ENVIRONMENT-BASED DESIGN (EBD) ENABLED SCENARIO ANALYSIS FOR AEROSPACE PRODUCT DEVELOPMENT

Alexandra Miklin

A Thesis

in

the Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science (Quality System Engineering) at the
Concordia Institute for Information Systems Engineering
Montreal, Quebec, Canada

January 2021

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: Alexandra Miklin

Entitled: Environment-Based Design (EBD) Enabled Scenario Analysis for Aerospace Product Development

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality System Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________ Chair
Dr. C. Wang

______________________________ Examiner
Dr. A. Akgunduz

______________________________ Examiner
Dr. C. Wang

______________________________ Supervisor
Dr. Y. Zeng

______________________________ Co-Supervisor
Dr. S. Dufresne

Approved by ______________________________
Dr. A. Ben Hamza, Director
Concordia Institute for Information Systems Engineering

____
Dr. M. Debbabi, Interim Dean
Faculty of Engineering and Computer Science

Date ______________________________

ABSTRACT

ENVIRONMENT-BASED DESIGN (EBD) ENABLED SCENARIO ANALYSIS FOR AEROSPACE PRODUCT DEVELOPMENT

Alexandra Miklin

The process of product development is one of the most important and complex activities of an industrial company, especially in the aerospace domain. The requirements of stakeholders play a central role here. Those requirements come from internal and external sources, and further in the process are translated into organizational knowledge (e.g. specifications, etc.). Utilization of this approved knowledge in new programs is very beneficial, and can save resources for design, may reduce the risk of failures, and may ease certification, manufacturing, maintenance and other processes.

The development of such complex systems is an extremely challenging process because of their growing complexity and frequently changing product environment and requirements. Furthermore, the development process affects the amount of information or knowledge (e.g. requirements) to be managed by an organization.

Building on information gained from past projects, this thesis proposes a method to structure the knowledge that was gained. The Environment-Based Design (EBD) methodology is a platform for characterization of this method. Utilizing the EBD tool, a Recursive Object Model (ROM), enables a graphical representation of the knowledge for ontology (components and relationships) that is found within the environment-based analysis. In addition, an example of analysis for the top-level requirement based EBD methodology is provided. This example proposes an opportunity for searching specific requirements by tagging with key words. The

results were compared with a model approach based on the Arcadia method, and was found to be effective.

ACKNOWLEDGEMENTS

I would first like to thank my supervisors Dr. Yong Zeng and Dr. Stéphane Dufresne for their advice and guidance that helped me and lit my way during the course work, as well as for their kindness and patience. Their doors were always open to me whenever I had questions. Their significant and extraordinary support has always guided me in the right direction at all stages of my research. I truly believe that what I have learned will open many new opportunities for me in the future, both professionally and personally.

Thanks are also due to those involved in this project from Bombardier. Their patience, knowledge, experience and open communication allowed me to progress with my research effectively and efficiently.

I would also like to acknowledge Dr. Lan Lin of the Department of Building, Civil and Environmental Engineering at Concordia University, as the second reader of this thesis, for her constructive comments.

I appreciate very much the help from all my colleagues in the Design Lab, as well as Mrs. Xiaoying Wang and Mr. Cheleeger Ken for many fruitful discussions that opened new ideas and gave me moral support during this journey.

I am thankful to Concordia University in general and the CIISE Department in the Gina Cody School of Engineering and Computer Science, in particular, the administrative staff. The university and staff have provided me an excellent educational environment and services.

I would like express my profound gratitude to my parents, my brother and my parents-in-law for their support and confidence in me, thank you all!

Lastly but most importantly, I would like to acknowledge my dear husband and children, who always supported and inspired me throughout all the years of my graduate study and work. We as a family have been waiting for this moment for so long. I dedicate this work to all of us. I would never be who I am without each of you.

Table of Contents:

List of Figures:

List of Tables:

CHAPTER 1 Introduction

1.1 Motivation

Product development companies must continually design and manufacture successful products to compete in a dynamic marketplace. This makes product development one of the most important and complex activities of an industrial company. The process is fundamentally based on stakeholders' requirements that have been acquired from internal and external sources (Nilsson & Fagerström, 2006).

These requirements are translated into organizational knowledge. This process involves organizational experience, successful and approved design contents, lessons learned, and many other elements. The requirements become a fundamental and essential part of organizational knowledge that plays a central role in an organization's success in business, quality, reliability and innovation. It is beneficial to utilize this approved knowledge in the development process of new products. This approach can save resources for design, and can ease certification, manufacturing, maintenance and other processes in the product lifecycle.

Development of complex systems is a real challenge since the requirements and product environment frequently change with time. Accordingly, the leveraging of previously successful design specifications for new design processes is essential to minimize the risk of mistakes and failures.

An example of failure in a project that involved partial understanding of requirements and constraints was transformational satellite communication system (TSAT), an orbit-to-ground laser communication program. After Operation Desert Storm, the defense officials realized that there was a need to increase data communication capabilities since the existing military satellite communication system (MILSATCOM) was insufficient for this purpose. Therefore, TSAT, an orbit-to-ground laser communication program, was developed. The program was generally on-schedule, but with high overall costs and an uncertain budgetary environment. The program was based on the use of existing Advanced Extremely High Frequency (AEHF) satellites, which were available at that time. Unfortunately, it was found later that there was no appropriate operating system to run TSAT. The TSAT program, which had cost about $2.3 billion, was eventually closed by Secretary of Defense in early 2009 (Rodriguez, 2014).

Dufresne (2008) explains that the greater the complexity of the system, the larger the number of requirements needed to define the system. Furthermore, these requirements should be complete as much as possible to reduce the potential of duality or ambiguity in the data acquisition process. For this purpose, conceptual design is an essential step of the product development process to either provide a high level of elicitation of new requirements or to make explicit the implicit ones.

This thesis proposes a method of knowledge creation for the system design requirements ontology, and particularly for the conceptual design domain of complex aircraft systems. This method defines the classification possibilities and the structure of the system requirements in order to create relationships within the knowledge base (KB). This makes it possible to use previous knowledge to define new requirements for a new product. Since the requirements are presented in text format, an ontology and knowledge for the ontology are required to classify them. In addition, this thesis presents an example of the possibility of tagging for information within the searching process. This was enabled by utilization of the Environment-Based Design (EBD) methodology rules. This process was evaluated in parallel by models that were created following the rules of modeling and the Arcadia Method (Bonnet, 2015).

1.2 Research objectives, questions and hypotheses

As mentioned above, it is essential to manage requirements in a way that will allow them to be utilized in new programs. This thesis will follow the Environment-Based Design (EBD) methodology in order to classify organizational knowledge (e.g. requirements). The main objectives of the research are:

1. To define taxonomies of requirements that will fulfil parts of the ontology to standardize the classification of requirements in text format. The taxonomies should relate to the aircraft, scenarios, requirements levels and other relevant information.
2. To improve the knowledge base as a function of the defined taxonomies.
3. To define the process of tagging for searching purposes of the knowledge base.

Formally, these research objectives lead to the definition of the following research questions (RQ):

RQ1 for Objective#1: How to classify requirements with respect to stakeholders, entities, entity types and relationships, functions, supported items in the functions, and systems?

RQ2 for Objective #1: How to structure the requirements in order to relate them to the knowledge base (KB)?

RQ3 for Objective #2: How to improve the KB with regard to defined taxonomies and relationships between entities?

RQ4 for Objective #3: How to utilize and reuse existing experience (requirements, functions, scenarios and others) in order to assist the architect in defining new or modified requirements for a future program?

The objective of these research questions is to direct the research by creating hypotheses to answer. Hence, the research questions and the hypotheses focus on and summarize the motivation of the research and concentrate attention on the specific problems to answer.

The abovementioned research questions lead to the creation of the following hypotheses that help to structure and direct solution generation for the current research problems:

H1: Classification of requirements can be performed by taxonomy models that relate to all important groups and their relationships. These groups include stakeholder needs, entities, entity types, functions and others.

H2: Structure or organization of requirements can be performed by a modeling process. This process will use a uniform language and key words.

H3: Transformation of requirements from textual to model format with respect to defined taxonomy groups, levels and relationships will improve the KB.

H4: Utilization or reuse of existing experience will be possible by definition of an appropriate searching process based on the modeled KB and defined taxonomies.

A representation of the main research steps is given in Section 1.4, Figure 1.

1.3 Research contributions

The main contribution of this research is to introduce a classification method and structure for system design requirements which provides knowledge for a development ontology diagram. The method defines categories and their relationships for standardization of requirements classification and common understanding of its structure within the organization. In addition, it allows one to understand the relationships within different requirements, and promote implementation of the Model-Based System Engineering (MBSE) approach to design and other domains.

The second contribution of this research is the proposal of a tagging method for searching past knowledge. This method was also defined based on the rules of the Environment-Based Design (EBD) methodology by using a performance network and reduction of conflicts.

1.4 Organization of the research

Motivation for the knowledge structure and management is discussed in Section 1.1. The definition of knowledge and structure is discussed in Sections 2.5 and 2.6, and the importance of searchable knowledge is covered in Sections 2.7 and 2.8. An outline of the research is represented in Figure 1 below.

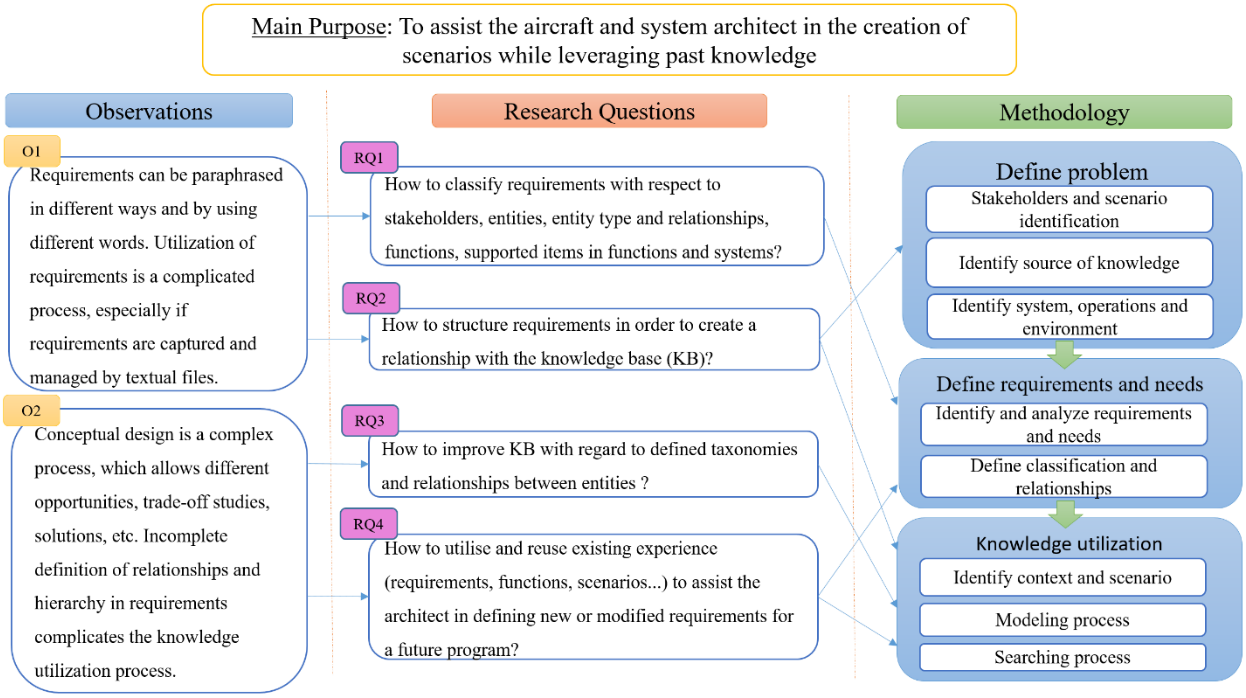

Figure 1: Key research stages

Represented above are the main observations for the specific problem, research questions that were defined, and the key research stages leading to solutions of the problem. The current chapter has introduced the research topic, research objectives, challenges and scope. In addition, it defined the research questions and hypotheses. Chapter 2 gives the background, literature review, and overview of the important areas forming the system environment. This section also describes the principal steps of the EBD methodology that are used in this research. Chapter 3 describes the implementation of the EBD methodology and as a result represents a knowledge for taxonomies and relationships for further opportunities of implementation in Aircraft Design Requirements Ontology (ADRO). Chapter 4 concludes the major findings of the study and proposes additional work for the future.

CHAPTER 2 Background and literature review

2.1 Aerospace product development process

An aircraft is a complex entity that includes systems, subsystems, components, software and others. The integration of all these with appropriate communication tools is essential for reliable functioning of the aircraft.

