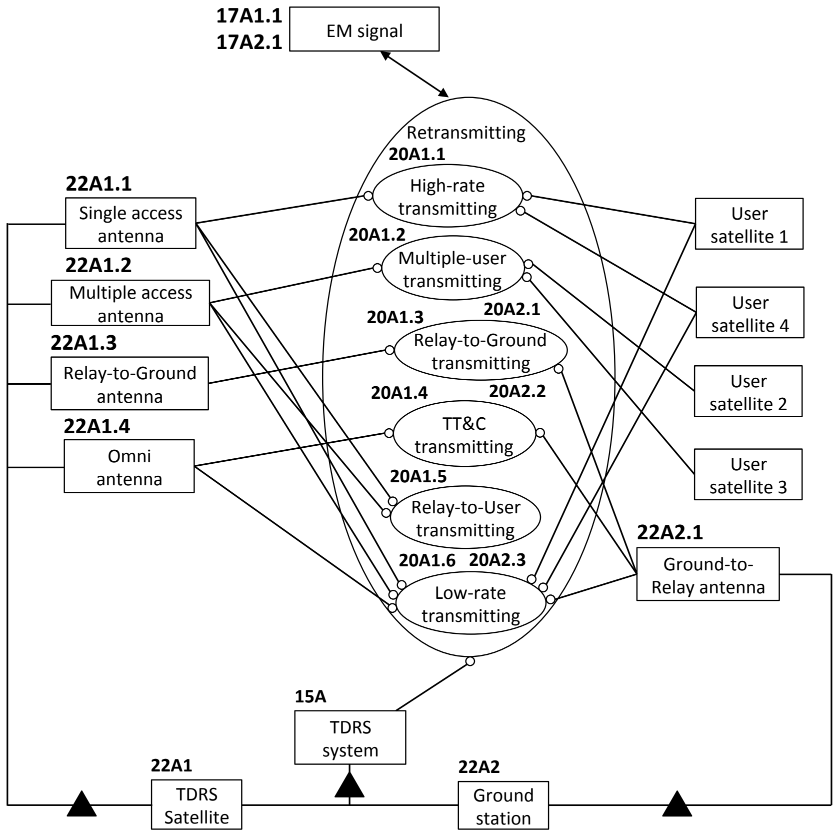


Figure 6.20: Model-based representation of the second level decomposition for the TDRS system

At the model presented in Figure 6.20 we see that the “TDRS system” (entry 15A) is decomposed into two sub-systems: “TDRS satellite” (entry 22A1) and “Ground station” (entry 22A2). TDRS satellite is further decomposed into four internal elements of form: “Single access antenna” (entry 22A1.1), “Multiple access antenna” (entry 22A1.2), “Relay-to-ground antenna” (entry 22A1.3), and “Omni-antenna” (entry 22A1.4). The ground station is further decomposed into the “Ground-to-relay antenna” (entry 22A2.1).

Figure 6.20 also informs us which internal element of form is used for which internal process. For example, “Single access antenna” (entry 22A1.1) performs three internal processes: “High-rate transmitting” (entry 20A1.1), “Relay-to-user transmitting” (entry 20A1.5), and “Low-rate transmitting” (entry 20A1.6). The “Multiple access antenna” (entry 22A1.2) is used for “Multiple-user transmitting” (entry 20A1.2), “Relay-to-User transmitting” (entry 20A1.5) and “Low-rate transmitting” (entry 20A1.6); the internal process for the “Relay-to-ground antenna” (entry 22A1.3) is “Relay-to-ground transmitting” (entry 20A1.3); the internal processes for “Omni antenna” (entry 22A1.4) are “TT&C transmitting” (entry 20A1.4) and “Low-rate transmitting” (entry 20A1.6).

The ground-to-relay antenna that is decomposed from the ground station has three internal processes: “Ground-to-relay transmitting” (entry 20A2.1), “TT&C transmitting” (20A2.2), and “Low-rate transmitting” (entry 20A2.3).

Note that in Figure 6.20 we also include 4 user satellites (labeled 1 to 4), which are consistent with the representation of Figure 6.21.

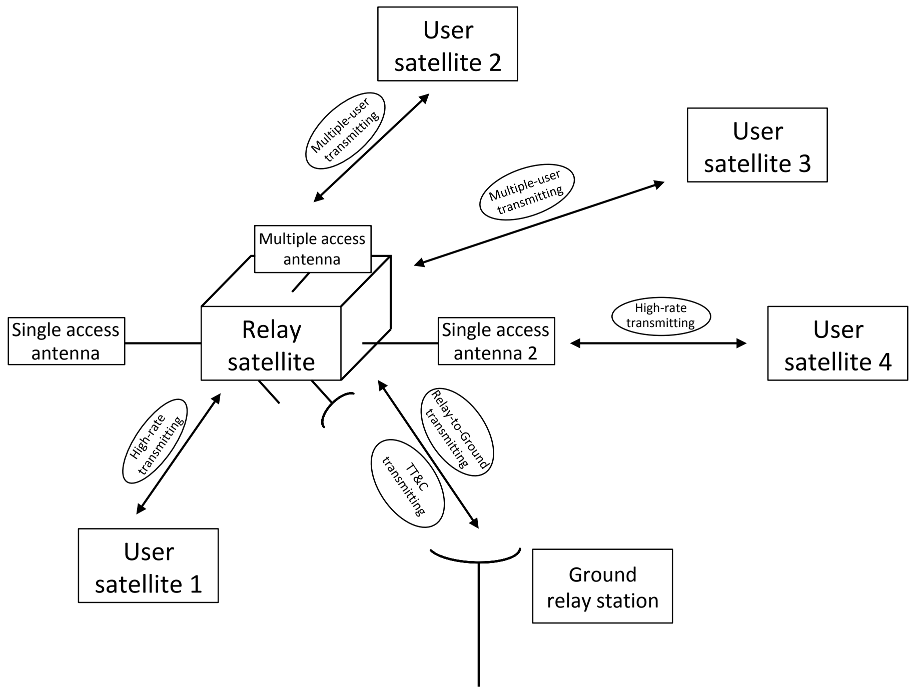


Figure 6.21: Example of interactions among TDRS system constituents

In Table 6.9 and Figure 6.22 the structure and interactions information is contained. From Figure 6.22(a) we see that the antennas (entries 22A1.1, 22A1.2, 22A1.3, and 22A1.4, and entry 22A2.1) are embedded into TDRS satellite (entry 22A1) or ground station (entry 22A2), respectively. The formal allocation of TDRS satellite and ground station is that they are in "line of sight". The same formal interaction is relevant for TDRS satellite and user satellite.

Table 6.9: Structure and Interactions for TDRSS concept (table format)

24	Structure	Single access antenna **Embedded** TDRS satellite Multiple access antenna **Embedded** TDRS satellite Relay-to-Ground antenna **Embedded** TDRS satellite Omni antenna **Embedded** TDRS satellite User satellite **Line of sight** TDRS satellite	Ground station **Line of sight** TDRS satellite Ground-to-Relay antenna **Embedded** Ground station
25	Interactions	Single access antenna **Transmits signal** TDRS satellite Multiple access antenna **Transmits signal** TDRS satellite Relay-to-Ground antenna **Transmits signal** TDRS satellite Omni antenna **Transmits signal** TDRS satellite User satellite **Transmits signal** TDRS satellite	Ground station **Transmits signal** TDRS satellite Ground-to-Relay antenna **Transmits signal** Ground station

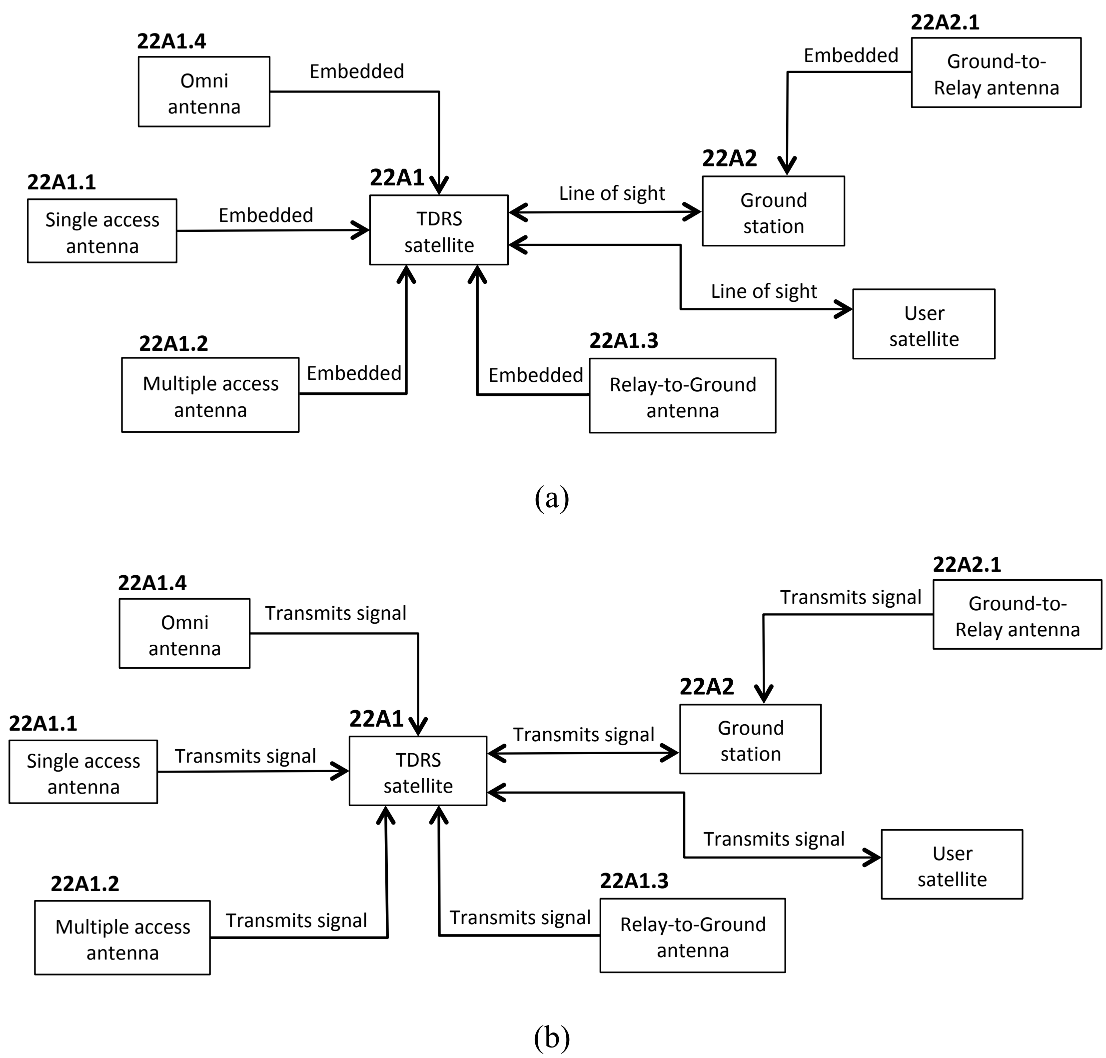


Figure 6.22: Structure (a) and interactions (b) information for the TDRS system

The type of interactions among the decomposed entries presented in Figure 6.22(b) is the typical one: the elements "transmit signal" from one location to another location.

The information presented here creates a benefit for the system architect as it contains the model-based representation keeping the ontology and semantics and providing the core information about the conceptual design phase for the TDRS system.

- Integrated concept for EDRS system: second level decomposition

Similarly to the previous example, EDRS system is decomposed into two sub-systems, or modules. The architectural decisions for the first module (EDRS) of EDRS system are presented in Figure 6.23, while the architectural decisions for the second module (ground segment) are presented in Figure 6.24.

N	Process	Form	Opt 1	Opt 2
1	Relay-to-Ground Linking	RG Link	Yes	No
2	Relay-to-User Linking	RU Link	Yes	No
3	Intersatellite Linking	IS Link	Yes	No
4	User-to-Ground Linking	UG Link	Yes	No
5	TT&C transmitting	TT&C transmitter	Yes	No
6	High-rate transmitting	High-rate transmitter	Yes	No
7	Low-rate transmitting	Low-rate transmitter	Yes	No
8	Multiple-user transmitting	Multiple access link	Yes	No
9	Storing the information	Storage device	Yes	No

Figure 6.23: Architectural decisions for the first module (EDRS satellite) of EDRS system

N	Process	Form	Opt 1	Opt 2
1	Relay-to-Ground Linking	RG Link	Yes	No
2	Relay-to-User Linking	RU Link	Yes	No
3	Intersatellite Linking	IS Link	Yes	No
4	User-to-Ground Linking	UG Link	Yes	No
5	TT&C transmitting	TT&C transmitter	Yes	No
6	High-rate transmitting	High-rate transmitter	Yes	No
7	Low-rate transmitting	Low-rate transmitter	Yes	No
8	Multiple-user transmitting	Multiple access link	Yes	No
9	Storing the information	Storage device	Yes	No

Figure 6.24: Architectural decisions for the second module (ground station) of EDRS system

We believe that at the second level decomposition the conceptual difference between competing alternatives become explicit. By decomposing the specific form into the internal elements of form we can assign the instrument to each internal process. Thus, we store the information about form-process allocation on different levels of granularity. The information about the decomposition of the EDRS system into both sub-systems is presented in Figure 6.25 (for EDRS satellite) and Figure 6.26 (for EDRS ground segment).