According to INCOSE, a system is a combination of interacting elements that are organized to achieve one or more stated purposes (INCOSE, 2006). Furthermore, according to (ISO/IEC 15288, 2002) a system is created and utilized to provide services in defined environments for stakeholder's benefits. They include or interact with humans, hardware, software, different processes and procedures, with facilities and components from the natural environment (e.g. water, minerals, organisms and others). The definition of a system, its architecture and elements depend on an observer's interests and responsibilities.

In general, the purpose of an aerospace product development process is to provide a valuable product (e.g. system) or service to the customer. The reason is to foster customer satisfaction and loyalty. This development process is the most important component of the entire procedure. The development process includes three levels: concept, system, and subsystem/component levels. A concept level provides a system concept description, a system level describe the system in performance requirement terms, and a subsystem/component level describes product performance based on the subsystem and its components, and furthermore provides a set of detailed descriptions of the product's essential characteristics (Department of Defense, 2001). Figure 2 below represents this process.

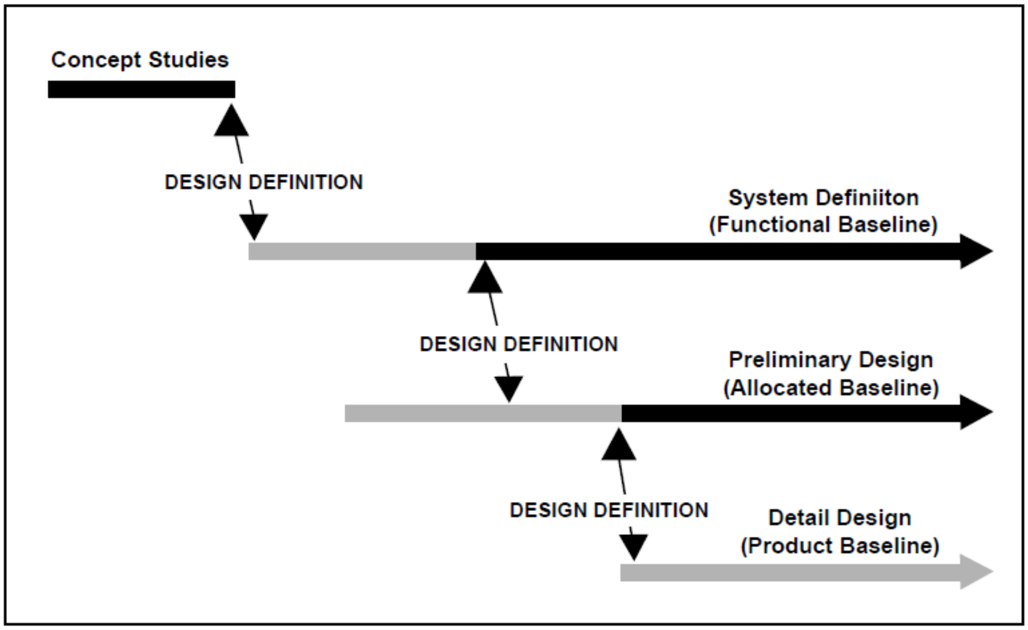

Figure 2: Development Phasing (Department of Defense, 2001)

In the aerospace domain, an aircraft design includes three major phases: conceptual, preliminary and detailed design phases. The conceptual design starts with requirements analysis. Here, the designers examine a wide range of aircraft configuration concepts, perform trade studies and ultimately settle on a single best design. The second phase is preliminary design that is characterized by a maturation of the selected design approach, detailed analysis, and increasing level of confidence that the design will function as desired. The termination of this phase results in a "freeze" for design, forbidding further changes to the overall design arrangement. The third step is detailed design. This is the most expensive phase of the design process. It is characterized by detailed drawings or CAD files of all components, creation of production files, procedures, design tooling and fabrication processes, tests, etc. (Raymer, 2012).

The major issues in aerospace product development are the complexity of the systems that comprise them, the amount of data and information to be managed, constant evolving technologies, and changing requirements. These affect system architecture, integration between elements of the system, and create additional risks to the development of new systems.

(Eppinger & Browning, 2012) explain that the world is growing more complex every day. System environment analysis, at micro and macro levels, accumulates an exponentially increasing amount of information and data. This information empowers engineers to design and build ever more complex systems. Currently, however, the development domain is overwhelmed with more information than can be digested, which stimulates the desire to characterize and implement effective search and filtering of information to help accessing only the relevant information for new projects.

2.2 Scenario analysis for aerospace product development

Scenarios and scenario analysis processes play a central role in the aerospace product development process. INCOSE states that scenarios and what-if thinking are essential tools for designers who must cope with the uncertainty of the future. This is a strategic planning tool that serves as a methodology for planning and decision-making in complex and uncertain environments. The scenario creation process makes people to think creatively, it helps to see important factors and identify requirements that might otherwise be overlooked (INCOSE, 2006).

Kahn K.B. explains also that the scenario analysis process is a tool for envisioning alternate futures so that a strategy can be formulated to respond to future opportunities and challenges (Kahn, 2010).

Aerospace Recommended Practice ARP4754A expands on the use of scenario analysis in the aerospace product development domain. The scenario analysis process here is essential for effective identification of missing requirements early in the design process, and for describing the procedures in the operating and maintenance manuals. This process provides clarity to users that interact with the system on how the system is proposed to work in different operational scenarios. This clarity should aid in the identification of missing desired behaviors or protection features. The scenarios may also describe the behavior of the system under different conditions and operating modes. Each scenario can examine a sequence of steps, from initiation of an action by the user, through each action step taken by an identified system or person on the way to the end goal. The scenarios should cover anomalous operating conditions as well, including possible misbehaviors. They may also be used to allocate specific functions. The scenarios in the aerospace product development domain are also useful for validation and verification purposes (SAE Aerospace, 2010).

This process leads to the generation of new requirements and functions, and those in turn providing opportunity for iterative analysis or creation of a new sequential series of scenarios such as represented in Figure 3 below.

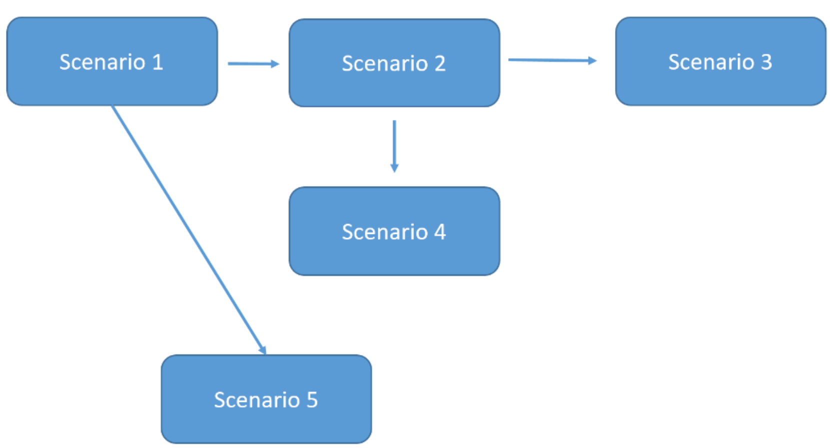

Figure 3: General creation of new sequential series of scenarios.

Creation of scenarios is an essential activity that may involve interviews with operators of current or similar systems, potential users, and meetings of an Interface Working Group. The outcomes of these activities can be expressed by models and simulations (INCOSE, 2006).

There is a huge amount of information and data that have been accumulated by airframe manufactures. This knowledge might be used for creation of new scenarios. Effective utilization of this knowledge is very important, hence it is essential to structure it in a way that will support its utilization.

2.3 EBD Methodology and environment analysis

Environment-Based Design (EBD) was developed over the last three decades (Zeng, 2011). The EBD methodology is based on the idea that the design changes the environment to a desired one by generating a new artefact. The key message of this methodology is the importance of understanding a problem environment and its challenge, then focus on the milestones that lead to solution generation.

The motto of EBD can be expressed as "Design starts from the environment, functions for the environment, and brings changes to the environment" (Zeng, 2015). Herein, the design is a recursive process that defines the problem, generates design knowledge, and design solutions.

Environment-Based Design (EBD) is a methodology that provides step-by-step procedures to guide a designer in the environment changing process(Zeng, 2011). This methodology includes three main interdependent activities: environment analysis, conflict identification, and solution generation. The newly generated solutions apply to the environment, and produce the new environment, where the solution can be viewed as a new component of the generated environment.

The objective of the environment analysis activity is to identify the environment in which the desired product is expected to work. This methodology explains that the environment includes the components and the relationships that occur between those components and the product (Zeng, 2011). The main purpose of implementing EBD methodology in this research is to allow a structured but flexible approach in the definition of components and relationships for the requirements ontology. Graphically, all this knowledge will be represented by using a Recursive Object Model (ROM) diagram that is widely used in the EBD approach.

The first step in the EBD methodology is to fully understand the design problem. Yong Zeng in his research paper proposed a Recursive Object Model (ROM) (Zeng, 2008).

ROM is a platform for representing the design problem for the EBD methodology, and recursive logic is the backbone of design reasoning in this model. The ROM diagram was originally developed to deal with linguistic information, but now is widely used in different processes of analysis and understanding of different kinds of information and design challenges (Zeng, 2011). ROM can be seen as a refined representation of the environment structure that usually includes three types of interaction operations: constraint, predicate and connection.

Pop R. explains that the theory underlying the ROM diagram is the "Axiomatic Theory of Design Modeling" (ATDM). This is a logical tool that can be used to represent and reason about object structures. Here, axiomatic theory provides the designer with a logical approach. This theory defines axioms that deal with objects (Pop, 2011).

Gonzalez A.M. also discusses ROM and ATDM in his research. He states that ATDM defines the axioms of objects as follows:

Axiom 1: Everything in the universe is an object.

Axiom 2: There are relations between objects.

He explains that because it can follow from the axioms that are defined above, the characteristics of the relations play a critical role in ATDM. For this reason, it is essential to define a set of basic relations in order to capture the nature of the object representation (Gonzalez, 2008).

Table 1 below lists types of symbols, their graphic representations and their definitions that occur in ROM diagrams. The ROM diagram is composed of single words and their mutual relations that represent effectively, structurally and graphically the statements or any kind of information and their functional relationships in the language (Zeng, 2008).

Type		**Graphic Representation**	**Definition**
Object	Object	O	Everything in the universe is an object.
	Compound Object	O	It is an object that includes at least two other objects in it.
Relations	Constraint		It is a descriptive, limiting, or particularizing relation of one object to another.
	Connection	ι	It is to connect two objects that do not constrain each other.
	Predicate	ρ	It describes an act of an object on another or that describes the states of an object.

Table 1: Elements of the recursive object model (ROM) (Zeng, 2008)

In addition, it is important to mention that the fundamental process of environment analysis in the EBD methodology is to ask questions. In this step of the analysis, two types of questions can be asked. The first type is generic questions for the clarification and extension of the meaning of the design problem; the second type is domain specific questions for implicit design information related to the current design problem. The ROM diagram, as a linguistic tool in design, is used here like a platform for generating these questions (Zeng, 2011).

The procedure, rules and questions described in Tables 2, 3 and 4, respectively, help in proceeding with the environment analysis, question asking step, and generation of the refined ROM diagram with components that are found within the process.

Step 1:	Generate the ROM diagram for the design problem.
Step 2:	Ask a question using the rules given in Table 4 and templates in Table 5.
Step 3:	Find answers to the question.
Step 4:	Generate the ROM diagram for the answer and merge it back to the original ROM diagram.
Step 5:	Repeat Step 1-4 until no more questions.

Table 2: Procedure for generic question asking (Zeng, 2011)

Rule 1:	Before an object can be further defined, the objects constraining them should be further refined.
Rule 2:	An object with the most undefined constraints should be considered first.

Table 3: Rules for generic questions (Zeng, 2011)

T1	For a concrete, proper, or abstract noun *N*	Question: What is *N*?
T2	For a noun naming a quantity *Q* of an object *N*, such as height, width, length, capacity, and level	How many / much / long / big / … is the *Q* of *N*?
T3	For a verb *V*	How to *V*? Or Why *V*?
T4	For a modifier *M* of a verb *V*	Why *V M*?
T5	For an adjective or an adverb *A*	What do you mean by *A*?
T6	For a relation *R* that misses related objects	What (who) *R* (the given object)? Or (the given object) *R* what (whom)?

Table 4: Question template for object analysis (M. Wang & Zeng, 2009)

The EBD methodology provides a roadmap as guidance for the identification of all environment components and their relationships in each of the design stages of the product. This roadmap focuses on defining environments such as natural, built and human through all lifecycle stages of the product. Here the natural environment represents all of the components that are present naturally in the product environment. The built environment describes components that are built by humans, and the human environment represents people that have direct or indirect relationships with the product in each of its lifecycle stages.

Figure 4 represents an example of a product environment categorization, with seven lifecycle stages of a product. The stages that are defined in this example are: design, manufacture, sales, transportation, use, maintenance, and recycling. Each stage may include environment components that can be represented as a pyramid with the natural environment in the base, built environment in the middle and the human environment at the top (Zeng, 2011).

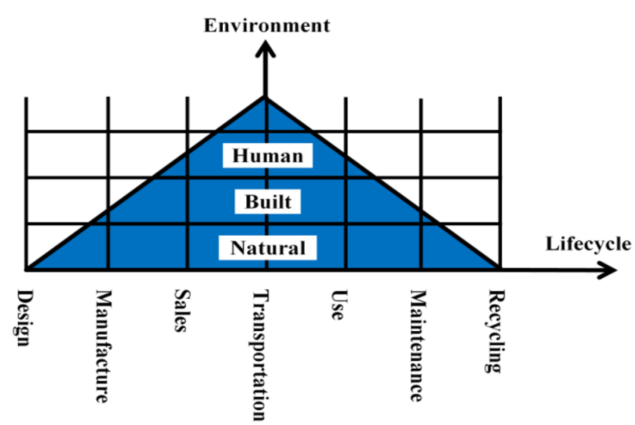

Figure 4: Roadmap for domain related environment: an example (Zeng, 2011)

This roadmap and a procedure for identifying environment components by asking domain-specific questions, described in Table 5, help to find and categorize the components that exist in specific domain environment. Using these procedures makes it easier to focus on the identification of components vis-à-vis the specific environment affected by the project (Zeng, 2011).

Step 1:	Ask and answer the question: what is the lifecycle of the product to be designed?
Step 2:	For each event included in the lifecycle, ask and answer the question: what are the relevant components for natural, built, and human environments for this event?
Step 3:	Generate the ROM diagram for each answer and merge them back to the original ROM diagram.
Step 4:	Apply the procedure for generic question asking.

Table 5: Procedure for asking domain specific questions (Zeng, 2011)

For the next step, an updated ROM diagram can be generated that will summarize the results of the environment analysis. This diagram is a graphical representation of information that is found using the environment analysis process (Tan et al., 2013).

Gutierrez (2018) followed the EBD approach. Step by step from the definition of the main problem, where he represented two "concepts" (nouns), he continued with the method and proposed the core ontology. This proposed core ontology has 50 core components and relationships between them.

The main purpose of this ontology was to overcome communication challenges existing in the engineering domain of mechatronic products. The ontology was designed for communication purposes in the domain of requirements and system life cycle processes. Finally, the proposed ontology was expected to be an initial model for communication and understanding in multidisciplinary design projects. Gutierrez (2018) explained that the ontology can help teams to define and develop specific vocabulary and requirements in their domain of interest. This will extend the ontology with new components particular to a domain of interest. The relationships (verbs) in the ontology also may suggest how to develop and logically manage requirements during the design process. He introduced that his development was an initial stage, and further research in design guidelines is needed.

In this research, the EBD methodology will be examined for its capability as a suitable platform for developing knowledge for aircraft design requirements ontologies. By following the main procedure of the method, then probing recursively into domain specific information, a trial will be conducted to introduce the list of key components and their relationships. The relationships and/or components can be used for further searching of relevant information that is stored in the knowledge base. An example of definition of key words for tagging process will also be provided. This process will assist with the characterization step for later possible development of the software searching process based on the EBD methodology.

One of the main design activities in the EBD methodology is conflict identification (Zeng, 2011). The product environment includes many components. Usually, conflicts exist between those environment components or in the relations between these components (Sun et al., 2011). This is further explained (Zeng, 2014): "*Conflict is rooted in the nature of design problems, in that there is a conflict between form and function, and between design problem and design solutions*."

In this thesis, the EBD methodology provides guidance and helps to identify existing potential conflicts. Implementation of the rules of the EBD methodology are directed at resolving these conflicts (Sun et al., 2011).

The EBD methodology explains (Zeng, 2014) that a conflict refers to an insufficiency of resources for an object to produce a desired action on its environment or to accommodate the object's action on its environment. There may be different actions in the product environment that could appear as interactions between them. Those interactions may also consume resources. The resource is a trigger for an action and the resource is usually consumed before interactions occur. The EBD methodology has also defined two kinds of conflicts: active and reactive. Active conflict refers to an insufficiency of resources for an object in order to produce a response from that object. Reactive conflict, in turn, refers to an insufficiency of resources to accommodate an object or the responses from the object. In order to identify and respond to the conflicts, the active actions can be extracted from the ROM diagram. The ROM diagram represents the product-environment system behind a design problem statement. The actions will be checked for possible combinations of interactions that could lead to conflicts.

Figure 5 below represents four kinds of basic interactions that can be identified in the ROM diagram. Here, each action verb defines one interaction.

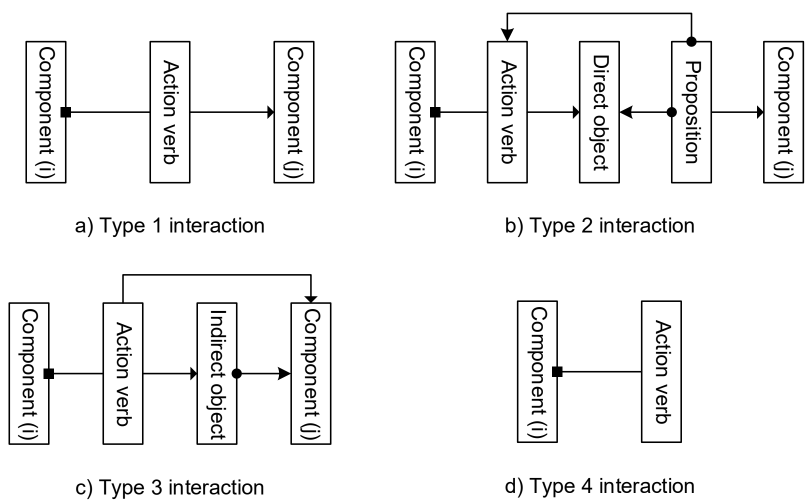

Figure 5: Basic actions and responses on an object: ROM representation (Zeng, 2014)

The result of the environment analysis will be an interaction-dependency network. This network represents the dependency of relations between interactions and leads to identification of the conflicts (Zeng, 2014).

The EBD methodology considers the undesired conflicts in the existing environment as the driving force of design. *“By proper identification of all those conflicts, a designer will be able to develop satisfactory solutions for the design problem”* (Zeng, 2011).

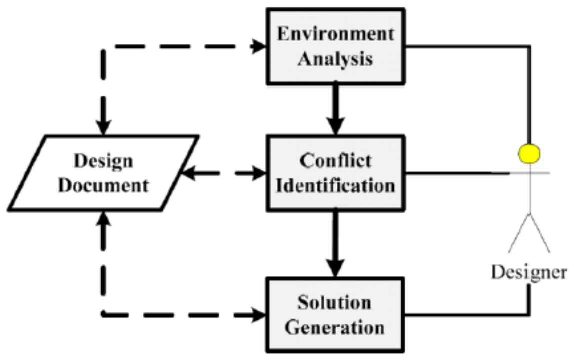

Figure 6: Environment-Based Design: Process Model (Zeng, 2011)

Furthermore, as shown in Figure 6, *"Recursive resolution of a complex design problem can be conducted through environment decomposition, which will identify the key conflict to start with"* (Zeng, 2015).

The EBD methodology has proven effective in many applications (Tan et al., 2013),(Gutierrez, 2018), (Pop, 2011), (Tan et al., 2012). It can be used in different ways, using every EBD step or only some, depending on the requirements of the researchers. Certain steps of this methodology must be performed sequentially from the very beginning of the analysis, while others can be used or not, depending on the direction of the research. When implementing the EBD methodology for the development of ADRO (Aircraft Design Requirements Ontology) the main steps of the analysis (linguistic and environment) were utilized.

Linguistic analysis is supported by a ROM diagram, which should be used throughout the entire analysis and representation steps to the final results. The final results are a representation of the components and relationships that represent knowledge for ADRO.

The environment analysis can move recursively, and/or by proceeding down through different hierarchical levels of information by asking questions and answering them, following the rules of the EBD methodology. This analysis involves a decomposition process that includes aspects of the natural, built and human environments throughout the different lifecycle stages of the main constrained word.

The above mentioned analysis provides a structured way to define components (e.g. knowledge) for ontology with an efficient level of modularity. Following this direction in the research it is easy to update or enrich each part of the ontology and at the same time does not interfere with other parts.

The EBD approach can focus and guide engineers and other specialists to search for relevant information (requirements, designs, reports, etc.) throughout the ontology. It can also reduce mistakes, save time and resources, and can be suggested as a supportive method for junior engineers.

The main reason for using EBD is that the methodology can assist in extracting more relevant and focused information in short period of time compared to other approaches. EBD has the advantage and the capability of identification and representation of the key environment components and their relationships intuitively, which is helpful for understanding the design concepts in a logical and systematic manner (Tan et al., 2013).

2.4 Purpose of requirements and requirement analysis

"Product requirements are descriptions of the desired solution to a design problem. In engineering design, just as in all other design problems, the efficient, precise, and complete specification of design requirements is critical if designers are to deliver a quality design solution within a reasonable range of cost and time."(M. Wang & Zeng, 2009).

According to Weissman et al. the developing process of a new electro-mechanical device begins with the generating of product design requirements. These describe the functions of the desired product and the operational environment, and are expressed as requirement statements (Weissman et al., 2011).

A logical management process is necessary for these requirements. This process should support the collection, analysis, and validation of the requirements. It should supply an appropriate platform of communication and negotiations within the teams in an organization. This process should trigger elicitation of new requirements, better understanding of the stakeholders needs and should support of the architecture of the design process (INCOSE, 2006).

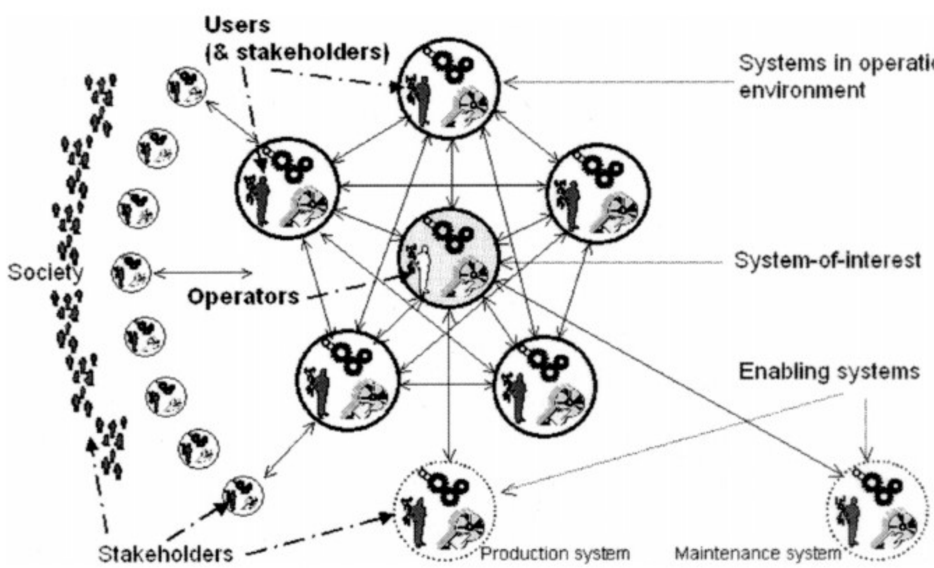

Figure 7: Operators, users and stakeholders (Arnold & Lawson, 2004)

Figure 7 above represents in general the basic communications within the requirements elicitation process for the purpose of capturing the needs of stakeholders, operators and users across systems boundaries. The requirements analysis and elicitation process is an iterative one and it benefits from continuous communication and validation with stakeholders (INCOSE, 2006). This process is the backbone in the definition of project boundaries (Department of Defense, 2001).

Within the requirements analysis process, system engineers elicit the real system requirements and the emergent properties of the system and reduce complexity to avoid unexpected and unpredictable behaviour of the system and undesirable risk (INCOSE, 2006).

Weissman et al. discuss relating the requirements analysis and the elicitation process. This process, for a new product, can require significant time and expertise. They state that the designer can choose to write the requirements from scratch, or can utilize requirements from prior designed products. They believe that reusing existing requirements is a possible approach for ensuring completeness and to reduce the time involved (Weissman et al., 2011).

2.5 Organizational knowledge: requirements, design and other documents

The requirement statements, scenarios, design documents, validation and verification protocols, and reports are only a part of organizational knowledge. This knowledge can represent the uniqueness of the product and might have a specific effect on organizational involvement in the global market. For this and other reasons, it is important to manage this knowledge properly.

In this research attention will be focused on the structure of requirement statements. Structured product requirements are more likely to lead to the stability of an organization's knowledge platform. For this reason, the competitive market is increasingly interested in leveraging successful product requirements from previous projects (Dufresne, 2008).

Weissman et al. stated that requirements development is an important part of the product development process. Any incompleteness, ambiguity or inconsistency here can lead to unnecessary design iterations and increased design time and cost. They explain that many organizations now use a word processor approach to prepare the requirements statements. This makes the process inconsistent with the meaning of vocabulary that is transferred from one speciality to another and it compromises the ability to search effectively (Weissman et al., 2011).