17B1	Internal Operands (IO)	.1	.1	.1	.1	.1	.1	.1	.1	.1	.1
		EM signal	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal
18B1	IO value attribute	.1	.2	.3	.4	.5	.6	.7	.8	.9	.10
		Location	Location	Location	Location	Location	Location	Location	Location	Location	Location
19B1	IO other attribute	.1	.2	.3	.4	.5	.6	.7	.8	.9	.10
		Frequency	Frequency	Frequency	Frequency	Frequency	Frequency	Frequency	Frequency	Frequency	Frequency
20B1	Internal Processes (IP)	.1	.1	.1	.2	.3	.3	.4	.4	.5	.5
		High-rate transmitting	High-rate transmitting	High-rate transmitting	Relay-to-Ground transmitting	TT&C transmitting	TT&C transmitting	Relay-to-User transmitting	Relay-to-User transmitting	Inter-satellite transmitting	Inter-satellite transmitting
21B1	IP attribute	.1	.2	.3	.4	.5	.6	.7	.8	.9	.10
		Secure	Secure	Secure	Secure	Secure	Secure	Secure	Secure	Secure	Secure
22B1	Internal Elements of Form (IEoF)	.1	.2	.3	.3	.1	.3	.1	.2	.3	.2
		LCT	OISL terminal	ISL terminal	ISL terminal	LCT	ISL terminal	LCT	OISL terminal	ISL terminal	OISL terminal
23B1	IEoF attribute	.1	.2	.3	.4	.5	.6	.7	.8	.9	.10
		Cost	Cost	Cost	Cost	Cost	Cost	Cost	Cost	Cost	Cost

Figure 6.25: Second level decomposition of the EDRS' first module (EDRS satellite)

For example, from Figure 6.25 we may notice that the EDRS satellite is decomposed into the three internal elements of form – "Laser communication terminal (LCT)" (entry 22B1.1), "Ka-band OISL terminal" (entry 22B1.2), and "Ka-band ISL terminal" (entry 22B1.3). There are five internal processes executed by these internal elements of form: "High-rate transmitting" (entry 20B1.1), "Relay-to-ground transmitting" (entry 20B1.2), "TT&C transmitting" (entry 20B1.3), "Relay-to-user transmitting" (entry 20B1.4), and "Intersatellite transmitting" (entry 20B1.5).

The ground segment is decomposed into three internal elements of form (see Figure 6.26): "Mission Operations Center" (MOC) (entry 22B2.1), "Devolved Payload Control Center (DPCC)" (entry 22B2.2), and "Ground station" (entry 22B2.3). They execute the four internal processes: "High-rate transmitting" (entry 20B2.1), "Ground-to-relay transmitting" (entry 20B2.2), "TT&C transmitting" (entry 20B2.3), and "Relay-to-user transmitting" (entry 20B2.4).

17B2	**Internal Operands (IO)**	**.1**	**.1**	**.1**	**.1**	**.1**
		EM signal	**EM signal**	**EM signal**	**EM signal**	**EM signal**
18B2	IO value attribute	.1	.2	.3	.4	.5
		Location	Location	Location	Location	Location
19B2	IO other attribute	.1	.2	.3	.4	.5
		Frequency	Frequency	Frequency	Frequency	Frequency
20B2	**Internal Processes (IP)**	**.1**	**.2**	**.2**	**.3**	**.4**
		Relay-to-User transmitting	**Ground-to-Relay transmitting**	**Ground-to-Relay transmitting**	**TT&C transmitting**	**High-rate transmitting**
21B2	IP attribute	.1	.2	.3	.4	.5
		Secure	Secure	Secure	Secure	Secure
22B2	**Internal Elements of Form (IEoF)**	**.1**	**.2**	**.3**	**.3**	**.3**
		MOC	**DPCC**	**Ground station**	**Ground station**	**Ground station**
23B2	IEoF attribute	.1	.2	.3	.4	.5
		Cost	Cost	Cost	Cost	Cost

Figure 6.26: Second level decomposition of the EDRS' second module (ground station)

The model-based framework for EDRS system is presented in Figure 6.27 that contains the same information as it is in Figures 6.25 and 6.26. From this model-based representation the system architect can extract all data about form to process allocation.

The model-based representation has the same ontology and semantics for both sub-systems, thus we may notice the core information about both sub-systems on the same level of granularity.

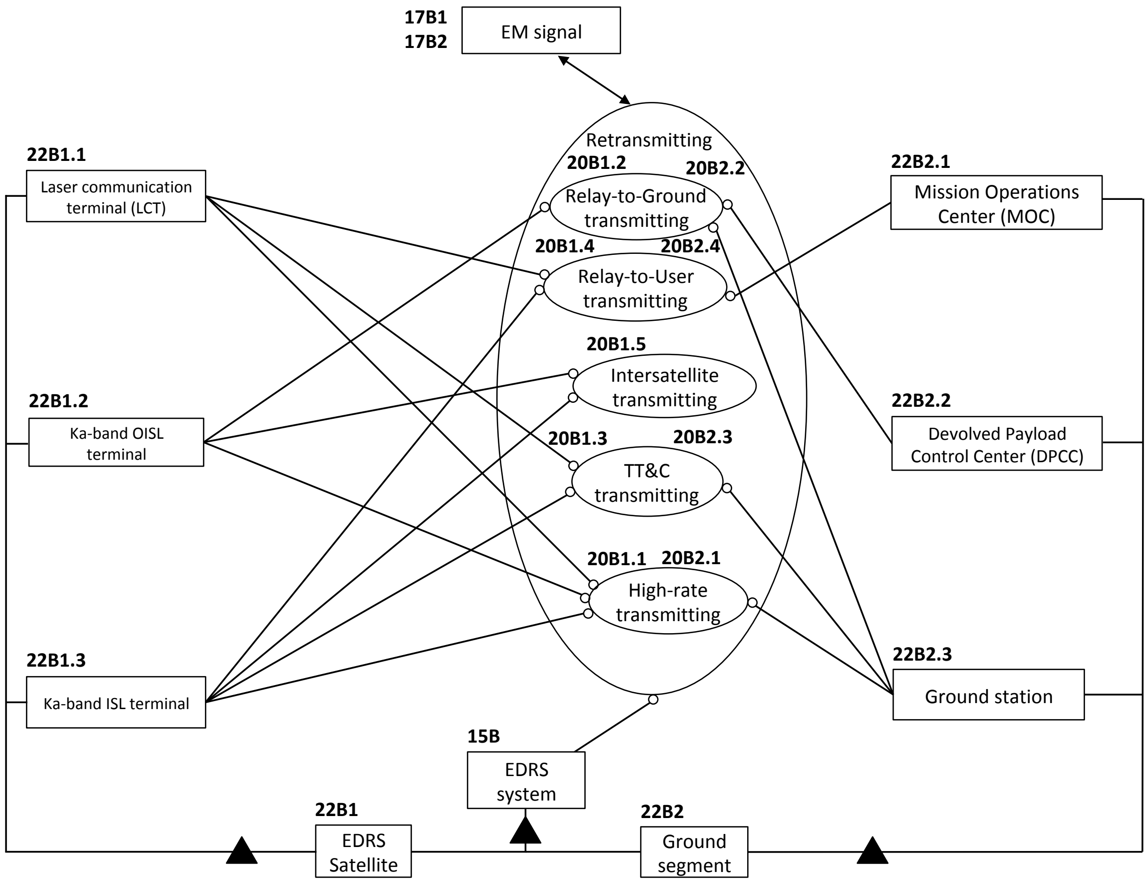
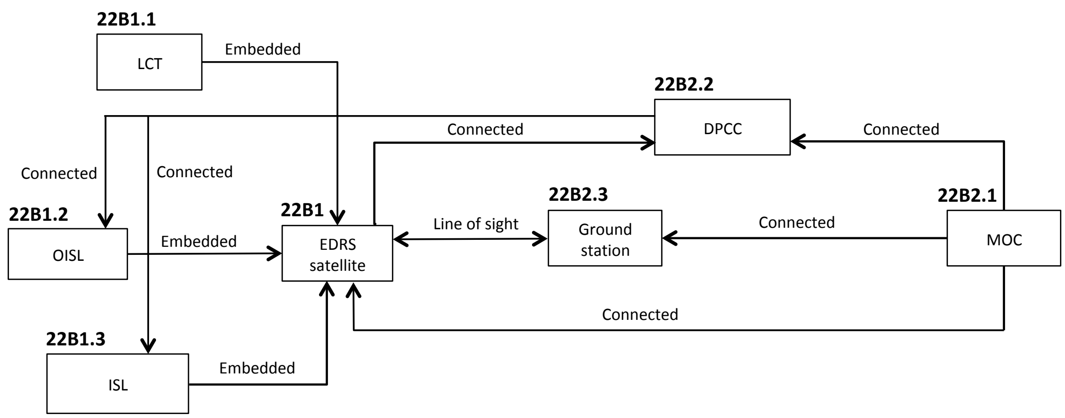


Figure 6.27: Model-based representation of the second level decomposition for EDRS system

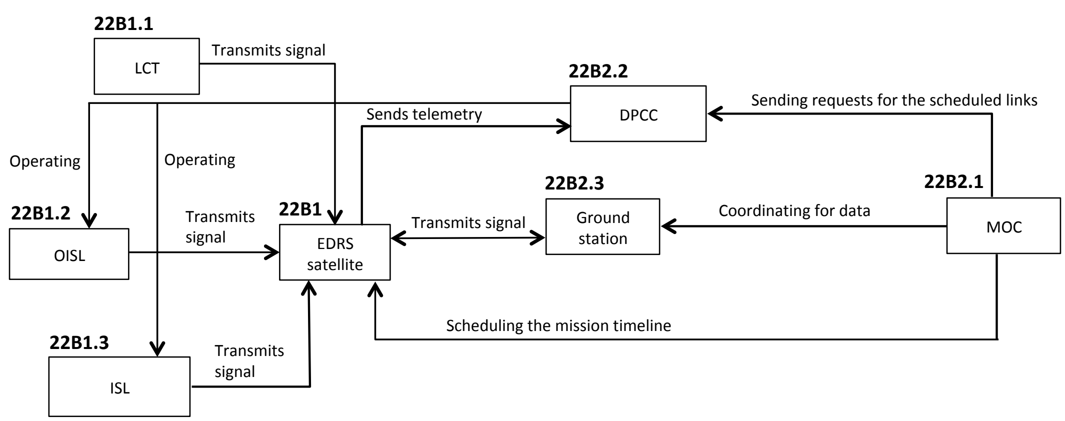
The structure and interactions are presented in Table 6.10 and Figure 6.28. The formal relationships of such elements of EDRS system as EDRS satellite and Ground segment at the first level of decomposition of specific form, and ISL terminal, LCT, OISL terminal, and DPCC, MOC, and Ground station at the second level decomposition are demonstrated in Table 6.10 (table view) and Figure 6.28 (graphical representation).

Table 6.10: Structure and Interactions for EDRSS concept (table format)

24	Structure	ISL **Embedded** EDRS satellite LCT **Embedded** EDRS satellite OISL **Embedded** EDRS satellite EDRS satellite **Line of sight** Ground station EDRS satellite **Connected** DPCC	DPCC **Connected** OISL DPCC **Connected** ISL MOC **Connected** DPCC MOC **Connected** Ground station MOC **Connected** EDRS satellite Ground station **Line of sight** EDRS satellite
25	Interactions	ISL **Transmits signal** EDRS satellite LCT **Transmits signal** EDRS satellite OISL **Transmits signal** EDRS satellite EDRS satellite **Transmits signal** Ground station EDRS satellite **Sends telemetry** DPCC	DPCC **Operating** OISL DPCC **Operating** ISL MOC **Sends requests for the scheduled links** DPCC MOC **Schedules the mission timeline** EDRS satellite MOC **Coordinating for data** Ground station Ground station **Transmits signal** EDRS satellite



(a)



(b)

Figure 6.28: Structure (a) and interactions (b) for the internal elements of form for EDRS system at the second level of decomposition

- Integrated concept for NEN system: second level decomposition

The architectural decisions for NEN system are presented in Figure 6.29 that summarize the information about the most impactful decisions that shape the NEN design.

N	Process	Form	Opt 1	Opt 2
1	Relay-to-Ground Linking	RG Link	Yes	No
2	Relay-to-User Linking	RU Link	Yes	No
3	Intersatellite Linking	IS Link	Yes	No
4	User-to-Ground Linking	UG Link	Yes	No
5	TT&C transmitting	TT&C transmitter	Yes	No
6	High-rate transmitting	High-rate transmitter	Yes	No
7	Low-rate transmitting	Low-rate transmitter	Yes	No
8	Multiple-user transmitting	Multiple access link	Yes	No
9	Storing the information	Storage device	Yes	No

Figure 6.29: Architectural decisions for NEN system

At the first level decomposition we noticed a key difference between NEN and TDRSS and EDRS: NEN doesn't decompose into sub-systems at the first level of decomposition. Thus, we may see the decomposition at the second level, which is presented in this sub-section.

Figure 6.30 contains the information about two internal elements of form (ground antenna and storage device), the internal processes and internal operands on which the processes act.

17C	**Internal Operands (IO)**	**.1**	**.1**	**.1**	**.1**	**.2**
		EM signal	**EM signal**	**EM signal**	**EM signal**	**Information**
18C	IO value attribute	.1	.2	.3	.4	.5
		Location	Location	Location	Location	Location
19C	IO other attribute	.1	.2	.3	.4	.5
		Frequency	Frequency	Frequency	Frequency	
20C	**Internal Processes (IP)**	**.1**	**.2**	**.3**	**.4**	**.5**
		High-rate transmitting	**TT&C transmitting**	**Low-rate transmitting**	**User-to-Ground transmitting**	**Storing**
21C	IP attribute	.1	.2	.3	.4	.5
		Secure	Secure	Secure	Secure	Secure
22C	**Internal Elements of Form (IEoF)**	**.1**	**.1**	**.1**	**.1**	**.2**
		Ground antenna	**Ground antenna**	**Ground antenna**	**Ground antenna**	**Storage device**
23C	IEoF attribute	.1	.2	.3	.4	.5
		Cost	Cost	Cost	Cost	Cost

Figure 6.30: Second level decomposition for NEN system

The model-based representation of the same information is presented in Figure 6.31. The numbers that are assigned to internal operands, processes and forms correspond to the ones presented in Figure 6.30.