According to Soon Chong Johnson Lim et al., in today's marketplace, which is overflowing with a range of products with varying functional complexity, organizational knowledge management is of great importance. This process refers to efficient storage of knowledge and timely retrieval, which becomes more complex. They suggest that ontology can be a very promising approach for knowledge management, sharing and retrieval. This knowledge management approach helps to improve the completeness of the design information modeling process. It also simplifies and structures the change management process. Therefore, they believe that this process is very important for design analysis and timely decision making (Lim et al., 2009).

2.6 Organizational knowledge classification

Knowledge classification has different approaches and techniques. Some researchers classify knowledge logically or semantically, such as by using a taxonomy approach, or in an ontology representation.

Dufresne S. provides an example of taxonomy for system requirements. He explains that taxonomy of requirements is created to classify the information gathered during the problem definition. Furthermore, that a taxonomy approach should be used to store and manage this knowledge that has been gathered throughout the requirements analysis process. A requirement hierarchy leads to the creation of requirement taxonomy for conceptual and other design stages (Dufresne, 2008).

Weissman et al. used also a taxonomy approach to implement classification for device representation. They explain that taxonomy is a collection of predefined, hierarchically related subsystems. The classification of these subsystems is related to the keywords that the subsystems represent and certain criteria for the primary operating principle. They also advise organizations to develop their own taxonomies in order to adapt them to their own system and specific environment (Weissman et al., 2011).

On the other hand, Sanya I.O. and Shehab utilized an ontology approach. They state that ontology includes the taxonomy and relationships between its entities or components. They defined different kinds of main ontologies within the aerospace industry. These are product, process, resource and functional model ontologies. Here, they describe the product ontology (e.g. system ontology) as a product hierarchical breakdown of components. Process ontology includes the specific interdependent procedures and activities. Resource ontology was defined as specific assets, services, roles and toolsets. Finally, functional model ontology represents a hierarchical definition of concepts that describes geometry and analysis models generated as a result of employing specific resources (Sanya & Shehab, 2015).

Furthermore, Ast et al. state that ontologies are formal representations of knowledge that can be read by humans and machines. They are mechanisms that capture semantics, hence they can serve as a general semantic reference as well (Ast et al., 2013).

Interestingly, Soon Chong Johnson Lim et al., represented the usage of semantic annotation in the process of developing ontology or taxonomy for information extraction and retrieval. They used product information such as Bill of Materials (BOM), Product Data

Management (PDM) or Product Lifecycle Management (PLM) databases, catalogues, engineering texts, handbooks and others for extraction of product entities, concepts and the corresponding properties that a particular product constitutes (Lim et al., 2009). In other words they described the structuring of concepts by a “Top-Down” approach using extracted product information.

There are different ways to support extraction of relevant components, entities and understanding of the relationships for representing the platform used for defining taxonomy or ontology (Guarino, 1997).

In this research, a methodology is utilized that enables one to understand both the internal and external environments of a product or system (Zeng, 2015). The EBD methodology can support the process of extraction of components in both internal and external environments. EBD can also be helpful in characterizing the knowledge for basic semantic requirements ontology. One example of this research represents knowledge for ontology based on categories, taxonomies and relationships in the aerospace domain.

It is important to divide the process of creating knowledge for ontology into two general phases: characterization and implementation. Here, this research will focus on the characterization process. This process explores the main components and the components in their environments as they pass through lifecycle stages, and it defines the relevant relationships. These relationships can describe the interconnection between the components and possibility to structure the knowledge (e.g. requirement statements) in the main knowledge base.

2.7 Ontology

Staab S. et al. explain that ontology focuses on the nature and structure of things. Ontology is based on a hierarchy of concepts, i.e. taxonomies. Ontologies are most important in fields such as knowledge management, information integration, and retrieval (Staab et al., 2009).

Hai Wang and Shouhong Wang also consider that ontology is a tool for knowledge representation and sharing. They introduce it as a core of organizational knowledge with semantic relationships between concepts. They claim that the use of visualized and modularized ontologies can enable one to transform unstructured actions into structured tasks (H. Wang & Wang, 2016).

According to Gutierrez R., ontology can be used as a foundation to increase competitiveness in design and manufacture, and can serve a variety of innovative products. In his research he claims that ontology as an information technology product contributes to the actual knowledge base in two major aspects: 1) by establishing a common vocabulary in the context of requirements for system lifecycle processes, and 2) by creating a replicable ontology design process that can be extended to other domains of knowledge. He believes that the ontology approach might lead to economic prosperity (Gutierrez, 2018).

Following all of the above-mentioned arguments, the importance of high-quality ontologies is clearly understandable, especially in the knowledge-base domain and in the activity of knowledge sharing. Benefits could include improving the structure of knowledge (e.g. requirements statements) and leveraging it for future projects.

2.8 System requirements ontology

The knowledge management process that uses ontologies can also be helpful for design processes. It can provide essential support in the requirements analysis process by leveraging of requirements from the past projects.

Because the requirements analysis process is so fundamental in product design processes and leads to an understanding of the spectrum of constraints, it is essential to support this process with a good system requirements ontology that can structure the requirements and can ease its leveraging.

A system of uniform requirements categories can be represented by this ontology that may be able to describe the relationships between those categories and constitute the basics for semantic understanding.

Ontology defines a common vocabulary for its users and helps to share information across the domain. The reasons for using ontology are primarily to share a common understanding of the structure of information among people, to enable leveraging of domain knowledge, to make domain assumptions explicit, and to facilitate analysis of domain knowledge (Noy & McGuinness, 2001).

2.9 Aircraft Design Requirements Ontology (ADRO)

In today's competitive world, companies are required to develop and manufacture successful and marketable products. This makes the product development process one of the most important stages (Nilsson & Fagerström, 2006).

The development process in the aerospace domain is challenging. Li X. et al. explain that companies strive to reduce the cost of product research and development processes, and to minimize time-to-market. An aircraft is a complex system that has its own complex architecture, where any kind of customization could increase design workload and costs, and may require different tooling (Li et al., 2015).

The basic aircraft development process begins with gathering and analysis of system requirements. Here, the requirements describe characteristics that must be performed by a process, system, or component (Dufresne, 2008).

These requirement statements include the intended function of the product being designed, and the environment in which it will be used. They describe the constraints and the intended device behaviour. Any incompleteness in requirement statements, ambiguity, or inconsistency can lead to the problems within the different lifecycle stages of the product or system (Weissman et al., 2011).

According to Sanya I. and Shehab E. the knowledge management process within engineering design is becoming an exciting and important component in aerospace development. They explain that there is growing interest for creation and maintenance of engineering ontologies within the engineering community (Sanya & Shehab, 2015).

There are different types of ontologies, depending on their purpose. Top-level ontologies cover general and abstract concepts, domain or task-specific ontologies cover knowledge about a specific domain (e.g. aircraft) or a specific task, and other ontologies are typically developed for application purposes and with specific use scenarios in mind (Ast et al., 2013).

The main reason to develop an aircraft design requirements ontology is to structure and enable management of organizational knowledge (e.g. design requirements) in order to assist the aircraft and system architect, system engineers and other users in leveraging past design knowledge. This can be useful for creation of new scenarios, definition of new requirements and providing traceability at different levels.

The Aircraft Design Requirements Ontology (ADRO) can also be useful for aircraft design (conceptual, preliminary and detailed) and for maintenance domains; at the same time the production stage can also benefit from this proposed method.

There is no single correct way or procedure for developing or extending an ontology. In general, it is suggested to begin by defining the domain scope, terminology and class hierarchy. After defining the classes, the properties of each class and their constraints are specified (Sanya & Shehab, 2015).

Weissman et al. highlights also the importance of implementation of environment analysis throughout various lifecycle stages of the product within development of the system design requirements ontology. They explain that each category of the ontology also contains a number of objects, which are physical or abstract entities that interact with the system through its lifecycle stages. In addition, they state that the use of uniform requirement categories helps later in the searching process (Weissman et al., 2011).

For the current research, the EBD methodology was chosen for the purpose of developing knowledge for ADRO. This methodology was introduced in Section 2.3. It establishes a process of decomposing the system and its environment, then reorganizing the system environment components and relationships in a ROM diagram for representation of knowledge and relationships for ADRO.

2.10 Systems Engineering (SE)

One of the main goals of Systems Engineering (SE) discipline is to arrange the parts of the system in a way that system performance will be optimal. To achieve this goal, SE defines the system's requirements with respect to customer/user needs, with the objective that the product will not need to be redesigned. In addition, the focus is to make a product as reliable as possible and to achieve customer satisfaction (Jackson, 2015).

"Systems engineering" is defined as a methodical, multi-disciplinary approach for the design, realization, technical management, operations, and retirement of a system" (*NASA*, 2016) .

Figure 8 below represents the system engineering approach:

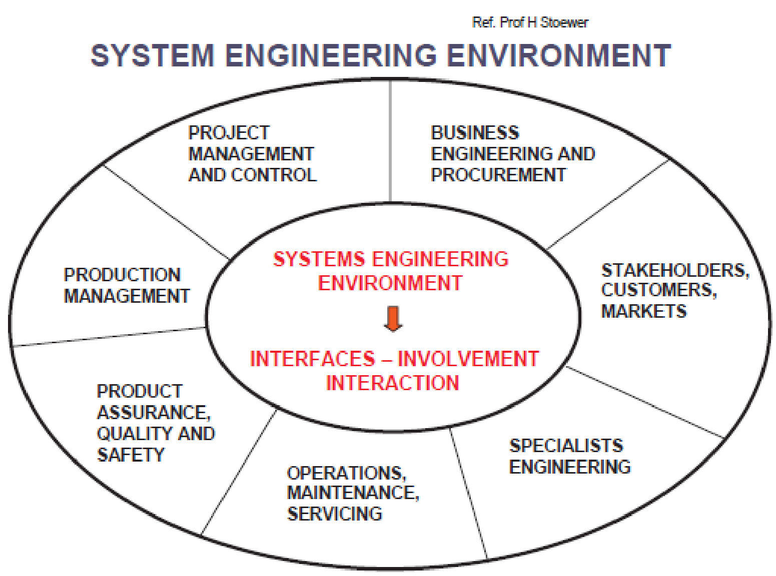

Figure 8: Context of Systems Engineering Technical Processes (INCOSE, 2006)

SE also has processes that provide transformation of requirements into specifications, architectures, and configuration baselines in a structured but flexible manner. It provides control and traceability to develop solutions that meet customer needs. The process may be repeated once or more often during any phase of the system development process. The system user's needs are emphasized because their needs generate the requirements for the system. Sometimes, basic user needs can effect all of the lifecycle functional areas. Then SE processes can generate and control the requirements that are based on these needs (Department of Defense, 2001).

2.11 System Architecture

The process of defining the architecture of complex systems usually involves decomposition of the system into smaller elements, such as subsystems, components and modules. In order to achieve the desired performance from the system, its elements must be integrated to work together. The SE domain is mainly responsible for planning and controlling the network responsible for interactions of the system's elements (Eppinger & Browning, 2012).

"*The System Architecture describes the entire system. It includes the physical architecture produced through design synthesis and adds the enabling products and services required for life cycle employment, support, and management*" (Department of Defense, 2001).

Development of the system architecture is a creative process. Here, intuition and experience can play an important role. Throughout the process of system architecture development, there is no unique solution to satisfy user requirements. Figure 9 below describes this by the simple example of "Alternative Architectural Concepts". The process depends upon the knowledge, experience, judgment skills, and intuition of the SE teams. In general, the system architecture is critical and it provides the framework for system development (INCOSE, 2006).

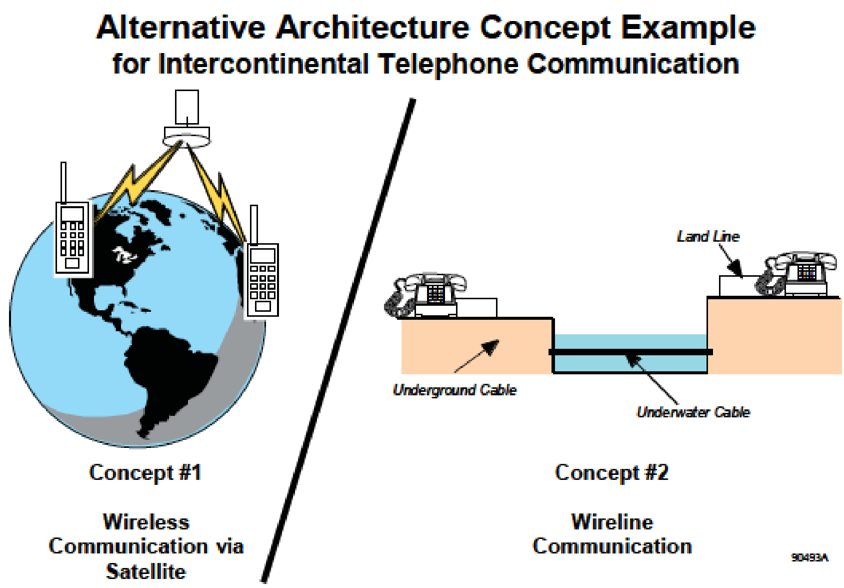

Figure 9: Example of Alternative Architectural Concepts (INCOSE, 2006)

Crawley et al. (2016), explain that systems architecture is an abstract description of the entities of a system and the relationships between them. The architected systems must meet stakeholder requirements, needs, and deliver good value. These systems integrate easily, evolve flexibly, and operate simply and reliably.