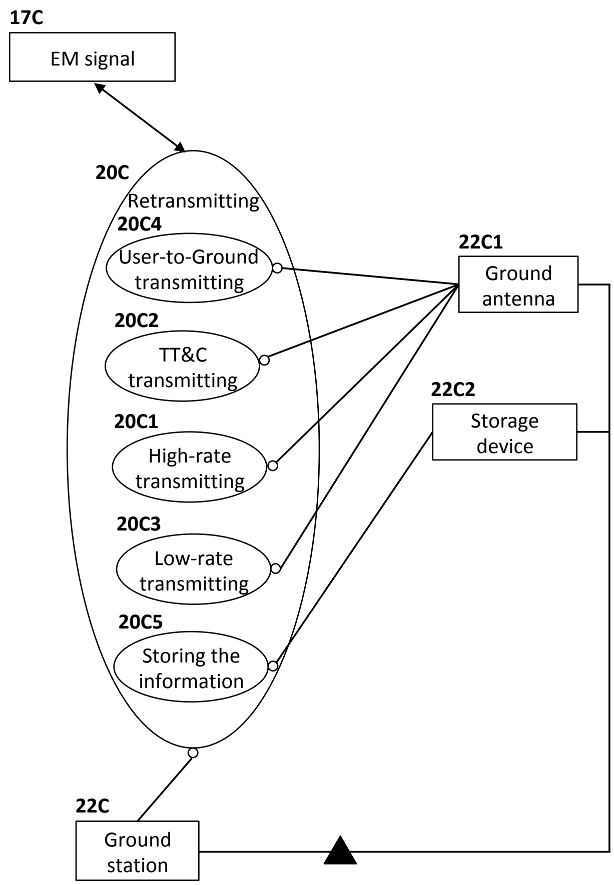


Figure 6.31: Model-based representation of NEN system

The model-based view presented in Figure 6.31 informs us that the NEN is decomposed into two sub-systems: “Ground antenna” (entry 22C1), and “Storage device” (entry 22C2). “Ground antenna” is used for four internal functions: “High-rate transmitting” (entry 20C1), “TT&C transmitting” (entry 20C2), “Low-rate transmitting” (entry 20C3), and “User-to-ground transmitting” (entry 20C4); the “Storage device” is used for internal function “Storing the information” (entry 20C5).

Structure and interactions for NEN system are presented in Table 6.11 and Figure 6.32. Note that NEN is one module system, which implies that there is one level of decomposition of the specific form.

Table 6.11 is the table representation of NEN's structure and interactions information, while Figure 6.32, is the graphical format of the same information on formal arrangement and functional interactions among internal elements of form.

Table 6.11: Structure and Interactions for NEN concept (table format)

24	Structure	Ground antenna **Embedded** Ground station Storage device **Embedded** Ground station
25	Interactions	Ground antenna **Transmits signal** Ground station Storage device **Transmits signal** Ground station

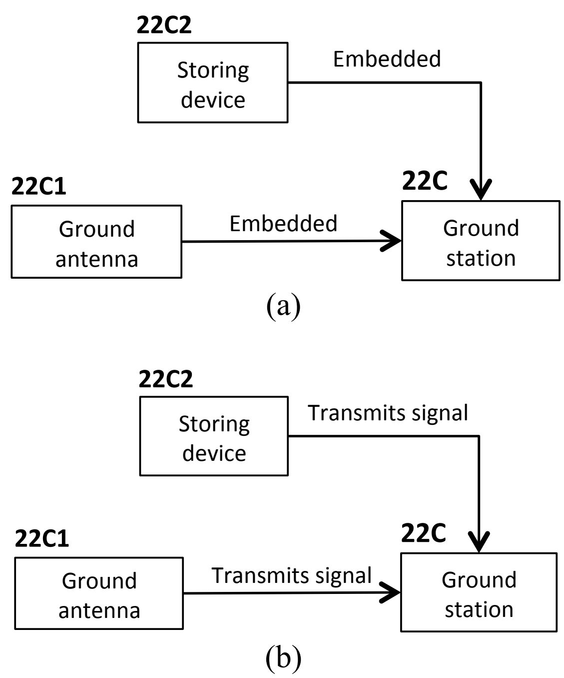


Figure 6.32: Structure (a) and interactions (b) for the internal elements of form for NEN system at the second level of decomposition

6.5.9 Outcomes of the Second Level Decomposition of Alternative Concepts

In this sub-section we will compare the pairs of the concepts' modules based on their functions. For example, the function of the TDRS system's first module (TDRS satellite) is “transmitting EM signal”. The analog of this module in case of EDRS system is EDRS satellite, whose function is also “transmitting EM signal” (See Figure 6.33). The analysis of such a comparison is contained in the Figure 6.34.

Figure 6.33: The first modules of TDRS system (left) and EDRS system (right) concepts

The upper table of Figure 6.34 contains the information about the internal elements of form and internal processes that are related to TDRS satellite. The lower table of Figure 6.34 refers to the EDRS satellite. If we compare them column-by-column, we will see the differences in assigning the internal element of form to the internal process in each alternative, as it is, for example, in case of “Single access antenna” used for high-rate transmitting internal process in case of TDRS satellite, and LCT and OISL terminal used for the same internal process in case of EDRS satellite. From these tables we may also see that, for instance, TDRS satellite has a process “Multiple user transmitting” that does not have a counterpart in case of EDRS satellite.

Overall the representation of Figure 6.34 contains an important information about which exactly internal element of form is responsible for each exactly internal function.

17	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal
20	High-rate transmitting	Multiple user transmitting	Relay-to-Ground transmitting	TT&C transmitting	Relay-to-User transmitting	Relay-to-User transmitting	Low-rate transmitting	Low-rate transmitting	Low-rate transmitting
22	Single access antenna	Multiple access antenna	Relay-to-Ground antenna	Omni antenna	Single access antenna	Multiple-access antenna	Single access antenna	Multiple-access antenna	Omni antenna

17	EM signal	EM signal		EM signal	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal
20	High-rate transmitting	High-rate transmitting		Relay-to-Ground transmitting	TT&C transmitting	Relay-to-User transmitting	Relay-to-User transmitting	Intersatellite linking	Intersatellite linking	Intersatellite linking
22	LCT	OSL terminal		ISL terminal	LCT	LCT	OSL terminal	ISL terminal	LCT	OSL terminal

Figure 6.34: Comparison of first modules of TDRS system and EDRS system

The second module of TDRS system concept is ground station with internal function “Transmitting EM signal”, while the second module of EDRS system concept is ground segment whose internal function is also “Transmitting EM signal”. Both modules are presented in Figure 6.35. The comparison of the second modules of TDRS system and EDRS system is demonstrated in Figure 6.36.

Figure 6.35: The second modules of TDRS system (left) and EDRS system (right)

We may see a bigger number of differences between TDRS system and EDRS system at the comparison of second modules than we saw during the comparison of the first modules. As such, TDRS ground station does not support the process of high-rate transmitting, while in case of EDRS ground station this process is executed by the MOC and DPCC. On the other hand, the TDRS ground station has the low-rate transmitting capability, which does not have a counterpart in case of EDRS ground station. Another important difference is that the EDRS ground station has the intersatellite linking capability executed by the MOC.

17			EM signal	EM signal	EM signal		
20			Ground-to-Relay transmitting	TT&C transmitting	Low-rate transmitting		
22			Ground-to-Relay antenna	Ground-to-Relay antenna	Ground-to-Relay antenna		

17	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal		EM signal	EM signal
20	High-rate transmitting	High-rate transmitting	Ground-to-Relay transmitting	Ground-to-Relay transmitting	TT&C transmitting	TT&C transmitting		Relay-to-User transmitting	Intersatellite transmitting
22	MOC	DPCC	DPCC	Ground station	MOC	DPCC		MOC	MOC

Figure 6.36: Comparison of second modules of TDRS system and EDRS system

In Figure 6.37 the first modules of TDRS system and NEN concepts are presented. Note that NEN has one module, but we will call it as the first module while comparing with TDRS satellite, and the second module during the comparison with TDRS ground station.

Figure 6.37: The first modules of TDRS system (left) and NEN (right) concepts

Figure 6.38 contains the details on comparison of the first modules of TDRS system and NEN. As such we may notice that the TDRS satellite has “Multiple user transmitting”, “Relay-to-ground transmitting”, and “Relay-to user transmitting” internal processes that do not appear in NEN system. In turn, NEN has the internal process “User-to-ground transmitting”, but the core difference is that it has the “Storing” capability allowing the system to store the information rather than relay it in almost real time.

17	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal		
20	High-rate transmitting	Multiple user transmit.	Relay-to-Ground transmit.	TT&C transmitting	Relay-to-User transmitting	Relay-to-User transmitting	Low-rate transmitting	Low-rate transmitting	Low-rate transmitting		
22	Single access antenna	Multiple access antenna	Relay-to-Ground antenna	Omni antenna	Single access antenna	Multiple-access antenna	Single access antenna	Multiple-access antenna	Omni antenna		

17	EM signal			EM signal			EM signal			EM signal	Information
20	High-rate transmitting			TT&C transmitting			Low-rate transmitting			User-to-Ground transmitting	Storing
22	Ground antenna			Ground antenna			Ground antenna			Ground antenna	Storage device

Figure 6.38: Comparison of first modules of TDRS system and NEN

Comparison of the second modules is demonstrated in Figure 6.40. TDRS ground station has one internal element of form that is performing the internal functions “Ground-to-relay transmitting”, “TT&C transmitting”, and “Low-rate transmitting” – ground-to-relay antenna. NEN has two internal elements of form that perform the internal functions “High-rate transmitting”, “TT&C transmitting”, “Low-rate transmitting”, “User-to-ground transmitting”, and “Storing” – ground antenna and storage device.

Figure 6.39: The second modules of TDRS system (left) and NEN (right)

17		EM signal	EM signal	EM signal		
20		Ground-to-Relay transmitting	TT&C transmitting	Low-rate transmitting		
22		Ground-to-Relay antenna	Ground-to-Relay antenna	Ground-to-Relay antenna		

17	EM signal		EM signal	EM signal	EM signal	Information
20	High-rate transmitting		TT&C transmitting	Low-rate transmitting	User-to-Ground transmitting	Storing
22	Ground antenna		Ground antenna	Ground antenna	Ground antenna	Storage device

Figure 6.40: Comparison of second modules of TDRS system and NEN

In Figure 6.41 the first modules for EDRS system and NEN concepts are presented with the same assumption as it was discussed for previous example.

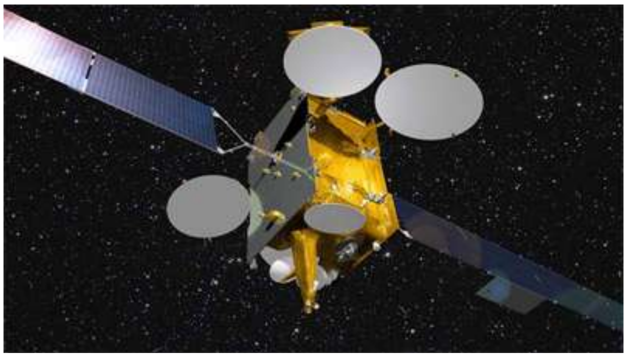
 

Figure 6.41: The first modules of EDRS system (left) and NEN (right) concepts

Figure 6.42 demonstrates the comparison of both modules. We may see that such internal process as "Relay-to-ground" as it appears for EDRS satellite cannot be found for the NEN. In turn, NEN has the "Low-rate transmitting" capability, but the core difference is the presence of "Storing" the information internal process in case of NEN (see the lower table in Figure 6.42).

17	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal			
20	High-rate transmit.	High-rate transmit.	Relay-to-Ground transmit.	TT&C transmit.	Relay-to-User transmit.	Relay-to-User transmit.	Intersat. transmit.	Intersat. transmit.	Intersat. transmit.			
22	LCT	OSL terminal	ISL terminal	LCT	LCT	OSL terminal	ISL terminal	LCT	OSL terminal			

17	EM signal			EM signal						EM signal	EM signal	Information
20	High-rate transmitting			TT&C transmit.						Low-rate transmitting	User-to-Ground transmitting	Storing
22	Ground antenna			Ground antenna						Ground antenna	Ground antenna	Storage device

Figure 6.42: Comparison of first modules of EDRS system and NEN

Figure 6.43 represents the second modules for EDRS system and NEN concepts.

Figure 6.43: The second modules of EDRS system (left) and NEN (right) concepts

From Figure 6.44 we see that the EDRS ground station supports the "Intersatellite transmitting", while there is no such internal process for NEN concept.

17	EM signal	EM signal	EM signal	EM signal	EM signal	EM signal		EM signal	EM signal		
20	High-rate transmit.	High-rate transmit.	Ground-to-Relay transmit.	Ground-to-Relay transmit.	TT&C transmit.	TT&C transmit.		Relay-to-User transmit.	Intersatellite transmit.		
22	MOC	DPCC	DPCC	Ground station	MOC	DPCC		MOC	MOC		

17	EM signal		EM signal	EM signal			EM signal	Information
20	High-rate transmitting		TT&C transmit.	Low-rate transmitt.			User-to-Ground transmitting	Storing
22	Ground antenna		Ground antenna	Ground antenna			Ground antenna	Storage device

Figure 6.44: Comparison of second modules of EDRS system and NEN

6.5.10 Quantifying a Conceptual Similarity Through the DSM-Based Approach

In our work (Menshenin and Crawley 2018a) we have demonstrated how both types of relationships – decomposition and specialization – could be represented in DSM-based methods. These proposals can be effectively used for the conceptual design phase of space communications missions. In Figure 6.45 these types of relationships are captured by the OPM notations – black triangle is used to denote the decomposition, whereas white triangle denotes the specialization relationship.