The system architecture identifies all the systems, subsystems, components, enabling systems and others that are necessary to support the complex system. It also determines the necessary processes for development, production, construction, deployment, operations, support, disposal, training, and verification (Department of Defense, 2001).

2.12 System and system of interest

There are different definitions of the system; for example, the Department of Defense (DOD) explains that a system is an integrated composite of people, products, and processes that provide a capability to satisfy a stated need or objective (Department of Defense, 2001).

On the other hand, the IEEE15288-2002 standard defines systems as man-made, created and utilized to provide services in defined environments for the benefit of users and other stakeholders (ISO/IEC 15288, 2002).

The system can be viewed in different levels; it can be a system that is a part of something larger or a product by itself. A product is considered herein as a representation of a particular system, or a system of interest that provides services to defined environments.

The environment of the systems or the system itself can interact or integrate with one or more different components such as: hardware, software, humans, processes, procedures, facilities and naturally occurring entities (e.g. water, organisms, and minerals). The definition of a particular system, its architecture and components depend on an observer's interests and responsibilities. One's perception of a system-of-interest can be as a system component in another system-of-interest, or as a component as part of the operational environment of another system of interest (ISO/IEC 15288, 2002).

Figure 10 below represents a system of interest in an operational environment. Here highlighted are the presence of the system of interest itself, the enabling systems, and other systems in operational environment. A view of the interactions between them is also shown.

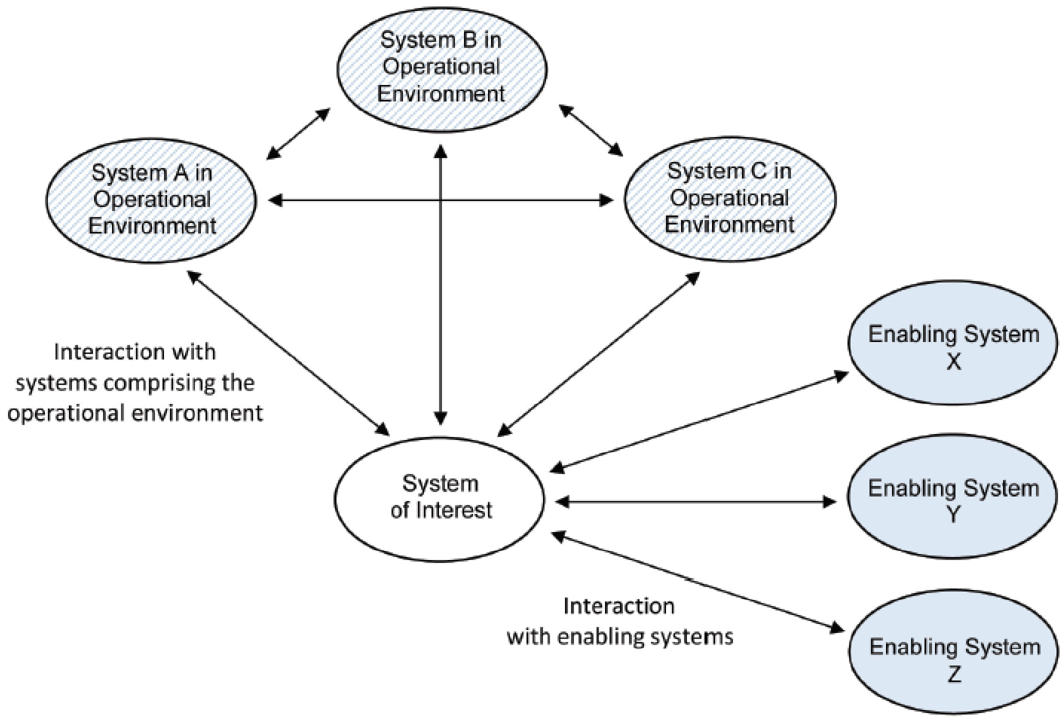

Figure 10: System-of-interest, its operational environment and enabling systems (ISO/IEC/IEEE15288, 2015)

The enabled systems here are the systems that facilitate progression of the system of interest through its lifecycle. Those systems are not necessarily a part of the system of interest or its operational environment. Enabled systems provide essential services for systems of interest throughout their lifecycle stages. Examples of enabled systems include mass-production, and training and maintenance systems (ISO/IEC/IEEE15288, 2015).

Another example of more complicated systems-of-interest is represented in Figure 11. This is an example of the multitude of perceivable systems-of-interest in an aircraft and its environment of operation within a transport system-of-systems (ISO/IEC 15288, 2002).

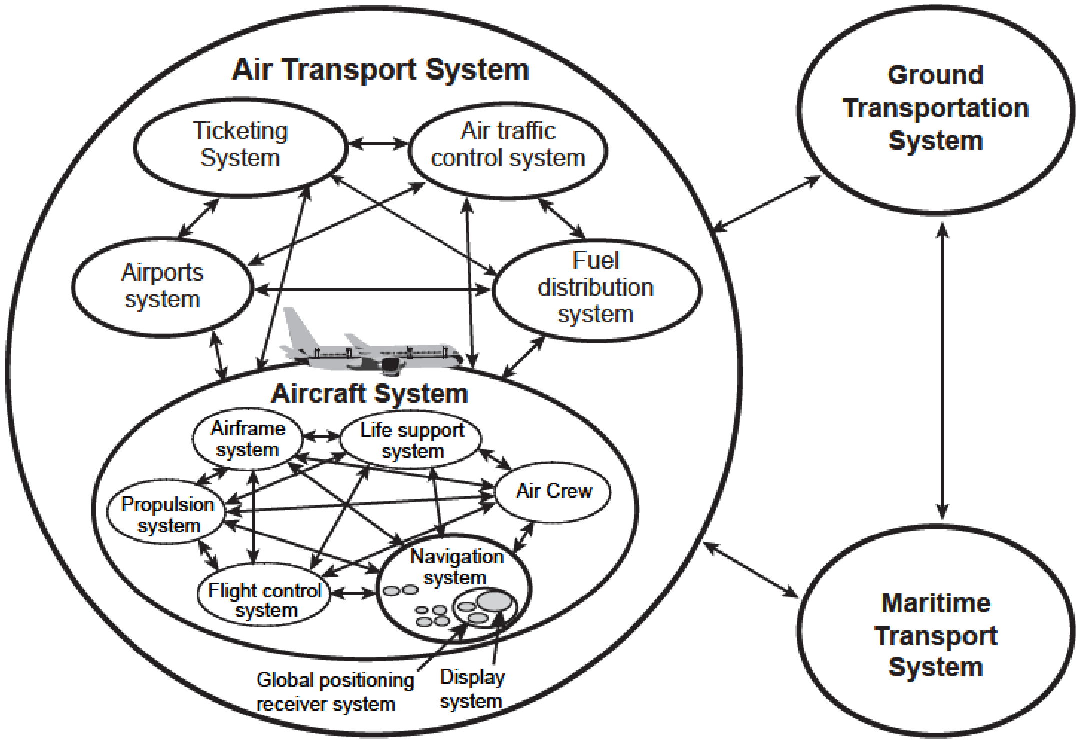

Figure 11: Example of the multitude of perceivable systems-of-interest in an aircraft and its environment of operation within a transport system-of-systems (INCOSE, 2006 & ISO/IEC 15288, 2002)

The example represented in Figure 11 above can highlight in general the complexity of the systems and their operational environments. This emphasizes the importance decomposing the general desires of customers to their particular product (e.g. system-of-interest) and its environment. The decomposition process leads to the generation of the system requirements. Herein, these requirements characterize the future system.

2.13 System Lifecycle

The process of decomposing the general desires of customers may have an influence on the system in the future. This can affect the system, its components or the system environment in different stages of the system lifecycle. This lifecycle includes the period on the timeline from the early steps of definition to utilization and retirement of the system.

INCOSE explains that: *"Every man-made system has a life cycle, even if it is not formally defined. In keeping with increased awareness of environmental issues, the life cycle for any system-of-interest must encompass not only the development, production, and usage stages but also provide early focus on the retirement stage when decommissioning and disposal of the system will occur"* (INCOSE, 2006).

The lifecycle stages can vary according to the nature, purpose, use and prevailing circumstances of the system. Nevertheless, there is an underlying, essential set of characteristic lifecycle stages that exists in the complete lifecycle of any system. Each lifecycle stage has a specific purpose and contribution. Each stage is considered at the appropriate point during the planning and executing of the system lifecycle. These stages describe in general the progress and achievement milestones of the system through its lifecycle (ISO/IEC 15288, 2002).

Table 6 below represents the main lifecycle stages of a system, their purpose and possible decisions of the project thorough its timeline.

LIFE CYCLE STAGES	PURPOSE	DECISION GATES
CONCEPT	*Identify stakeholders' needs* *Explore concepts* *Propose viable solutions*	*Decision Options* *– Execute next stage* *– Continue this stage* *– Go to a preceding stage* *– Hold project activity* *– Terminate project*
DEVELOPMENT	*Refine system requirements* *Create solution description* *Build system* *Verify and validate system*	
PRODUCTION	Produce systems Inspect and test	
UTILIZATION	*Operate system to satisfy users' needs*	
SUPPORT	*Provide sustained system capability*	
RETIREMENT	*Store, archive, or dispose of the system*	

Table 6: An example of stages, their purposes and major decisions gates (ISO/IEC 15288, 2002)

2.14 Stakeholder definition

Throughout all stages of the system's lifecycle, there are participants that are influenced directly or indirectly by the system and its environment. These are the stakeholders, as described by ISO/IEC 15288 (2002): *"Stakeholder is a party having a right, share or claim in a system or in its possession of characteristics that meet that party's needs and expectations"* (ISO/IEC 15288, 2002).

There are various stakeholders; they have different types of stakes in the decisions made during the development process of the product (i.e. system of interest). There can be internal and external stakeholders. The external stakeholders include users/customers, distributors, governments, suppliers, communities, laws and regulations. The internal stakeholders include management, marketing experts, designers, purchasing, manufacturing, assembly and sales. It is important for the design team to recognize all the stakeholders. In this way the design team can more easily share information about different requirements and design objectives. It is very important to ensure that all team members have a common vision for the product, its functionality and performance in the early phases of product lifecycle (Nilsson & Fagerström, 2006).

2.15 Requirements and Needs

Within the framework of each lifecycle stage of the system, the system engineers, marketing group, and other specialists gather and map information and desires of potential customers and other stakeholders. These desires are translated further to specific requirements and needs. Those are essential for product characterization and development.

Jackson S. explains that a requirement is a statement of required performance or design constraint to which a product must conform. The requirements must be verifiable and they should be applied to the people, products, and processes (Jackson, 2015).

The requirements analysis process is the key process used in defining the project boundaries. This process includes the definition of customer needs and objectives with consideration of planned customer use, system characteristics, and environment analysis, for determining requirements for system functions. It is an iterative process that strives for optimization of performance, for identification of functions, for the synthesis of both in the product, and for verification that the customer requirements are satisfied (Department of Defense, 2001).

Figure 12 represents the progression of the requirements and requirements analysis process. The process begins from acquisition of information from external sources that transitions to internal processes of requirements analysis and implementation.