Figure 6.45 is the full DMM matrix that integrates core design information about 3 alternatives – TDRS, EDRS, and NEN. In particular this matrix contains the data about the formal conceptual similarity among competing alternatives (DSM matrix, left-hand side of Figure 6.45), and the information about the integrated concept (DMM matrix, right-hand side of Figure 6.45).

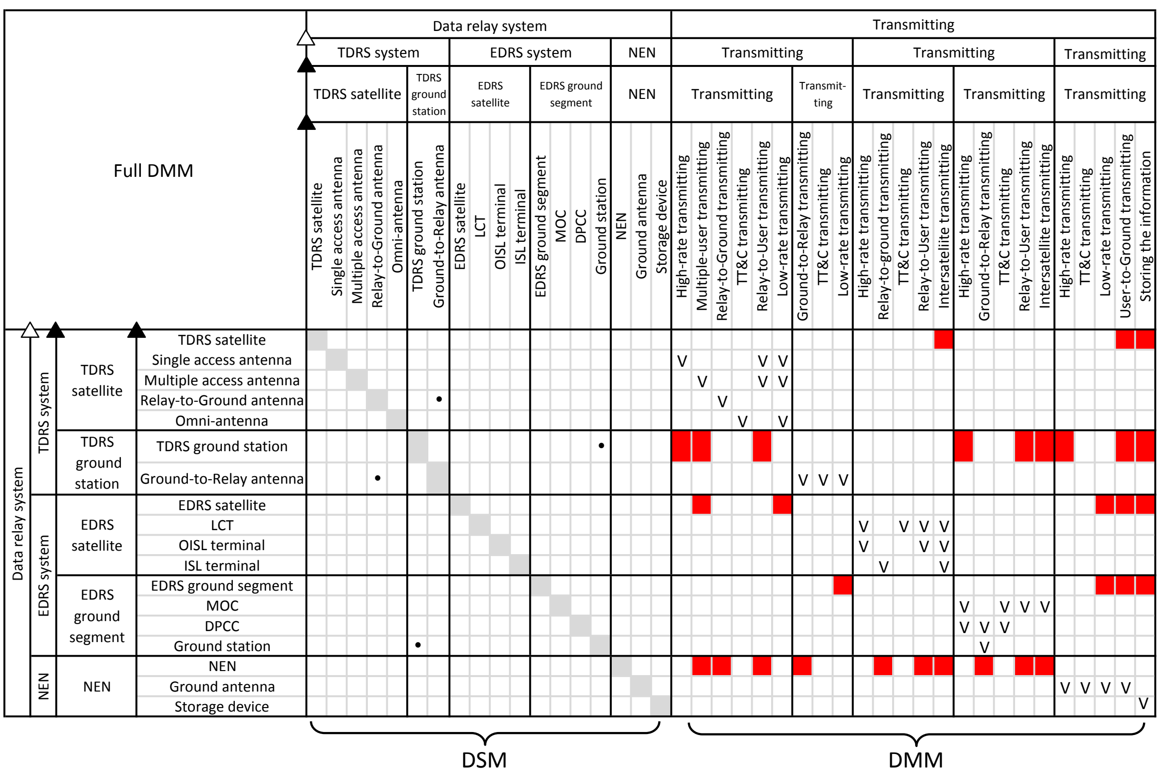


Figure 6.45: Full DMM matrix with the core conceptual design information

The structure and interactions for the first alternative, TDRS system, is demonstrated in Figure 6.46. From it we see that the single access antenna embedded at TDRS satellite (structure), or that the ground station is at line of sight to TDRS satellite (structure). The functional interactions are that the single access antenna transmits signal to TDRS satellite, while ground station transmits signal to TDRS satellite. At the bottom left part of the Figure 6.46 we may notice the information about formal conceptual similarity of the two sub-systems – TDRS satellite and Ground station. For example, both of them have ground-to-relay antenna as the instrument.

Structure and Interactions for TDRS system		22A1	22A1.1	22A1.2	22A1.3	22A1.4	22A2	22A2.1
		TDRS satellite	Single access antenna	Multiple access antenna	Relay-to-Ground antenna	Omni antenna	Ground station	Ground-to-Relay antenna
22A1	TDRS satellite		Embedded Transmits signal	Embedded Transmits signal	Embedded Transmits signal	Embedded Transmits signal	Line of sight Transmits signal	
22A1.1	Single access antenna							
22A1.2	Multiple access antenna							
22A1.3	Relay-to-Ground antenna							•
22A1.4	Omni antenna							
22A2	Ground station							Embedded Transmits signal
22A2.1	Ground-to-Relay antenna				•			

Figure 6.46: Structure, interactions, and conceptual similarity for TDRS system

The same idea is present in Figure 6.47 for the EDRS system. As opposed to the previous concept, the EDRS' sub-systems are completely different: they do not share any identical elements of form. The information about structure and interactions is contained in the intersections of corresponding instruments. For example, MOC is connected to the EDRS satellite (structure) and scheduling the mission timeline for EDRS satellite (interactions).

Structure and Interactions for EDRS system		22B1 EDRS satellite	22B1.1 LCT	22B1.2 OISL terminal	22B1.3 ISL terminal	22B2 Ground segment	22B2.1 MOC	22B2.2 DPCC	22B2.3 Ground station
22B1	EDRS satellite		Embedded Transmits signal	Embedded Transmits signal	Embedded Transmits signal		Connected Scheduling the mission timeline		Line of sight Transmits signal
22B1.1	LCT								
22B1.2	OISL terminal							Connected Operating	
22B1.3	ISL terminal							Connected Operating	
22B2	Ground segment								
22B2.1	MOC								
22B2.2	DPCC	Connected Sends telemetry					Connected Sending requests		
22B2.3	Ground station						Connected Coordinating for data		

Figure 6.47: Structure, interactions, and conceptual similarity for the EDRS system

In Figure 6.48 we present the structure and interactions for the NEN system. We may notice that ground antenna is embedded to ground station (structure), and that storage device transmits signal to ground station (interactions).

Structure and Interactions for NEN project		22C	22C1	22C2
		Ground station	Ground antenna	Stroing device
22C	Ground station		Embedded Transmits signal	Embedded Transmits signal
22C1	Ground antenna			
22C2	Storing device			

Figure 6.48: Structure, interactions, and conceptual similarity for the NEN system

Thus, DSM representations, combined with our methods to include the specialization and decomposition relationships are powerful tools to *encode* the conceptual design information and to *measure* a conceptual similarity between competing alternatives, as well as among sub-systems of specific concept.

6.5.11 Proposition V: Concept of Operations

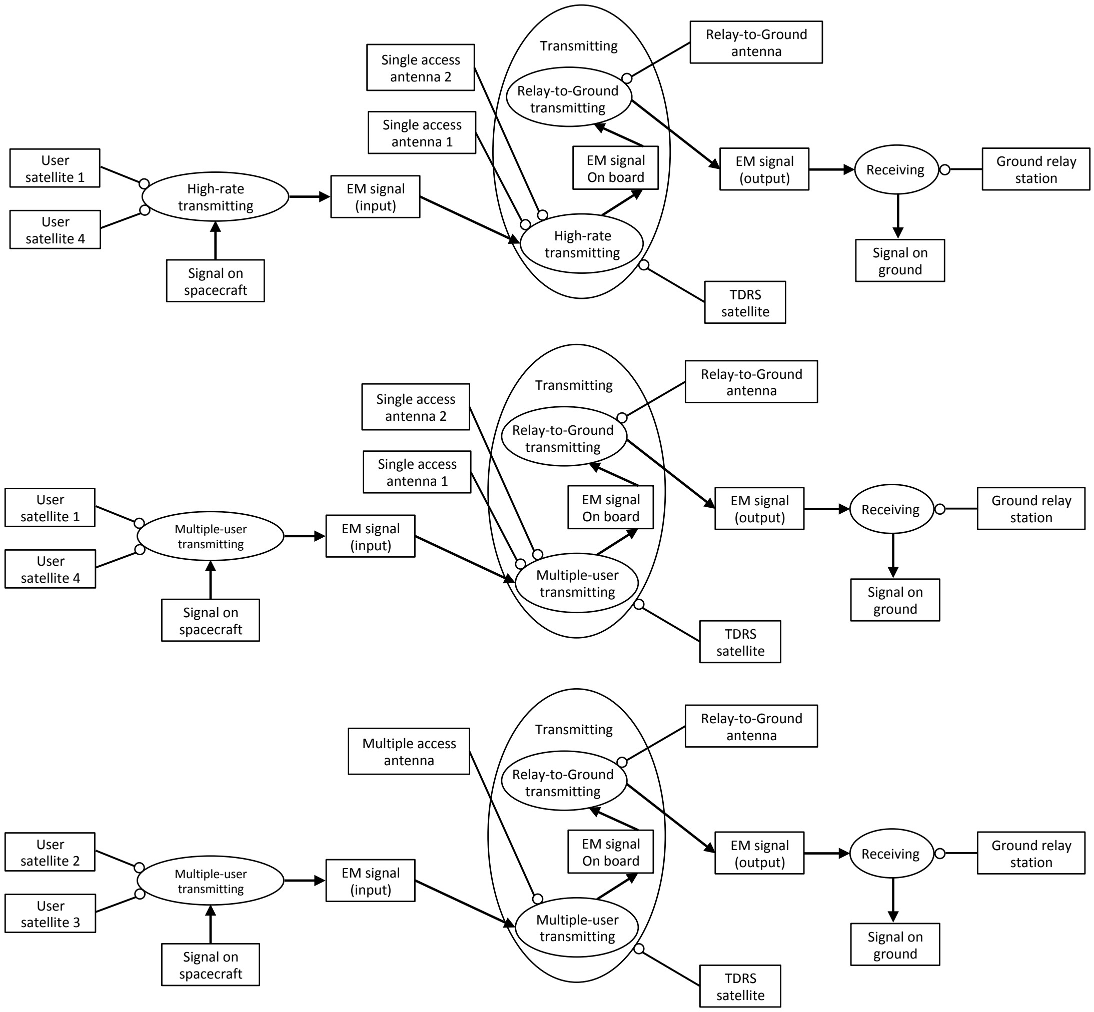
As we have discussed in Chapter 3, the fifth and last proposition of the *system concept representation framework* is the concept of operations. This proposition includes ConOps itself (entry 26), operator (entry 27), and context (entry 28).

This proposition also contains the information about the whole product system, including the system of interest and the context, which can be represented by accompanying systems. System boundary separates the system from the other systems that should be taken into account.

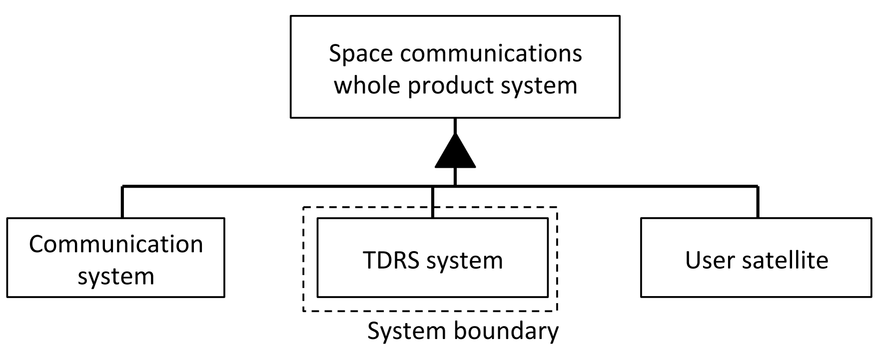
ConOps for TDRS system is presented in Figure 6.49(a) and the context is presented in Figure 6.49(b). From the context diagram we may see that inside the system boundary we have the TDRS system itself, while accompanying systems are communication system, and user satellite.

The Figure 6.49(a) demonstrates the sequence of activities that lead to the delivery of the primary function for examples of high-rate transmitting and multiple-user transmitting internal processes.

Having these two conceptual representations in hands the system architect should be able to understand how the system is intended to operate. The context - Figure 6.49(b) – under which the system operates, is supporting the system architect with identification of accompanying systems.



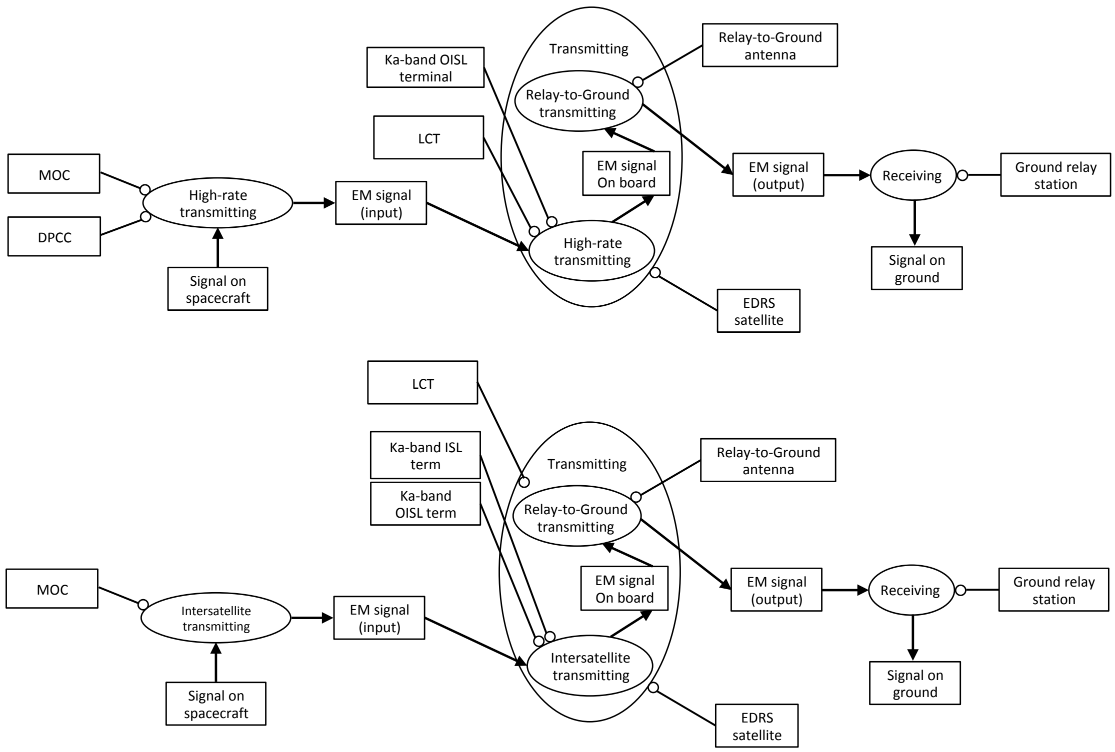
(a)



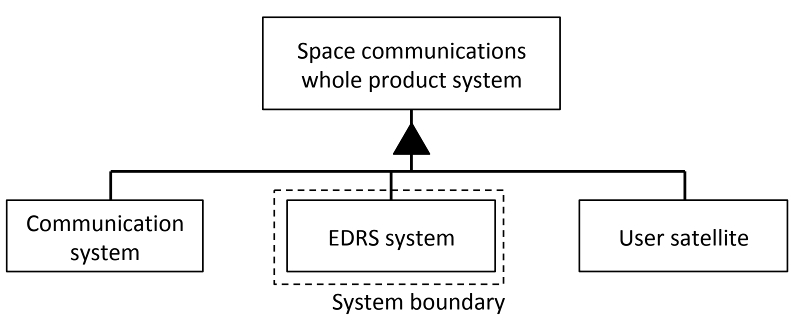
(b)

Figure 6.49: ConOps in OPD view (a), and Context (b) for TDRS system

In Figure 6.50(a) and 6.50(b) the ConOps and Context for EDRS system is presented (in OPD diagrams). The ConOps is presented for two internal processes: high-rate transmitting and intersatellite transmitting. The accompanying systems are communication system and user satellite.



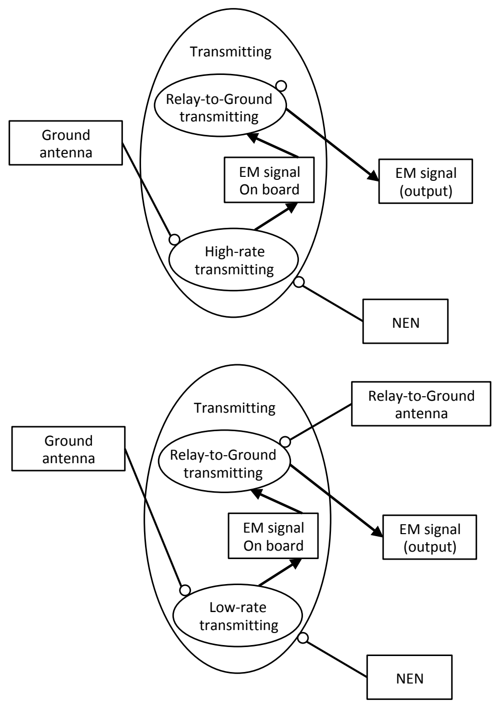
(a)



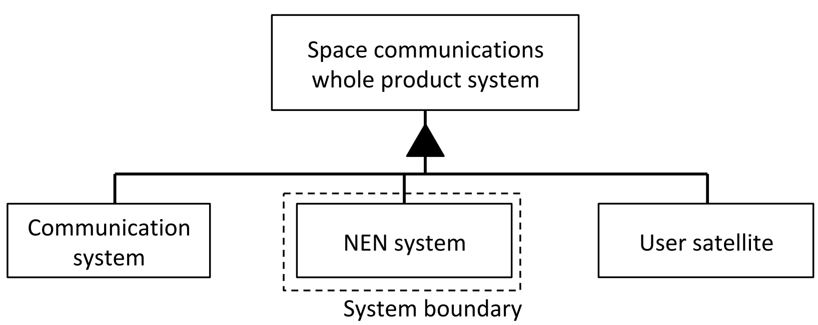
(b)

Figure 6.50: ConOps (a) and Context (b) for EDRS system

In Figure 6.51(a) and 6.51(b) the ConOps and Context for NEN system is presented (in OPD diagrams). The ConOps is demonstrated for the internal processes high-rate transmitting and low-rate transmitting. The accompanying systems are communication system and user satellite.



(a)



(b)

Figure 6.51: ConOps (a) and Context (b) for NEN system

Thus, we see how precise and useful the information contained in the model-based concept frameworks for all three alternative concepts of space communication case study.

6.6 Conceptual Similarity Between Alternative Concepts

Conceptual similarity can be measured based on architectural decisions, demonstrated in Table 6.2. Figures 6.16 and 6.17, 6.23 and 6.24, 6.29 are summarized architectural decisions for TDRS system, EDRS system, and NEN system, respectively.

Using these Figures, the system architect and project team can precisely define the decomposed internal instruments of form and the internal functions that these instruments perform. This information is required for system integration and manufacturing purposes. There are also the details that are necessary not only for technical side, but also economical one. Having the space-based assets as it is in cases of TDRS system and EDRS system require not only operational capabilities, but also space systems manufacturing ones. In case of NEN system this is a terrestrial-based system that does not have any objects placed in space.

Another conceptual difference between these systems is that both – TDRS system and EDRS system – are near real-time data transmission systems, while the NEN system allows storing the information.

Also, there is a key difference between the TDRS system and EDRS system – EDRS has an intersatellite linking capability.

6.7 Interconnections Between Solution Diagrams and Architectural Decisions

The power of the solution diagrams, presented in Figures 6.20 (TDRS system), 6.27 (EDRS system), and 6.31 (NEN system) is that they contain the architectural decisions, shown in Figures 6.16 and 6.17 (TDRS system), 6.23 and 6.24 (EDRS system), and 6.29 (NEN system).

In order to demonstrate the interwoven character of these representations the Figures 6.52 (TDRS system), 6.53 (EDRS system), and 6.54 (NEN system) are shown in this Chapter.

Consider the Figures 6.50-6.52 dedicated to connections between solution diagram and architectural decisions for TDRS, EDRS, and NEN systems. The system architect might see that all 9 decisions, mentioned in Table 6.2 are shown in Figures 6.52 to 6.54.

The importance of having these diagrams is that they contain the core information about the architectural decisions that are needed to be implemented in one or another concept that were explored in this Chapter.

In Figure 6.52 the architectural decisions are mapped into the solution diagram for TDRS system; in Figure 6.53 – into the EDRS system; and in Figure 6.54 – into the NEN system.

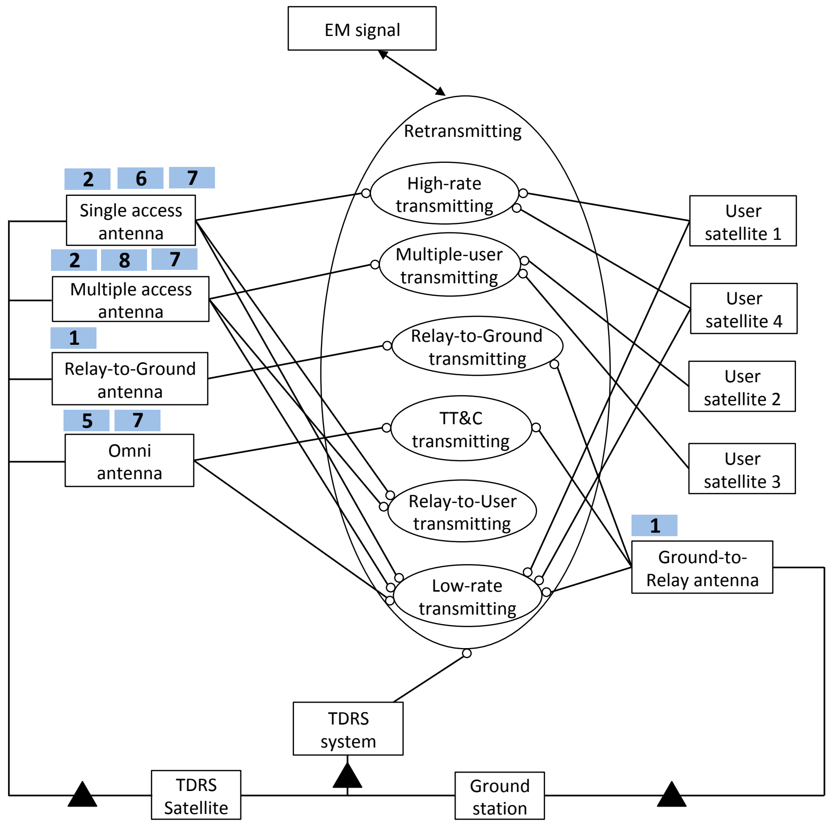


Figure 6.52: Architectural decisions in solution diagram for TDRS system

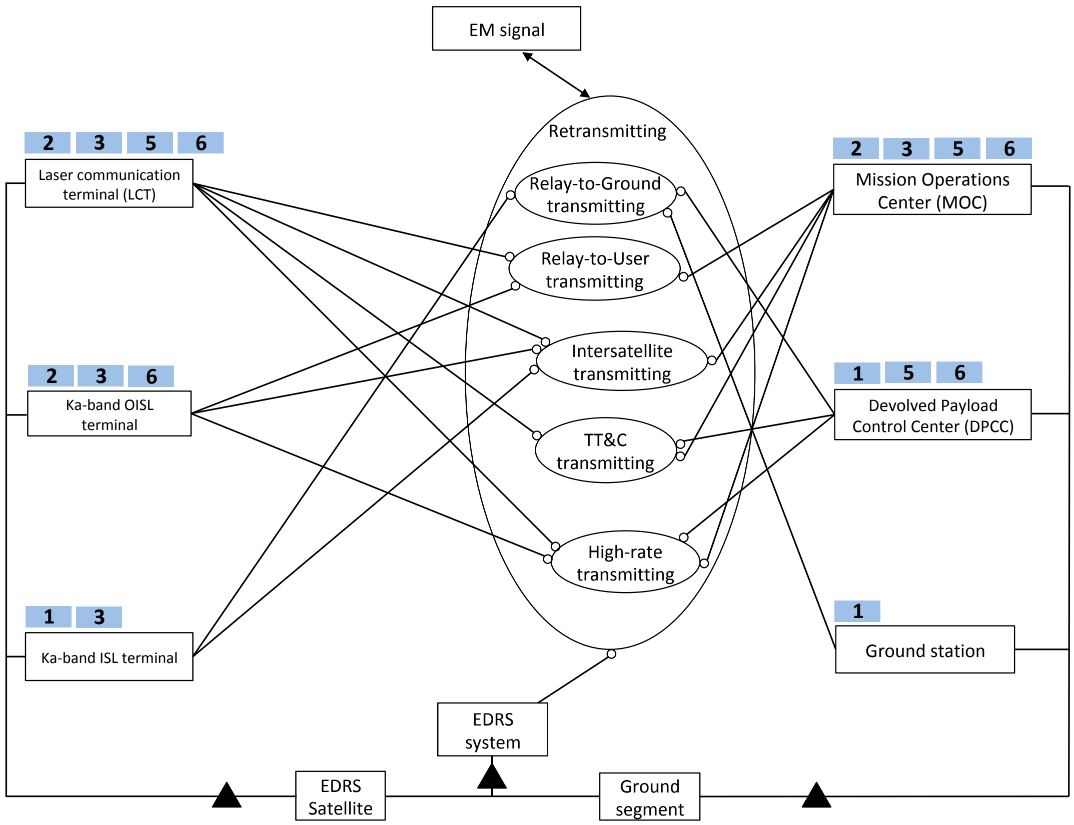


Figure 6.53: Architectural decisions in solution diagram for EDRS system

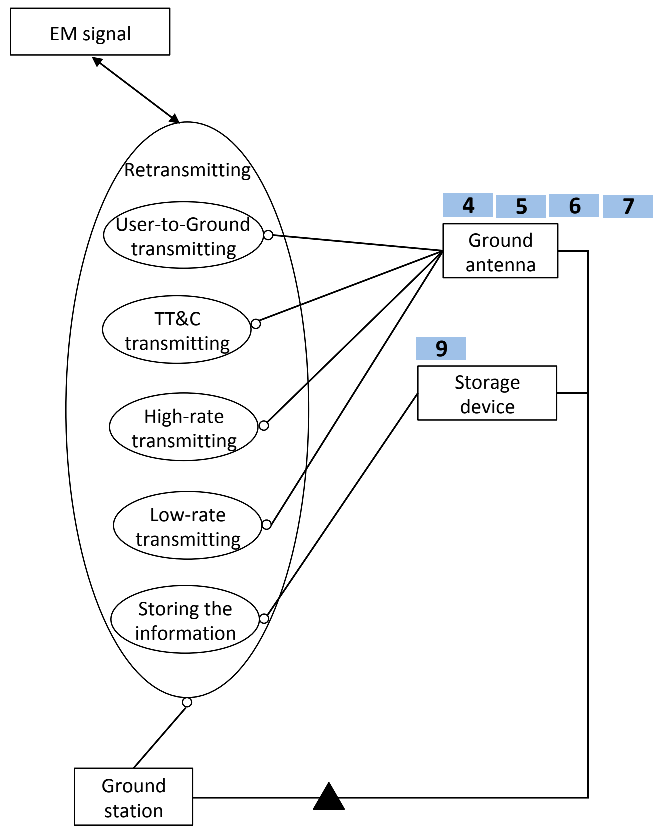


Figure 6.54: Architectural decisions in solution diagram for NEN system

These Figures demonstrate the importance of solution diagram. It contains not only the information about from-function allocation, but also the key architectural decisions that guide system design. Another advantage of these representations is their ability to demonstrate that the architectural decisions could effectively be represented by means of proposed model-based framework.