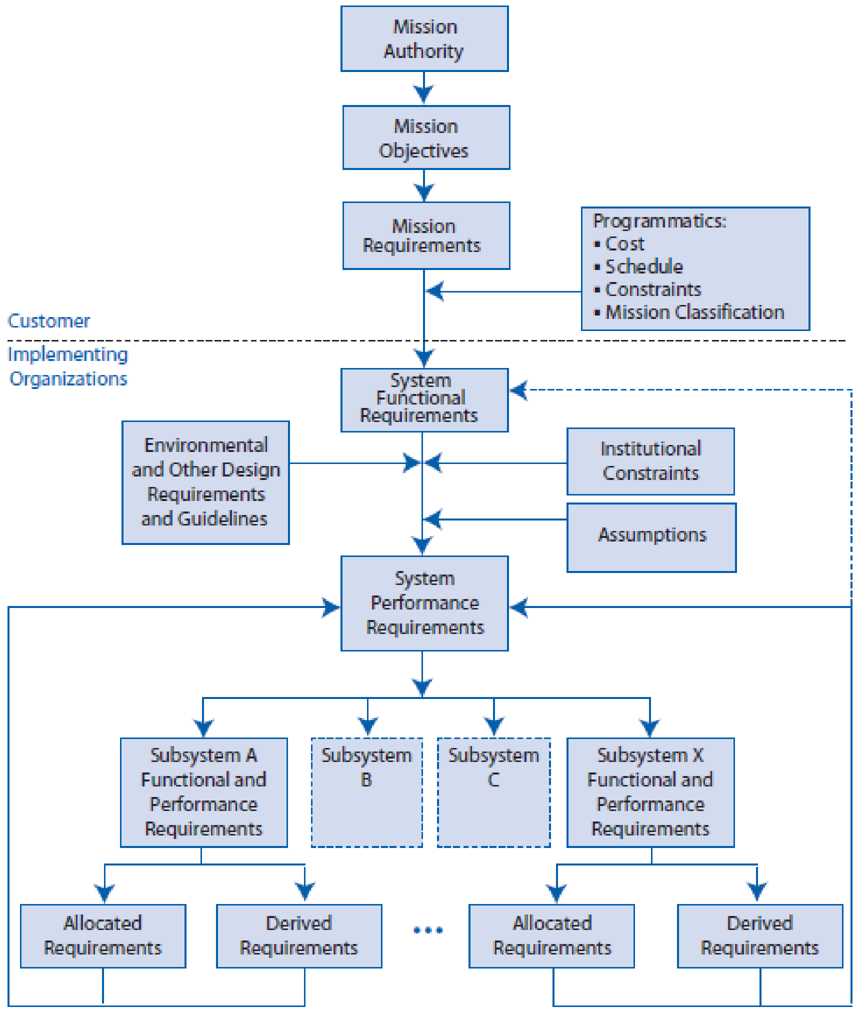

Figure 12: The flowdown of requirements (NASA, 2007)

Requirements management and analysis processes relate to all stakeholder expectations, customer requirements, and technical product requirements. These processes transfer down to the lowest level product component requirements. They are used to manage the product requirements that are identified, baselined, and occur in the definition of the WBS (Work Breakdown Structure) model products during system design. They provide bidirectional traceability to the initial requirements and help manage any changes to established requirement baselines over the life cycle of the system products (NASA, 2007).

The requirements analysis process is used to develop functional and performance requirements. Here, customer requirements are translated into a set of requirements that define what the system must do and how well it must perform. This process clarifies and defines the functional requirements and design constraints. Then, the functional requirements define quantity, quality, coverage, time lines and availability (Department of Defense, 2001).

2.16 Functions and Function Allocation

The functional requirements that were discussed in Section 2.6 translate further in the process to the functions that the system will acquire. As explained by Jackson S., a function is a description of what a system element does. There are functions that are allocated to the system, subsystems and components of the system of interest throughout its lifecycle stages with consideration to its environment. Considered here are activities performed by the developer and users of the aircraft from the moment of conception to its disposal. This means that these functions are not only performed by the aircraft itself; they can be performed by interaction of the system and humans (or others) in the environment (Jackson, 2015).

Functions must be analyzed. This analysis includes decomposition of higher level functions, identified through requirements analysis, into lower-level functions. In this way, the performance requirements associated with a higher level are allocated to lower-level functions. As a result, this process represents a description of the system or component in terms of what it does logically and in terms of the performance required. This description is often called the functional architecture (Department of Defense, 2001).

Figure 13 represents the system engineering process and focuses on explanation of the flow-down process from requirements acquisition to system development stage.

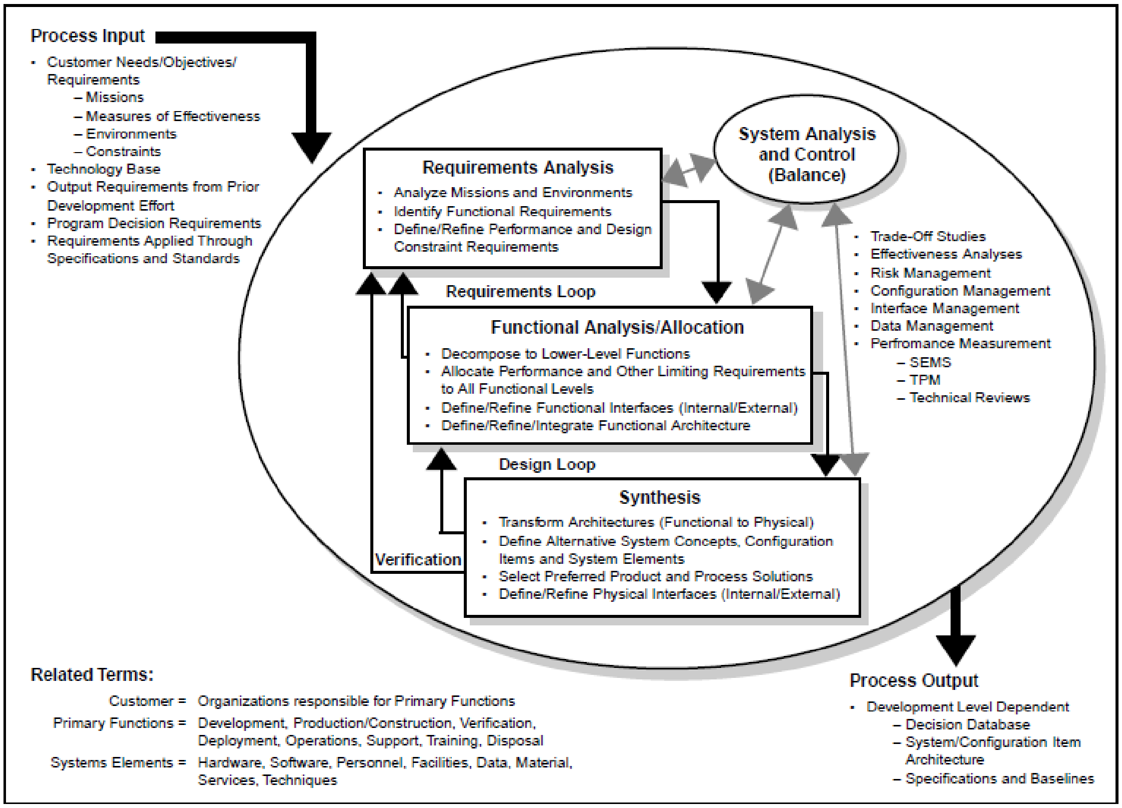

Figure 13: The Systems Engineering Process (Department of Defense, 2001)

To satisfy their future stakeholders, companies try to identify their requirements and functions. New functions can be included in the model at the appropriate level in the hierarchy. By adding these functions, the path will be updated at relevant lower levels in the functional decomposition. Adding new functions will allow for the exploration of what effects are associated with introducing the new function. By doing this, it will be possible to trace backwards and understand what requirements and stakeholders will be affected by the new function (Nilsson & Fagerström, 2006).

2.17 Requirements and scenario interconnection

A review of the requirements and requirements analysis was given in Section 2.4. In this section the relationships between customer requirements, operational requirements, and scenarios will be discussed.

The Department of Defense explains that customer requirements are statements of facts and assumptions that define the expectations of the system in terms of mission objectives, environment, constraints, and measures of effectiveness and suitability. Therein it is defined that the key customer of the system is an operator. It then follows that operational requirements define the basic needs that must be satisfied (Department of Defense, 2001).

Table 7 below gives the basic questions to guide the creation of operational requirements. Here, the emphasis is on understanding system performance within its missions in different environments and operating conditions.

Operational distribution or deployment: Where will the system be used?

Mission profile or scenario: How will the system accomplish its mission objective?

Performance and related parameters: What are the critical system parameters to accomplish the mission?

Utilization environments: How are the various system components to be used?

Effectiveness requirements: How effective or efficient must the system be in performing its mission?

Operational life cycle: How long will the system be in use by the user?

Environment: What environments will the system be expected to operate in an effective manner?

Table 7: Operational Requirements – Basic Questions (Department of Defense, 2001)

These questions also support creation of scenarios that further can be the basis for defining specific requirements for the system of interest or its components.

For example, operational scenarios are very important especially within the conceptual stage. These scenarios are a detailed and sequential description of how the system should operate and interact with its users and components in their environment. They should be described for all operational modes, mission phases (e.g., installation, startup, typical examples of normal and contingency operations, shutdown, and maintenance). The scenarios should include all relevant information, events, conditions (nominal, off-nominal, and stressful), interactions and actions for comprehensive understanding of the operational aspects of the system (NASA, 2007).

Utilization of scenarios is essential in system development processes and an important definition for "Concept of Operations" (ConOps). INCOSE explains that the Concept of Operations describes the way a system works from the operator's perspective (INCOSE, 2006). Figure 14 shows a general overview for the process of generating a Concept of Operations, from customer/ stakeholder expectations and needs to conceptualization.

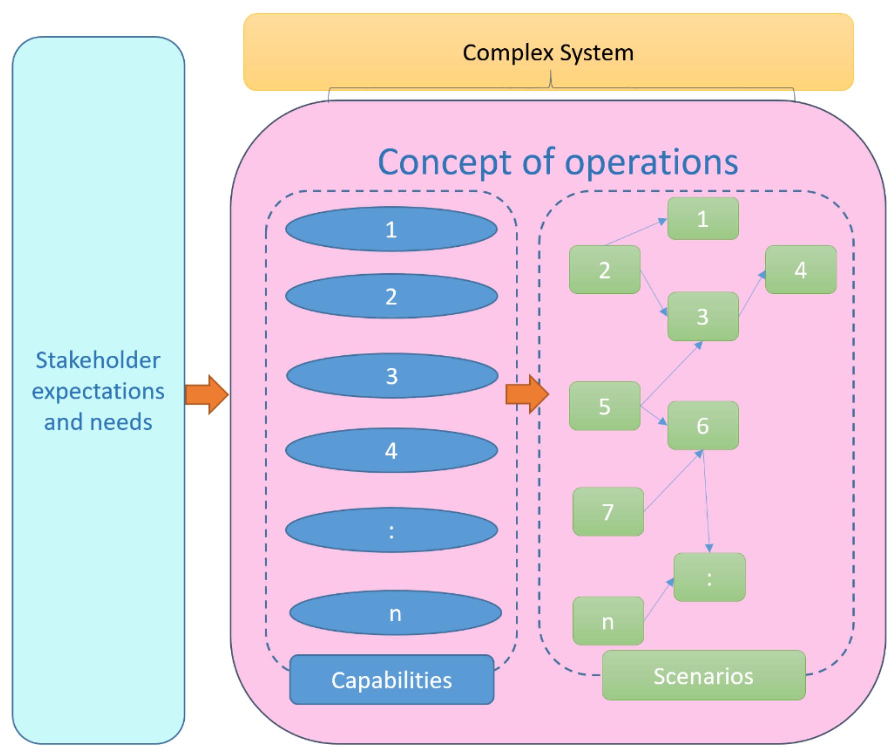

Figure 14: The process of generation Concept of Operations

Herein, the upper level of requirements, expectation and needs are being translated to general system requirements, and then to system possible capabilities. The analysis of possible capabilities leads to generation of many kinds of scenarios. By continual and iterative analyses, this process provides boundaries for the Concept of Operation for the system.

"The ConOps describes how the system will be operated during the life-cycle phases to meet stakeholder expectations. It describes the system characteristics from an operational perspective and helps facilitate an understanding of the system goals. It stimulates the development of the requirements and architecture related to the user elements of the system. It serves as the basis for subsequent definition documents and provides the foundation for the long-range operational planning activities" (*NASA*, 2007).

2.18 Organizational knowledge and possibility for re-use

The fundamental step in any domain is to generate and introduce novel and attractive products to the competitive market (Nilsson & Fagerström, 2006). These products can be tangible or intangible (Kahn, 2010), they can include a system or a service. Of importance here is not only the type of product that will be produced, but in understanding the needs, the expectations, the requirements of the market, and the domain customers. For example, the aircraft conceptual design domain produces complicated systems, even system of systems. These kinds of products can have a significant effect on success or failure within an organization. In general, each project has some amount of risk. The challenge is to define a product (e.g. a system), to meet overall requirements, to minimize the risk, and to achieve the highest chances of project success (INCOSE, 2006).

Meanwhile, the market is changing rapidly and this has its effects on organizational strategy. What may be a winning strategy at a certain time may not be relevant at all shortly thereafter (Aaker, 2008). These changes challenge organizations to create new products, with new features and new technologies. They also lead organizations to strive to improve the ability to retain their knowledge and reuse it in new projects. Utilization of existing knowledge can make the development process more productive and efficient, thus minimizing the timeline of development and production stages.

Wu et al. have explained that: "*Product development is a highly creative and knowledge-intensive process that involves extensive information, and knowledge exchange and sharing, among geographically distributed teams and developers. How to best integrate such heterogeneous product knowledge has become an extremely important knowledge management (KM) subject associated with product development*" (Wu et al., 2014).