6.8 Space Communications Case Study Summary and Conclusions

In Chapter 6 we demonstrated the utility of the proposed model-based *system concept representation framework* for conceptual design of space communication systems, such as TDRS system, EDRS system, and NEN. In particular:

- The architectural decisions for space communication missions have been developed and presented in Table 6.2. These architectural decisions are applicable to any system, which is intended to operate as space communication system (in terms of data relay systems). In Chapter 6 we applied these architectural decisions to three above-mentioned alternative projects. We presented the chosen architectural decisions for each of these projects in Figures 6.16-6.17, 6.23-6.24, 6.29, respectively;
- We *encoded* a conceptual information about all 3 alternative concepts of space communication missions into proposed model-based framework comprised of 28 entries. These entries are spread among different parts of framework: stakeholders and their needs; solution-neutral environment (problem statement); solution-specific environment (solution statement); integrated concepts at different levels of granularity, including structure and interactions; concept of operations, and context;
- The two representations of concept frameworks are developed for each project in a table view and in a model-based format based on a conceptual modeling language – Object-Process Methodology (OPM). We demonstrated that both approaches are supporting each other and are interconnected by semantics: Object-Process Language (OPL) has a strict notation, which follows the Object-Process Diagrams (OPD);

- We demonstrated how the proposed approach supports a *formal analysis*, such as a conceptual similarity assessment. In particular, we have shown that the DSM-based approach quantifies the number of internal elements of form, or the number of internal processes that are the same among two alternative concepts under comparison;
- The interconnections between concept framework's solution diagrams and architectural decisions are demonstrated, which allows to consider the methodology as a united framework to develop space communication missions.

Thus, in Chapter 6 we demonstrated the utility of the proposed model-based *system concept representation framework* for space communication missions. The Chapter 6 corresponds to the Descriptive Study II, according to the DRM framework that is used in this Thesis.

Chapter 7. Conclusions

7.1 Thesis Summary

This Thesis presents a model-based *system concept representation framework* that can systematically represent the concept's constituents, their definitions and interconnections. Such a framework would support the design process during the conceptual design phase and would contribute to the INCOSE's Model-Based Conceptual Design Initiative.

Throughout the Thesis we have tested the following hypotheses. The first of them is that the proposed *system concept representation framework* contains a necessary information to describe the system concept. Another hypothesis of our work is that such information can be encoded in a model-based manner to represent system concepts and their alternatives in a digital environment. The third hypothesis is that having such a framework supports design studies in terms of quantitative assessment of formal conceptual similarity between alternative concepts.

In this Thesis we also demonstrated the utility of the proposed *system concept framework*. To do so we have chosen the set of socio-technical systems and societal challenges (disclosed in set of analytical surveys – patents, urban architectural patterns, and software patterns); and purely technical systems (disclosed in two case studies – commercial suborbital human spaceflight systems and space communication missions).

The premises of the *system concept representation framework* development were discussed in Chapter 2, in which we provided an overview of what have already been developed and proposed in systems engineering and design literature. We have also provided the rationales for research opportunities that emerge in the intersection of four pillars: systems modeling, systems engineering, systems architecture, and design theory. In this Thesis we have used the Design Research Methodology (DRM). In its terminology, Chapter 2 covered the Research Clarification stage, as well as the Descriptive Study I.

The model-based *system concept representation framework* has been presented in Chapter 3. In this Chapter we presented 5 propositions of the concept framework: (I) the stakeholders, (II) the solution-neutral problem statement, (III) the solution-specific solution statement, (IV) the integrated concept, and (V) the concept of operations (ConOps). We demonstrated the model-based representation of each one of the propositions on example of aircraft concept development. It has two alternative solutions: tube and wing aircraft concept and blended wing body aircraft concept. We demonstrated that both conceptual modeling languages – SysML and OPM – could be used for the modeling purposes. To demonstrate the utility of the framework we have also applied it to a coffee maker example. The role of Chapter 3, cording to the DRM terminology, was to facilitate the Prescriptive Study.

In Chapter 4 we validated the proposed framework by means of applying the analytical surveys to it. As the analytical surveys we have chosen the wide variety of systems disclosed in patents, urban architectural patterns (Alexander 1977), and software patterns (Gamma et al. 1995). In Chapters 5 and 6 we applied the proposed framework to the suborbital human spaceflight systems (Virgin Galactic, Blue Origin, and XCOR) and space communication systems (TDRSS, EDRS, and NEN), respectively. These three Chapters (Chapter 4, Chapter 5, and Chapter 6) covered the Descriptive Study II, according to the DRM framework.

7.2 Thesis Contributions

This dissertation makes the contribution in two dimensions. The first is the theoretical and methodological contributions, which is more dealing with the contribution to the methodology of system concept development, its implications on what is specified and how this is specified at the conceptual design phase. This part is discussed in sub-section 7.2.1. The second contribution is the practical one, presented in sub-section 7.2.2. In that discussion we explain what are the practical outcomes of applying the proposed model-based *system concept representation framework* to the analytical surveys (patents, urban architectural patterns, and software patterns), and two case studies: suborbital human spaceflight systems (Virgin Galactic, Blue Origin, and XCOR), and space communication systems (TDRSS, EDRS, and NEN).

7.2.1 Theoretical and Methodological Contributions

There are numbers of theoretical and methodological contributions of this Thesis. In our work we presented a *system concept representation framework* comprised of 5 propositions – the rationale to include each of these propositions into the framework is provided in Chapter 3 and is based on the heritage of systems engineering and design science disciplines. We presented the definitions and essence of each one of the 28 entries of the framework. A clear contribution is that the proposed structured approach serves as a tool for a system engineer: having such a tool he or she can *encode* the core information about system concept and its alternatives; and to *generate* the novel system concepts.

Another contribution of our work is that we advanced the usage of the Design Structure Matrix (DSM). We introduced the way of keeping both types of relationships – specialization and decomposition – in one matrix. This enables keeping track of concept's hierarchy, as well as allows to quantitatively compare the alternative solutions.

This novel method has been applied to each case study presented in this dissertation – suborbital human spaceflight systems and space communication systems.

Another theoretical outcome of our work is that the proposed framework could be used to support the decision makers by identifying all alternative solutions that might be emerged based on the expectations of stakeholders. The frameworks themselves contain the information about architectural decisions that should be taken into account at the conceptual design phase. The ability to *encode* this information in a digital environment is an important feature enabling cross-functional multinational teams working together on complex projects regardless the geographical location and language.

7.2.2 Practical Contributions

The proposed *system concept framework* was applied to a number of analytical surveys – patents, urban architectural patterns (Alexander 1977) and software patterns (Gamma et al. 1995). These quite different analytical surveys were chosen with the assumption that they should contain the core information about the system or pattern they represent.

For each analytical survey we have conducted two types of studies: small-N analysis and large-N analysis. During the small-N analysis for patents we focused on four quite different types of patents: biological, thermodynamic, electro-mechanical, and software. In total we analyzed 8 randomly selected US patents. After this we conducted the large-N analysis by mapping 25 selected patents to the concept framework. These 25 patents were chosen out of 85 in Google Patents using the keywords "suborbital spaceflight vehicle". We chose those 25 patents that are most cited ones. Thus, in total around 30% of the full set of patents were analyzed during the large-N study.

The urban architectural patterns (Alexander 1977) are represented by three types: towns, buildings, and construction. In choosing the samples we have focused only on the patterns with two asterisks meaning that they are the most developed (85 patterns out of 253). For the small-N analysis of architectural patterns we focused on 3 samples per each type, totaling 9 architectural patterns. For the large-N analysis we chose 8 samples per each type, totaling 27 patterns. Thus, in total 36 architectural patterns were analyzed. Thus, around 42% of architectural patterns were analyzed during both studies.

In total 23 software patterns are contained in the work of Gamma et al. (1995). These patterns are one of the three types: creational, structural, and behavioral. During the small-N study we chose 1 pattern from each type – overall 3 patterns. For the large-N analysis 4 patterns per type were chosen, resulting in 12 patterns. Thus, around 65% of software patterns were chosen for both types of analysis.

The outcomes of the analytical survey studies are summarized in Figure 4.11. It is shown that such key entries of the *system concept framework* as stakeholders and their needs, solution-neutral/solution-specific operands and processes, as well as the generic/specific forms and the context are always present in all three analytical surveys. There is some rationale of why we always find such entries of concept framework as operands, processes, and forms in the patents/patterns. In order to convey the essence of either patent or pattern this information is vital, as operand-process-form structure is very close to the conventional sentences structure: noun-verb-noun. This is due to the fact that without this core data one cannot claim that he or she has a concept of patent, or a concept of architectural pattern, or a concept of software pattern. Another result of the analytical surveys is that the attributes do not always present in the patents or patterns. The explanation to this is that the adjectives (attributes) are not always used in the sentences, the adjectives are optional. These outcomes largely correspond with each other.

The core difference between these analytical surveys is that both patents and software patterns contain the information about the integrated concept, while the architectural patterns does not. There is some rationale that we do not see the integrated concept for architectural patterns. These patterns deal with the societal problems – a very abstract set of issues. The stated problems in architectural patterns are so broad that the identification of a specific form is already a solution. It should also be noted that for the purpose of our study we have only focused on analysis of the text.

We also see very strong results in regards to the integrated concept for patents. It has a vital role in case of patents, because as we have shown in our study, the claims are represented by the integrated concept. Having the concept framework in hands, the system engineer could engage a digital environment to convey the same amount of information in a much smaller volume of text.

The results of the analytical surveys studies mean that the proposed concept framework serves the purpose of either facilitating the development of a new patent or pattern, or the analysis of existing ones – depending on the system engineer's needs.

Having the concept framework in the toolset, the system engineer could develop the new system following a rigorous approach based on the design heritage.

In Chapter 5 we applied the proposed framework to the suborbital human spaceflight systems. In particular, we analyzed the Virgin Galactic, Blue Origin, and XCOR projects, building the model-based concepts for each one of them on different levels of granularity keeping the same ontology and semantics.

In particular, we identified and presented the architectural decisions that form the basis for suborbital human spaceflight systems – these decisions are presented in Table 5.1. We *encoded* the models for each concept at the first level of decomposition – see sub-section 5.5.6; and at the second level decomposition – see sub-section 5.5.8.

Based on the novel approach introduced in this Thesis, we demonstrated how the specialization and decomposition relationships could be represented in one DSM/DMM matrix (see Figure 5.14 for the first level decomposition and Figure 5.43 for the second level decomposition) to support the identification of conceptual similarity among competing alternative concepts. In Figures 5.44-5.46 we have also demonstrated that DSM is an appropriate tool to represent the structure and interactions.

In Figures 5.50-52 we demonstrated that the proposed approach could also serve as an appropriate tool to represent the architectural decisions in a model-based manner.

In Chapter 6 the proposed model-based *system concept representation framework* has been applied to the space communication systems. Particularly, we analyzed the TDRS system, EDRS system, and NEN system, building the model-based concepts for each one of them on different levels of granularity keeping the same ontology and semantics.

The architectural decisions have been developed and presented in Table 6.2. These nine decisions are the basis for space communication systems. We *encoded* the models for each concept at the first level of decomposition – see sub-section 6.5.6; and at the second level decomposition – see sub-section 6.5.8.

Following the proposed approach to specialization and decomposition by means of DSM methods, we demonstrated the different levels of granularity for the space communication systems (see Figure 6.15 for the first level decomposition and Figure 6.45 for the second level decomposition). This supports the identification of conceptual similarity among competing alternative concepts. In Figures 6.46-48 we applied DSM to representation of the structure and interactions.

In Figures 6.52-6.54 we demonstrated that the proposed approach could also be used as an appropriate tool to represent the architectural decisions in a model-based manner.

7.3 Limitations and Future Work

This work has some limitations. In this work, the information from analytical surveys was mapped into the proposed framework by means of human reasoning supported by a clear definitions and criteria. However, such mapping could be done with textual analysis by means of machine learning. Thus, one of the directions of future work would be the exploration of knowledge description in a machine-accessible way.

Another limitation of our work is that we have focused on the representation of the specialization and decomposition relationships in DSM. However, we could engage other structural relationships (such as exhibition and instantiation), as they defined in OPM. Moreover, the future studies will include not only structural relationships, but also the procedural ones. Thus, further integration of OPM and DSM and potential generation of a DSM from OPCloud (Dori et al. 2019) could lead to fruitful results.

Another direction of future work is to test the proposed *system concept representation framework* on industrial and commercial case studies. In particular, it would be valuable to involve system engineering architects and practitioners into the framework to get their feedback. Such work would engage close partnerships with Industry 4.0 and commercial ventures.

Bibliography

Airbus, 2017. Le Garage: home to innovation. Available at: https://www.airbus.com/newsroom/news/en/2017/06/le-garage-home-to-innovation.html (accessed 10.05.2019)

Al-Qaneei, J.M.A.J., 2016. Vehicle mounted traffic light and system, US Patent 9318021

Aldrin, B. and Davis, H.P., Starcraft Boosters Inc, 2003. Flyback booster with removable rocket propulsion module. US Patent 6,612,522

Alexander, C., 1977. *A pattern language: towns, buildings, construction*. Oxford university press.

Alexander, C., 1964. *Notes on the Synthesis of Form* (Vol. 5). Harvard University Press.

Altshuller, G., 1999. *The innovation algorithm: TRIZ, systematic innovation and technical creativity*. Technical innovation center, Inc.

Altshuller, G., 1996. An*d suddenly the inventor appeared: TRIZ, the theory of inventive problem solving*. Technical Innovation Center, Inc.

Andreasen, M.M., Hansen, C.T. and Cash, P., 2015. *Conceptual design: Interpretations, Mindset and Models*. Cham: Springer International Publishing.