Herein, the requirement statements are a part of organizational knowledge. This knowledge can represent the uniqueness of the product and might have a specific effect on organizational involvement in the global market. For this and other reasons, it is important to manage this knowledge (e.g. requirements). Having structured product requirements will contribute to the stability of an organization's knowledge platform.

Management of these requirements plays an essential part in product development. There may be serious consequences if any critical product requirements are missing in the early product development stages, hence many companies invest in effective approaches for managing product requirements (Chen & Zeng, 2006).

Alexander Weissman et al., in their research discuss the requirements authoring process. They explain that a designer can choose to write the requirements from scratch, and this process requires deep knowledge about how the product should behave within technical, economic, and social domains. Another approach is to reutilize requirement statements from prior designed products. They believe that reutilizing requirement statements is a possible approach for ensuring completeness and it is possible to reduce authoring times by doing so. They assert that the use of uniform requirement categories can further help to narrow the scope of the search based on the relevant stages of the product life cycle (Weissman et al., 2011).

The uniform requirements categories can be represented by an ontology that may able to describe the relationships between those categories and constitute the basics for semantic understanding.

Hai Wang and Shouhong Wang also consider that ontology is a tool for knowledge representation and sharing. They introduce it as a core of organizational knowledge with semantic relationships between concepts. They claim that understanding the ontology can enhance the ability to transform unstructured actions into structured tasks through the usage of visualized and modularized ontologies (H. Wang & Wang, 2016).

2.19 Models, model purpose and knowledge sharing

There are different kinds of requirements that are in use within system engineering processes. Sections 2.6 and 2.8 have represented the importance of the requirements and their analysis, transformation, allocation, and utilization.

Management of the requirements and other organizational knowledge is accomplished worldwide in different ways. There is a documentation approach using textual files, and a new approach to documentation management by modeling using digital knowledge-based systems.

DoD defines that a model is a physical, mathematical, or logical representation of a system entity, phenomenon, or process (Department of Defense, 2001).

"The model can form an essential support for this co-engineering between specialties, each bringing its own constraints to the needs, checking that they are met in the solution, and so helping the architect to evaluate the merits of each alternative architecture" (Voirin, 2018).

A modeling approach makes the problem analysis process more structured and uses consistent language and terminology for describing the problem and its solution. This process produces coherence in the design and verifiably for all the system requirements and solutions. In addition, the modeling approach provides an opportunity for easier understanding of existing solutions, supporting processes and technologies. This can then provide support for new designs and design improvements for meeting new requirements (Long & Scott, 2011).

Peter M. supports the modeling approach and provides a list: Understanding of the natural reality of the existing system or phenomena; Prediction of some properties; Management or control properties of the system or phenomena; Understanding of the existing human model and prediction of some of its properties; Better understanding, management and prediction of future artificial systems; Providing a locus for discussion between relevant stakeholders; Supporting capability of identification, articulation and potential resolution of alternative action options and alternative trade-offs; Making possible a structure way of thinking; Training participants (McBurney, 2012).

"A model can take many forms, for example, a system of equations describing the trajectory of a planet, a group of rules governing the treatment of an information flux, or a plastic model of an airplane. An interesting element of the definition is the fact that a model is "constructed for a given objective": there is no single unique model for a system. A system model that is entirely appropriate for one purpose may be completely useless for another purpose" (Cantot & Luzeaux, 2011).

Models are usually shared within different domains in the organization. This is a type of knowledge communication to a set of stakeholders. Different models represent the system from different perspectives and manners (Fosse, 2012).

Modeling helps produce improved design quality, to decrease ambiguity, to increase precision, to support evaluation of consistency, correctness and completeness. This approach increases the ability to manage system complexity, and supports traceability and impact/change analysis. It is also very helpful in validation and verification processes (Hart, 2015).

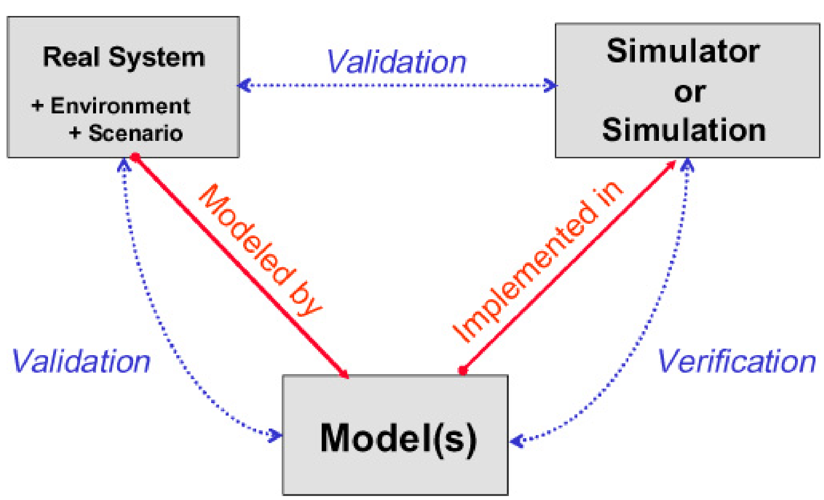

Figure 15: General principle of simulation (Cantot & Luzeaux, 2011).

Figure 15 above represents the general principle of simulation by using models. A real system, with its environment and a usage scenario, leads to an abstraction, the model, which is then made concrete by a simulation application (Cantot & Luzeaux, 2011).

2.20 Model-Based Systems Engineering (MBSE)

Section 2.19 introduces the concept of models. Models are used widely by different domains. For example, they are essential for systems engineering. Here they provide the foundation for simulation, validation, verification and other processes. In addition, they support the possibility of information and knowledge sharing.

Information and knowledge sharing or utilization includes archival aspects. Archives provide the opportunity for reuse of information by different groups and domains within the organization. Design information and knowledge (e.g. requirements, specifications, and other documents) should be archived and managed in a specific manner, by using traceability, sequences and relationship connections between the documents.

For this application, many organizations have begun to implement the Model-Based Systems Engineering (MBSE) approach. MBSE is defined by INCOSE as a formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning with the conceptual design phase and continuing throughout development and later life cycle phases (Technical Operations - INCOSE, 2007). This approach enables different domains in the organization to be connected so that they work using a single source of validated information. The benefits of moving from a document-centric approach to a modeled one include better communication of information and knowledge within relevant domains in the organization, high level validation of data, automatic report preparation, and easy performance of validation or verification steps.

Figure 16 below represents the transition from document to model-centric approach:

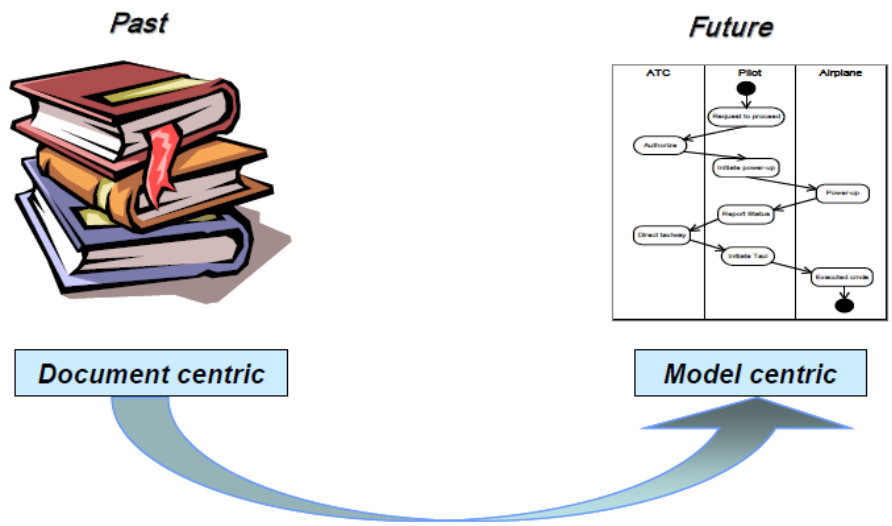

Figure 16: Transition from document-centric to model-centric systems engineering (Technical Operations - INCOSE, 2007)

Laura E. Hart explains that MBSE provides a mechanism for driving more systems engineering depth without increasing costs. The model-centric approach enables automation and optimization, allowing systems engineers to focus on value-added tasks and ensure that a balanced approach is taken. A better understanding of the systems can be achieved through integrated analytics, tied to a model-centric technical baseline (Hart, 2015).

Likewise, Long D. and Scott Z. claim that MBSE is fundamentally a thought process. This provides a framework to allow the systems engineering team to be effective and consistent immediately from the beginning of any project. At the same time, this process is flexible enough that it allows the "thought" process to adapt to special constraints or circumstances present in the problem (Long & Scott, 2011).

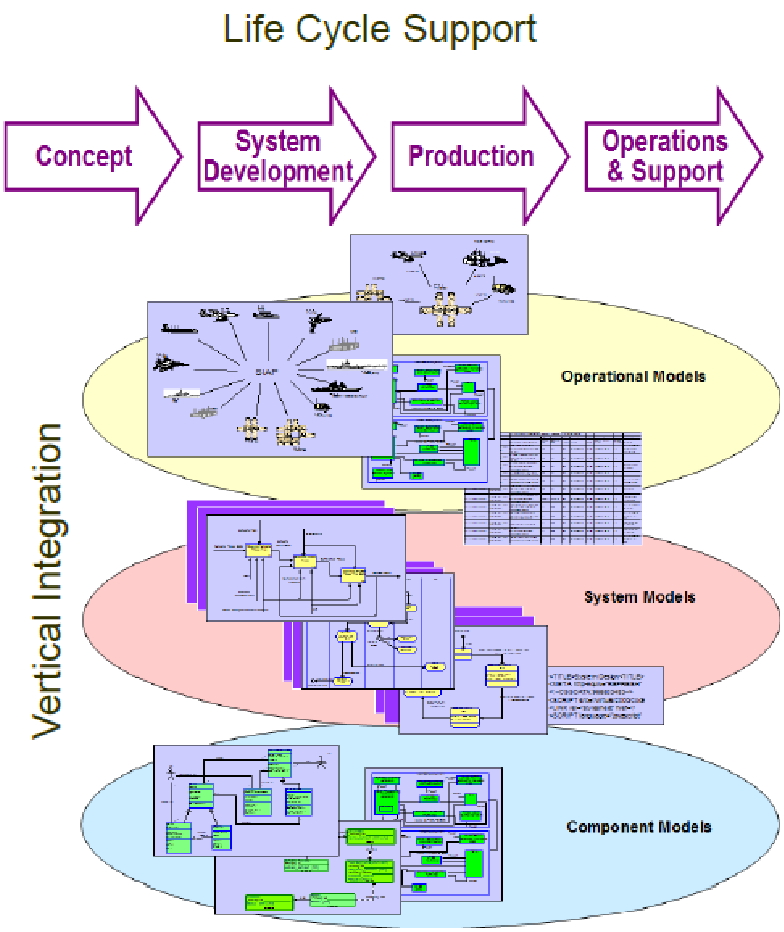

Figure 17: Model-based Systems Engineering (MBSE) (Friedenthal & Wolfrom, 2010)

This approach supports systems development processes through the use of models. In addition, MBSE also connects the models from the system-of-systems level vertically to the bottom levels that comprise components (see Figure 17). MBSE improves productivity and quality; it lowers risks, communicates among system/project stakeholders and manages complexity (Friedenthal & Wolfrom, 2010).

A clear definition of the structure is very important here for effective implementation of the MBSE approach. This structure can propose the basic rules for classification of the organizational knowledge and to provide the possibility for ontology characterization and identification of relationships. An appropriate knowledge for ontology will be defined in this research in order to ease the process of MBSE implementation. The knowledge for ontology will be based on the performance of the analysis by using the EBD methodology through different stages of the system life cycle and its environments. By using the knowledge for ontology or taxonomies that will be defined, it will facilitate the process of leveraging past knowledge.

2.21 Summary of the literature review

The design process of complex systems is complicated. This process involves communication among different domains. It requires a deep understanding of, among others, existing technologies, and the requirements and needs of the stakeholders. It is a highly creative and knowledge-intensive process that involves extensive information and knowledge exchange among teams and developers. The integration of product knowledge has become an extremely important issue, and is known as knowledge management (KM). Product development knowledge integration and sharing is becoming a key issue in large organizations (Wu et al., 2014).

There is a belief that structured knowledge (of requirements, designs, etc.) could be helpful in product development processes. This structured and computer-based knowledge is essential in sharing between departments of the manufacturing organizations. It is useful in preventing unnecessary design revisions and reduces mistakes (Anjum et al., 2013); it enables the creation of opportunities for new designs and improves communication within the organization.

In addition, computer-based knowledge management can be improved by utilisation of knowledge ontology. This ontology can support human-web interaction as a key component of knowledge sharing (H. Wang & Wang, 2016). It can also provide a basis for implementation of MBSE (Model-Based System Engineering).

Herein, MBSE provides a shared system model with discipline-specific models that include their characteristic information in a mathematical format. In this way all disciplines that participate in development of a project can "view" a consistent system model (Fosse, 2012).