Andreasen, M.M., Wallace, K. 2011. Reflection on design methodology research. *Keynote speech at ICED11*, Copenhagen

Andreasen, M.M., 1992. The Theory of Domains. E*DC-Workshop: Understanding Function and Function to Form Evolution*, Cambridge, pp. 21-47.

Antoniou G., Van Harmelen F. Web ontology language: OWL. *In Handbook on ontologies*. 2004; pp. 67-92. Springer, Berlin, Heidelberg

Bandecchi, M., Melton, B. and Ongaro, F., 1999. Concurrent engineering applied to space mission assessment and design. *ESA bulletin*, 99(Journal Article).

Baucon, A., 2010. Leonardo da Vinci, the founding father of ichnology. *Palaios*, 25(6), pp.361-367.

Blessing, L.T. and Chakrabarti, A., 2009. *DRM: A Design Reseach Methodology* (pp. 13-42). Springer London.

Bryson, J.M., Cunningham, G.L. and Lokkesmoe, K.J., 2002. What to do when stakeholders matter: The case of problem formulation for the African American Men Project of Hennepin County, Minnesota. *Public Administration Review*, 62(5), pp.568-584.

Cervisi, R.T., Toliver, D.M., Greenberg, H.S., Kilgore, T.R., Arnold, J.H. and Blake, J.C., Boeing North American Inc, 1995. Reusable flyback satellite. US Patent 5,402,965

Chebakov, M.S., Maslov I.S., Kaspersky Lab ZAO, 2016. System and method for launching a browser in a safe mode, US Patent 9292701

Chomsky N, 1956. Three models for the description of language. *IRE Transactions on information theory*; 2(3), pp.113-124.

Christensen, C.M., 2013. *The innovator's dilemma: when it comes to new technologies*. Harvard Business Review Press.

Crawley, E., Cameron, B. and Selva, D., 2015. *System architecture: strategy and product development for complex systems*. Prentice Hall Press.

Criswell, D.R., 1989. Aerospace vehicle having multiple propulsion systems on a relatively rotatable flying wing, US Patent 4836470.

Davidian, K. and Conrad, C., 2012. Suborbital research market industry structural analysis. *In Next Generation Suborbital Researchers Conference*, San Jose, California.

Davidian, K. and Foust, J., 2010. Suborbital Market Overview and Application of DIsruption Theory, The 61st International Astronautical Congress, Prague, Czech Republic.

de Weck, O., 2015. Fundamentals of Systems Engineering. *Course at the Massachusetts Institute of Technology*, from https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-842-fundamentals-of-systems-engineering-fall-2015/.

Department of Defense Architecture Framework, 2017. *OV-1: High Level Operational Concept Graphics*. Available at: http://dodcio.defense.gov/Library/DoD-Architecture-Framework/dodaf20_ov1/ (accessed 29.11.2017).

Deubzer, F. and Lindemann, U., 2009. MDM application to interrelate hierarchical layers of abstraction. *In DSM 2009: Proceedings of the 11th International DSM Conference*, Greenville, SC, USA, pp. 167-178.

Dori, D., Kohen, H., Jbara, A., Wengrowicz, N., Lavi, R., Soskin, N.L., Bernstein, K. and Shani, U., 2019. OPCloud: An OPM Integrated Conceptual-Executable Modeling

Environment for Industry 4.0. *Systems Engineering in the Fourth Industrial Revolution*, pp. 243-271.

Dori, D., Reinhartz-Berger, I. and Sturm, A., 2003. Developing complex systems with object-process methodology using OPCAT. *In International Conference on Conceptual Modeling* (pp. 570-572). Springer, Berlin, Heidelberg.

Dori, D., 2002. *Object-process methodology: a holistic systems paradigm*. Springer.

Dressler, G.A., Northrop Grumman Space and Mission Systems Corp, 2000. Non-propellant fluid cooled space craft rocket engine. US Patent 6,052,987.

Eden, C., Ackermann, F., 1998. *Making strategy: the journey of strategic management*, SAGE, London. http://dx.doi.org/10.4135/9781446217153

Eppinger, S.D., Browning, T.R., 2012. *Design structure matrix methods and applications*, MIT Press.

Eppinger, S.D. and Ulrich, K.T., 1995. *Product design and development (4th Edition)*. McGraw-Hill Education Singapore.

ESA, 2016. Start of service for Europe's SpaceDataHighway. Available at: http://www.esa.int/Our_Activities/Telecommunications_Integrated_Applications/EDRS/Start_of_service_for_Europe_s_SpaceDataHighway (accessed 10.05.2019)

ESA. EDRS. Available at: https://www.esa.int/Our_Activities/Telecommunications_Integrated_Applications/EDRS/Overview/ (accessed 10.05.2019)

Ferreirinha, P., Grothe-Moller, T., Hansen, C.T., 1990. TEKLA, a language for developing knowledge based design systems, *In Proceedings of ICED 1990, Dubrovnik*, The Design Society, pp. 1058-1065.

Freeman, R.E., McVea, J., 2001. *A stakeholder approach to strategic management*, Darden Business School Working Paper No. 01-02. http://dx.doi.org/10.2139/ssrn.263511

Freeman, R.E., 1984. *Strategic management: A stakeholder approach*, Boston: Pitman.

Frey, D.D. and Dym, C.L., 2006. Validation of design methods: lessons from medicine. *Research in Engineering Design*, 17(1), pp.45-57.

Friedenthal, S., Moore, A. and Steiner, R., 2014. *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J., 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley. Reading, MA.

Golkar, A., 2016. Lessons learnt in the development of a Concurrent Engineering Infrastructure. *In INCOSE International Symposium* (Vol. 26, No. 1, pp. 1759-1769).

Golovanov, S.Y. and Monastyrsky, A.V., Kaspersky Lab ZAO, 2016. System and methods for detection of fraudulent online transactions, US Patent 9363286

Grillos, E., Trimble, D.A. and Grose, D.L., 2015. Composite structures for aerospace vehicles, and associated systems and methods. US Patent 9,079,674

Guerster, M. and Crawley, E.F., 2018. Architectural options and optimization of suborbital space tourism vehicles. In 2018 IEEE Aerospace Conferenc, pp. 1-18.

Hallenberg, J., 2004. The Wright Brothers, the Invention of the Aerial Age. *National Air and Space Museum*, Smithsonian, http://nasm. si. edu.

Hatchuel, A. and Weil, B., 2009. CK design theory: an advanced formulation. *Research in engineering design*, 19(4), p.181-192.

Hatchuel, A. and Weil, B., 2003. A new approach of innovative design: an introduction to CK theory. *In DS31: Proceedings of ICED 03, the 14th International Conference on Engineering Design,* Stockholm.

Held T., Hoster S., Miller J., Hume B, Echogen Power Systems, 2012. Heat engine and heat to electricity systems and methods, US 8096128

Holt, J. and Perry, S., 2008. *SysML for systems engineering* (Vol. 7). IET.

Hickey, E.S., 1997. Space vehicle apparatus including a cellular sandwich with phase change material, US Patent 5669584.

Holzwanger, M., Mackey, J., Liu, X.H., Cohen, S., Holzwanger, H. and Guensler, R., Holzmac Llc, 2015. Traffic signal device for driver/pedestrian/cyclist advisory message screen at signalized intersections, US Patent 9153128

Hoyt, R.P. and Forward, R.L., 2002. Failure resistant multiline tether. US Patent 6,431,497.

Hoyt, R.P. and Forward, R.L., 2002. Failure resistant multiline tether. US Patent 6,386,484.

Hoyt, R.P. and Forward, R.L., 2002. Electrodynamic tether control. US Patent 6,419,191.

Hoyt, R.P. and Forward, R.L., 2001. Planar hoytether failure resistant multiline tether. US Patent 6,290,186.

Hoyt, R.P. and Forward, R.L., 2001. Failure resistant multiline tether. US Patent 6,260,807.

Hoyt, R.P. and Forward, R.L., 2001. Failure resistant multiline tether. US Patent 6,173,922

Hubka, V., Eder, W., 1988. *Theory of Technical Systems: A Total Concept Theory for Engineering Design*. Springer-Verlag, New York

Hubka, V. 1973. *Theorie der Maschinensysteme (Theory of machine systems)*. Springer, Berlin.

INCOSE. Model-Based Conceptual Design Working Group. http://www.incose.org/ChaptersGroups/WorkingGroups/transformational/model- based-conceptual-design

INCOSE. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. John Wiley & Sons.

International Organization for Standardization, 2015. Automation systems and integration – Object-Process Methodology. Accessed 29 November 2017 https://www.iso.org/standard/62274.html

Le Masson, P., Dorst, K. and Subrahmanian, E., 2013. Design theory: history, state of the art and advancements, *Research in Engineering Design*.

Leonard, B.P., 1992. Multistage launch vehicle employing interstage propellant transfer and redundant staging. US Patent 5,129,602

Lopata, J.B. and Kamel, M.R., Space Launch Corp, 2005. System for the delivery and orbital maintenance of microsatellites and small space-based instruments. US Patent 6,921,051

Jensen, T., 1999. Functional Modelling in a Design Support System – Contribution to a Designer's Workbench. *PhD thesis*, Technical University of Denmark.

Kennedy, J.F., 1961. Excerpt from the 'Special Message to the Congress on Urgent National Needs'.

Kroll E., 2013. Design theory and conceptual design: contrasting functional decomposition and morphology with parameter analysis. *Research in Engineering Design*, Vol. 24, pp. 165-183.

Kruchten, P., 2008. What do software architects really do? J*ournal of Systems and Software*, 81(12), pp.2413-2416.

Leonard, 1992. Multistage launch vehicle employing interstage propellant transfer and redundant staging. US Patent 5,129,602.

Lopata, J.B. and Kamel, M.R., 2005. System for the delivery and orbital maintenance of micro satellites and small space-based instruments. U.S. Patent 6,921,051.

Lewis D., Schmidt P., Hinds K., PR Pharmaceuticals, 2004. Methods and compositions for enhanced delivery of bioactive molecules, US Patent 6706289

Lewis, J.C. and Mascia, N.A., 2012. Airborne space simulator with zero gravity effects. US Patent 8,241,133.

Luther, D.I., 2013. In-line staged horizontal takeoff and landing space plane, US Patent 8,528,853

MacCallum, T.K. and Anderson, G.A., 2008. Variable-altitude testing systems, US Patent 7441473.

MacLeod, J., 2008. Estimating suborbital market structure. Personal assignment, ISU. Available from <www.iisc.im>

Maier, J.F., Eckert, C.M. and Clarkson, P.J., 2016. Model granularity and related concepts. *In Proceedings of ICED 1990,* Dubrovnik, The Design Society, pp. 1058-1065.

Menshenin, Y. and Crawley, E., 2020. A System Concept Representation Framework and its Testing on Patents, Urban Architectural Patterns, and Software Patterns. *In Systems Engineering Journal*, 1-24. https://doi.org/10.1002/sys.21547

Menshenin, Y. and Crawley, E., 2018a. DSM-Based Methods to Represent Specialization Relationships in a Concept Framework. *In DS 96: The 20th International DSM Conference* (pp. 151-157).

Menshenin, Y. and Crawley, E., 2018b. A Framework for Concept and its Testing on Patents. *In INCOSE International Symposium* (Vol. 28, No. 1, pp. 1564-1577).

Menshenin, Y. and Crawley, E., 2018c. Model-Based Concept Framework for Suborbital Human Spaceflight Missions. *In 69th International Astronautical Congress*, Bremen, Germany.

Minovitch, M.A., 1989. Electromagnetic transportation system for manned space travel. US Patent 4,795,113

Model−based Conceptual Design Working Group (MBCD WG) Charter. 2 October 2012, https://www.incose.org/FileNotFound.htm?aspxerrorpath=/docs/default-source/wgcharters/model-based-conceptual-%20design.pdf, (accessed 16.08.18).

Mortensen, N.H., 1999. Design Modelling in a Designer's Workbench. Contribution to a Design Language. *PhD thesis*, Technical University of Denmark

Murphy, M.J. and Zamora, S.P., United Technologies Corporation, 2015. Heat exchanger arrangement for turbine engine, US Patent 9212623

Nakamura, T. and Basili, V.R., 2005. Metrics of software architecture changes based on structural distance. *In 11th IEEE International Software Metrics Symposium (METRICS'05),* pp. 24-24.

NASA Facts, 2001. Robert H. Goddard: American Rocket Pioneer. FS-2001-03-017-GSFC.

NASA, Space Communications and Navigation (SCaN) Network Architecture Definition Document (ADD) Volume 1: Executive Summary. Washington: NASA, pp.2-17.

NASA Systems Engineering Handbook, 2016. Available at: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170001761.pdf (accessed 29.11.2017).

NASA, 2017. Tracking and Data Relay Satellite (TDRS). Available at: https://www.nasa.gov/directorates/heo/scan/services/networks/tdrs_main (accessed 10.05.2019)

NASA, 2016. Near Earth Network User's Guide.

NASA, 1998. Concurrent Engineering in the Jet Propulsion Laboratory Project Design Center. Available at: https://trs.jpl.nasa.gov/handle/2014/19149 (accessed 10.05.2019)

Nock, K.T., McRonald, A.D. and Aaron, K.M., Global Aerospace Corp, 2004. Balloon device for lowering space object orbits. US Patent 6,830,222

Nordlund, M., Taesik, L., Sang-Gook, K., 2015. Axiomatic Design: 30 years after. Available at: https://dspace.mit.edu/handle/1721.1/107378#files-area (accessed 29.11.2017).

Nutt, P.C., Backoff, R.W., 1992. *Strategic management of public and third sector organizations: A handbook for leaders*, Jossey-Bass.

Pahl, G. and Beitz, W., 2013. *Engineering design: a systematic approach*. Springer Science & Business Media.

Pahl, G., Beitz, W., Feldhusen, J. and Grote, K.H., 2007. *Engineering Design: A Systematic Approach*. Springer, London. doi: 10.1007/978-1-84628-319-2

Palsson, B. and Charusanti, P., The Regents Of The University Of California, 2015. Method to generate novel bioactive molecules, US Patent 9080199

Peeters, W., 2010. From suborbital space tourism to commercial personal spaceflight. *Acta Astronautica*, 66(11-12), pp.1625-1632.

Porter, M.E., 2008. *Competitive strategy: Techniques for analyzing industries and competitors*. Simon and Schuster.

Prasad, B., 1996. *Concurrent engineering fundamentals (Vol. 1)*. Upper Saddle River, NJ: Prentice Hall PTR.

Prasad, B., 1995. Sequential versus Concurrent Engineering—An Analogy. *Concurrent Engineering*, 3(4), pp.250-255.

Provitola, A.I., 2002. Electrically powered spacecraft/airship. US Patent 6,357,700.

Redding Jr, 2003. Hypersonic and orbital vehicles system, US Patent 6530543.

Redding Jr, 2001. Hypersonic and orbital vehicles system, US Patent 6257527.

Sanchez, M., Selva, D., Cameron, B., Crawley, E., Seas, A. and Seery, B., 2014. Results of the MIT space communication and navigation architecture study. *In 2014 IEEE Aerospace Conference (pp. 1-14)*.

Sanchez, M., Selva, D., Cameron, B., Crawley, E., Seas, A. and Seery, B., 2013. Exploring the architectural trade space of NASAs Space Communication and Navigation Program. *In 2013 IEEE Aerospace Conference* (pp. 1-16).

Sarver, G.L., 2013. Movable ground based recovery system for reuseable space flight hardware. US Patent 8,498,756.

Schrieverhoff, P., Lindemann, U., 2012. Design for adaptability – support of upgrade strategies to react to evolving stakeholder needs and requirements. *In Proceedings of the 14th International Dependency and Structure Modelling Conference,* DSM 2012, Kyoto, Japan, pp. 43-52.

Schumann, H., Braukhane, A., Gerndt, A., Grundmann, J., Hempel, R., Kazeminejad, B., Romberg, O. and Sippel, M., 2008. Overview of the new concurrent engineering facility at DLR. *In 3rd International Workshop on System & Concurrent Engineering for Space Applications (SECESA)* (pp. 15-17).

Shai O., Reich Y., Hatchuel A., Subrahmanian E., 2013. Creativity and scientific discovery with infused design and its analysis with C-K theory. *Research in Engineering Design*, Vol. 24, pp. 201-214.

Smaling, R. and Weck, O.D., 2007. Assessing risks and opportunities of technology infusion in system design. *Systems engineering*, 10(1), pp.1-25.

Suh, N., 2001. *Axiomatic Design: advances and applications*. MIT-Pappalardo series in Mechanical Engineering. Oxford University Press, USA.

Suh, N., 1990. *The Principles of Design*. Oxford University Press.

Suh, N.P., 1998. Axiomatic design theory for systems. *Research in engineering design*, 10(4), pp.189-209.

Tauri Group, 2014. Suborbital Reusable Vehicles: A 10-Year Forecast for Market Demand.

Toliver, D.M., Cervisi, R.T. and Vanore, A.R., 2000. Method of earth orbit space transportation and return. US Patent 6,068,211.

Tsiolkovsky K., 1903. Exploration of the World Space with Reaction Machines http://www.tsiolkovsky.ru/index.php?option=com_content&task=view&id=64

Trost, J.E., 1986. Statistically nonrepresentative stratified sampling: A sampling technique for qualitative studies. *Qualitative sociology*, 9(1), pp.54-57.

Uhlig, T., Sellmaier, F. and Schmidhuber, M. eds., 2015. *Spacecraft operations*. Springer Vienna.

Ulrich, K., Eppinger, S., 2015. *Product design and development*. McGraw-Hill Higher Education.

USPTO, 2007. Available at: https://www.uspto.gov/learning-and-resources/glossary (accessed 10.05.2019)

Van Bezooijen, R.W., Lockheed Missiles and Space Co Inc, 1998. Techniques for optimizing an autonomous star tracker. US Patent 5,745,869

Watson, J.D. and Crick, F.H., 1953. Molecular structure of nucleic acids. *Nature*, 171(4356), Vancouver, pp. 737-738.

WIPO, 2016. Available at: https://www.wipo.int/export/sites/www/sme/en/documents/pdf/ip_panorama_6_le%20arning_points.pdf (accessed 10.05.2019)

X PRIZE website http://ansari.xprize.org/teams

Yassine, A., Braha, D., 2003. Complex concurrent engineering and the design structure matrix method. *Concurrent Engineering*, Vol. 11 No. 3, pp. 165-176. https://doi.org/10.1177/106329303034503

Appendix A

The list of patents and patent applications based on

"Suborbital spaceflight vehicle" keywords

№	Patent/patent application №	Patent Name	Year	Citation №
1	US4836470A	Aerospace vehicle having multiple propulsion systems on a relatively rotatable flying wing	1989	39
2	US5669584A	Space vehicle apparatus including a cellular sandwich with phase change material	1997	32
3	US20100096491A1	Rocket-powered entertainment vehicle	2010	32
4	US20070194171A1	Rocket-powered vehicle racing information system	2007	59
5	US20080221745A1	Collection and distribution system	2008	83
6	US7287722B2	Rocket-powered vehicle racing competition	2007	19
7	US20060014122A1	Method for qualifying and/or training a private customer for space flight	2006	13

8	US20070068138A1	Rocket vehicle and engine	2007	9
9	US20110268816A1	Apparatuses and systems to process a fluid, and methods for using the same	2011	2
10	US20070128582A1	Method, apparatus, and system for private lunar exploration	2007	16
11	WO2003088187A1	Method for qualifying and/or training a private customer for space flight	2003	5
12	US6530543B2	Hypersonic and orbital vehicles system	2003	29
13	US6257527B1	Hypersonic and orbital vehicles system	2001	22
14	US7441473B2	Variable-altitude testing systems	2008	29
15	US8528853B2	In-line staged horizontal takeoff and landing space plane	2013	61
16	US6612522B1	Flyback booster with removable rocket propulsion module	2003	92
17	US20160031544A1	Glass panel for a space aircraft	2016	12
18	WO2007021781A2	Method, apparatus, and system for private lunar exploration	2007	1
19	US8498756B1	Movable ground based recovery system for reusable space flight hardware	2013	23
20	WO2010014753A2	Rocket-powered entertainment vehicle	2010	4

21	US6173922B1	Failure resistant multiline tether	2001	44
22	US20120228434A1	Pod for space or near-space flights	2012	27
23	US5137372A	Spherical fluid bearing apparatus	1992	14
24	US6431497B1	Failure resistant multiline tether	2002	31
25	US6260807B1	Failure resistant multiline tether	2001	24
26	US6386484B1	Failure resistant multiline tether	2002	27
27	US5129602A	Multistage launch vehicle employing interstage propellant transfer and redundant staging	1992	73
28	CN101522525B	Aircraft with hybrid aerodynamic and space flight, and associated flight control method	2013	3
29	US20170240301A1	Trans-orbital freight and passenger carrier apparatuses supporting trans-orbital pipeline operations	2017	0
30	US20160244188A1	Trans-orbital freight and passenger carrier apparatuses supporting trans-orbital pipeline operations	2016	0
31	US6290186B1	Planar hoytether failure resistant multiline tether	2001	21
32	RU2006129377U	Suborbital and orbital vehicle	2007	0

33	US6286788B1	Alternate interconnection hoytether failure resistant multiline tether	2001	20
34	US4795113A	Electromagnetic transportation system for manned space travel	1989	58
35	CN101580133A	Gas rocket space vehicle	2009	4
36	US20140158812A1	In-line staged horizontal takeoff vehicles and related methods	2014	17
37	JP5677092B2	Spacecraft (spacecraft) rear fuselage equipment	2015	1
38	US20050230529A1	Hiigh wing monoplane aerospace plane based fighter	2005	7
39	US6921051B2	System for the delivery and orbital maintenance of micro satellites and small space-based instruments	2005	46
40	US6357700B1	Electrically powered spacecraft/airship	2002	40
41	WO1998048089A1	Failure resistant multiline tether	1998	13
42	US8241133B1	Airborne space simulator with zero gravity effects	2012	25
43	CN106781835A	Juvenile science education dedicated Star coach model	2017	0
44	US20100327107A1	Bidirectional control surfaces for use	2010	74

		with high speed vehicles, and associated systems and methods		
45	CN204943932U	Close on space solar heat storage device	2016	0
46	US5402965A	Reusable flyback satellite	1995	56
47	US20130261876A1	Novel systems and methods for non-destructive inspection of airplanes	2013	45
48	US20090140101A1	Direct Flight Far Space Shuttle	2009	27
49	US6068211A	Method of earth orbit space transportation and return	2000	31
50	WO2000066425A2	Airship/spacecraft	2000	11
51	RU2015147386A	Speed control apparatus aerospace plane during the transition from space flight phase to atmospheric flight phase and a corresponding method of transitioning	2017	0
52	CN206218243U	Approaching spacecraft with pod	2017	0
53	US20060217169A1	Variable gravity gaming	2016	12
54	CN106524535A	Near space solar heat storage device	2017	0
55	US20150068052A1	Mechanical and fluid system and method for the prevention and control of motion sickness, motion- induced	2015	14

		vision sickness, and other variants of spatial disorientation and vertigo		
56	US6419191B1	Electrodynamic tether control	2002	32
57	JPH07196098A	Reusable flyback satellite system, reusable flyback vehicle, and method for earth orbit space transportation and return using reusable flyback satellite	1995	2
58	CN105954763A	Real-time tracking system for flight test of sphere body near space aerocraft	2016	0
59	US6830222B1	Balloon device for lowering space object orbits	2004	127
60	US20140306065A1	Launch vehicle and system and method for economically efficient launch thereof	2014	24
61	CN102745336A	Method and equipment for modulating gravity and launching vehicle	2012	0
62	RU2202500C2	Method of recovery of recoverable launch vehicles and device for realization of this method	2003	2
63	US20100326045A1	Multiple-use rocket engines and associated systems and methods	2010	11

64	US20150232204A1	Aerospace plane system	2015	8
65	US20150273179A1	Mechanical and fluid system and method for the prevention and control of motion sickness, motion- induced vision sickness, and other variants of spatial disorientation and vertigo	2015	0
66	JP2012530020A	Marine landing and related systems and methods of space launch vehicles	2012	12
67	US20140166815A1	Tether for spacecraft reaction control system	2014	6
68	US20110256512A1	Methods and apparatus for modulating variable gravities and launching vehicles	2011	4
69	US9079674B1	Composite structures for aerospace vehicles, and associated systems and methods	2015	70
70	US20120175466A1	Space debris removal using upper atmosphere	2012	22
71	JP2010540830A	Device for powering the pump of a rocket engine using an inertia disc	2010	11
72	US20150076287A1	Tether for spacecraft reaction control	2015	8

		system		
73	US5745869A	Techniques for optimizing an autonomous star tracker	1998	133
74	US20130007935A1	Rocket Launch System and Supporting Apparatus	2013	54
75	ES2617380T3	Apparatus and method of direct broadcast alert	2017	0
76	US6052987A	Non-propellant fluid cooled spacecraft rocket engine	2000	35
77	CN106081126A	Application and design for embedding bionic honeycomb-shaped active safety escape capsule into aircraft	2016	0
78	US20160363529A1	Windowless microbolometer array	2016	3
79	CN104919166B	Rocket motor for the turbo pump actuator device	2017	5
80	CA2875466C	Lift ring assembly for a rocket launch system	2016	0
81	US20160001899A1	Gas Envelope Propulsion System and Related Methods	2016	0
82	US20150007549A1	Rotary Turbo Rocket	2015	24
83	CN1833084A	Variable gravity gaming	2006	0

84	US20160347482A1	Systems and methods for estimating parameters of a spacecraft based on emission from an atomic or molecular product of a plume from the spacecraft	2016	19
85	CN105066994A	Data fusion method for flush air data system and inertial navigation system	2015	4

Appendix B

32 projects of the suborbital human spaceflight missions

№	Name of the Company	Type of Vehicle
1	Acceleration Engineering	Single stage rocket
2	Advent Launch Services	Single stage rocket
3	Aeronautics and Cosmonautics Romanian Association	Two stages rocket and capsule
4	Armadillo Aerospace	Single stage rocket
5	American Astronautics Corporation	2 stage rocket and booster
6	Bristol Spaceplanes	Single stage rocketplane
7	Canadian Arrow	Two stage rocket
8	Pablo de Leon & Associates	Two stages rocket/capsule
9	Flight Exploration*	Two stages rocket/ capsule

10	Fundamental Technology Systems	Single stage rocketplane rocket
11	Interorbital Systems	Single stage rocket
12	Kelly Space and Technology	Two stage plane/rocket
13	Lone Star Space Access Corporation	Single stage rocketplane
14	Micro-Space Inc.	Single stage rocket
15	Panaero Inc.	Single stage rocketplane
16	Pioneer Rocketplane	Single stage rocketplane
17	Starchaser Industries	Two stage rocket with strap-on boosters for 1st stage
18	Suborbital Corporation	Two stage plane/rocket
19	TGV Rockets	Single stage rocket
20	ARCA	Two stage rocket
21	Blue Origin	Two stage rocket
22	Virgin Galactic	Two stage plane/rocket
23	XCOR	Single stage rocketplane

24	Copenhagen Suborbital	Single stage rocket
25	Dassault-Aviation	Two stage plane/rocket
26	EADS	Single stage rocketplane
27	Cosmocourse	Two stage rocket
28	Scaled Composites	Two stage plane/rocket
29	Vela Technology Development	Two stage plane/rocket
30	David L. Burkhead - Spacecub	Single stage rocket
31	Andrew Space and Technology	Single stage rocketplane
32	Myasishchev Design Bureau	Two stage plane/rocket