CHAPTER 3 EBD enabled scenario analysis

3.1 Introduction

The first objective of this chapter is to specify the knowledge components and relationships for the ontology diagram (Sections 3.1 and 3.2); the second is to represent the tagging process based on the requirements analysis (Sections 3.3 through 3.4). Here, an analysis based on the procedures of the EBD methodology will be performed. Furthermore, this analysis will be summarized and evaluated versus the modeling approach.

The analysis begins from an overview of transportation system in general, and step-by-step continues to the specific domain of aerospace (e.g. aircraft). The basic analysis for the transportation system will provide boundaries and following which the focus will be on the specific environment of the aerospace industry.

Transportation systems are usually designed with five key capabilities, as shown in Figure 18, which provide basic expected outcomes and meet customers' expectations and needs.

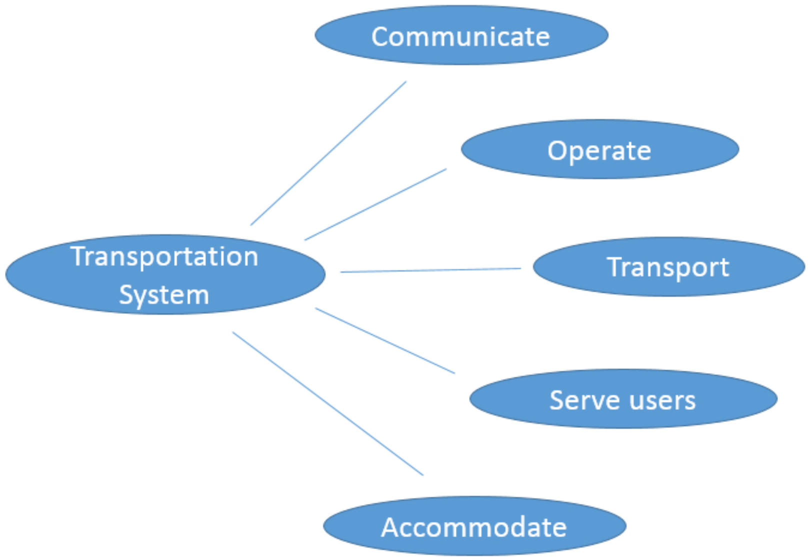

Figure 18: Transportation system - general capabilities

Each individual system also has its environment that includes different actors who use or provide service to the system and interact with it. As an example, Figure 19 shows the main actors in the aircraft environment and their possible interactions with this system. Table 8 lists typical questions and answers for each capability of an aircraft, following the EBD methodology, which provides basic information on the characteristics of the system environment. Such information will then be considered in the development of knowledge for the Aircraft Design Requirements Ontology (ADRO).

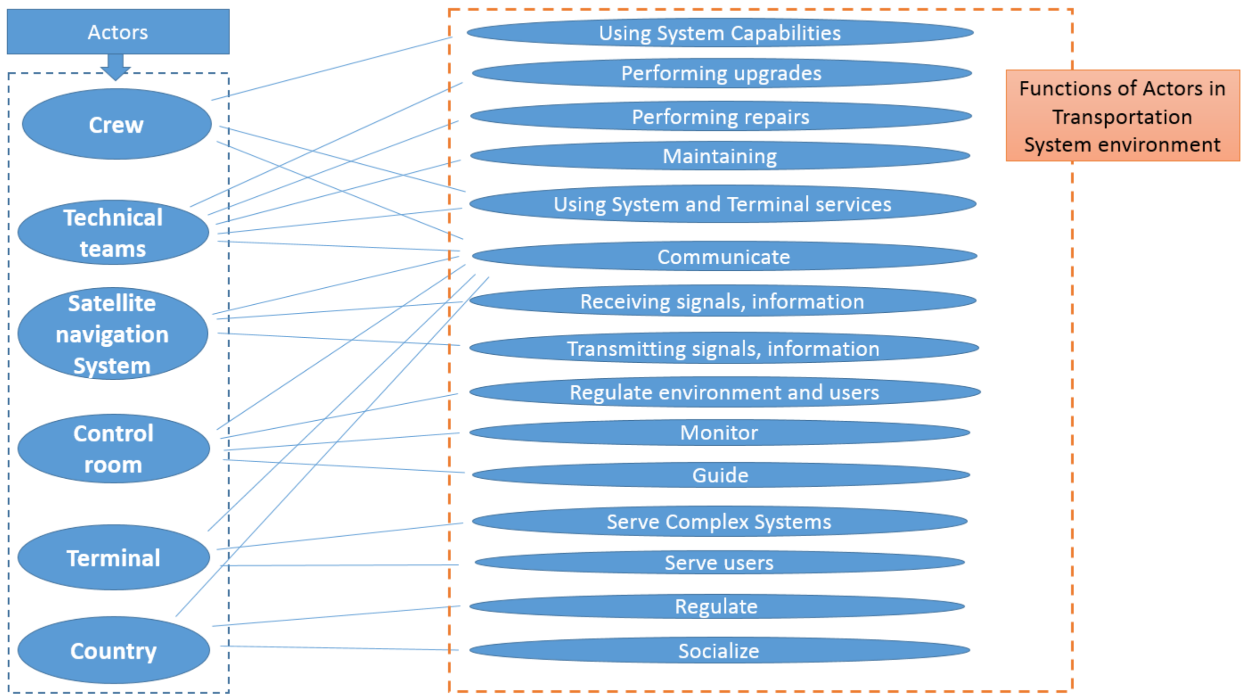

Figure 19: Main actors and general interactions with the system and its environment

Capabilities	Questions	Answers
Communicate	Why?	To be more coordinated. To comply with regulations (safer). To be aware of your surroundings and to provide information.
	How to?	By navigation satellite system, radio or cellular systems. By datalink and ICAS Crew Alerting System message to pilot. By encoding and decoding (digital data, written text, speech, pictures and gestures). By interpretation (verbal communication). By visual channels.
Operate	Why?	To transport humans, animals and goods from location A to location B.
	How to?	According to the way the vehicles are operated on established routes. “Regulated airspace”.
Transport	Why?	In order to meet users and stakeholders needs.
	How to?	By air, land (rail + road), water, space. With: automobiles, buses, trains, trucks, helicopters, aircrafts, watercrafts and space crafts.
Serve users	Why?	To provide or to meet/match the user needs.

	How to?	By assistance, training, support, responding, etc.
Accommodate	Why?	To transport humans, animals and goods from location A to location B.
	How to?	By plans, procedures, etc. By providing environmental condition for safe and comfortable travel.

Table 8: General analysis of environment for transportation system

3.1 ADRO knowledge development

Different methodologies support an extraction of relevant components, entities and understanding of the relationships for representing the platform for definition of taxonomy or ontology (Guarino, 1997). Interestingly, Lim S.C.J.et al., represent the usage of semantic annotation in the process of developing ontology or taxonomy for information extraction and retrieval. They used product information such as Bill of Materials (BOM), Product Data Management (PDM) and Product Lifecycle Management (PLM) databases, catalogs, engineering texts, handbooks and others for extraction of product entities, concepts and their corresponding properties that a particular product constitutes (Lim et al., 2009). In other words they describe the structuring of concepts by a "Top-Down" approach using extracted product information.

Here, in the research utilized an advanced method to understand the internal and external environments of the system. Environment-Based Design (EBD) methodology can support the process of components extraction in both environments; this methodology can be helpful in characterizing knowledge for the system design requirements ontology. This includes categories, taxonomies and relationships for the system in the aerospace domain. The main components and the components in their environments will be explored through passing lifecycle stages of each main constrained word and definition of their relationships. These relationships will describe the interconnection between the components and how to structure the knowledge (including requirement statements and data) in the main knowledge base.

3.1.1 Analysis of the general problem statement

As the first step in the EBD methodology, the environment analysis begins with general statement: "EBD enabled Scenario Analysis for Aerospace Product Development". The Figure 20 represents the general statement of the problem explored in the thesis by using a ROM diagram and its symbols (see Table1) as a linguistic representation of its components and relationships. Based on the ROM diagram, the environment analysis can be initialized by asking generic questions. These questions provide insight to one's understanding of the general environment of the main components in the problem definition.

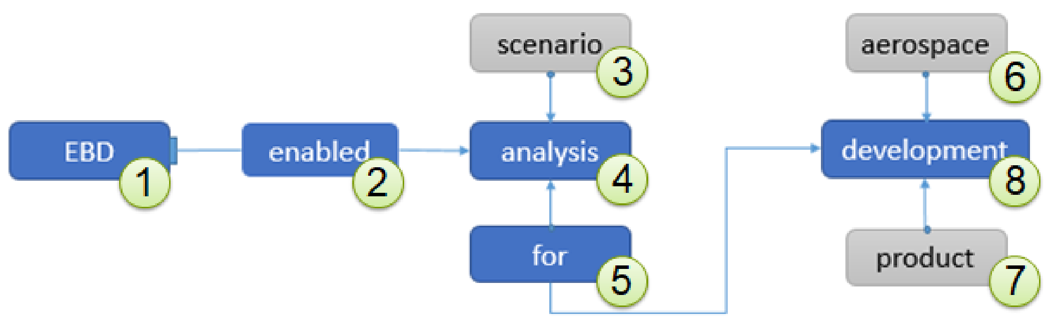

Figure 20: Recursive object model (ROM) for the general statement of the problem

The ROM diagram shows the main and most constrained components. Based on number of arrows, they are: "**Analysis**" and "**Development**". The next step in the EBD methodology is defining the object list for questioning. Table 9 represents the relations between the components of the ROM diagram following the rules presented in Figure 21.

$$r_{ij} = \begin{cases} 1 & \text{objects } i \text{ and } j \text{ have a subject - verb relation} \\ 2 & \text{objects } i \text{ and } j \text{ have a verb - object relation} \\ 3 & \text{objects } i \text{ and } j \text{ have a constraint relation from object i to object j} \\ 0 & \text{otherwise} \end{cases}$$

Figure 21: Rules for the relations between objects in the ROM diagram (Zeng, 2014)

Object	1	2	3	4	5	6	7	8
1		1						
2	1			2				
3				3				
4		2						
5				3				2
6								3
7								3
8					2			

Table 9: Relations between the objects/components of the ROM diagram

The number of relations between the objects /components of the ROM diagram can be summarized as follows:

Number of relations	3	2	1
Objects	4, 8	2	1, 5

Table 10: Number of constraint and predicate relations on an object

From Table 10 it is clear that the central objects are 4 and 8. These objects have a higher number of constraints and predicate relations. Figure 22 below illustrates the structure of the objects around the central objects 4 and 8 in terms of constraint and predicate relations derived from the ROM diagram that is represented in Figure 20.

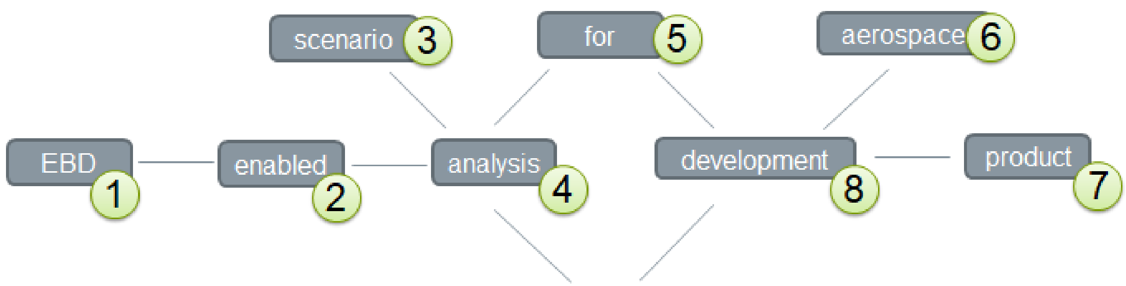

Figure 22: Structure tree of constraint and prediction relations of objects

Following Figure 22 and the rules of the EBD methodology, an object list for questioning can be defined. Table 11 represents this list:

Questioning object list	1; 3; 6; 7; 1&2; 4&3; 8&6; 8&7; 1&2&4 ; 2&3&4; 3&4&5&6&7&8;

Table 11: Object list for questions generation

Based on this, Table 12 summarizes the generic questions that were defined by the rules and templates described in Section 2.3 of the EBD methodology and environment analysis. These questions give the boundaries but do not limit in searching for information that is of interest. Table 12 only provides a focus on a specific stream of information.

#	Component/s	Question
1	1	T1: What is an EBD?
2	3	T1: What is a scenario?
3	6	T1: What is an aerospace?

4	7	T1: What is a product?
5	1 & 2	T6: What is enabled by EBD?
6	4 & 3	T3: What is analysis? What is scenario analysis?
7	8 & 6	T3: What is development? What is development in aerospace?
8	8 & 7	T3: What is development of product? What is a product?
9	1 & 2 & 4	T5: What do you mean by enabled in the statement "EBD enabled analysis"? How / Why / When / Where EBD enabled analysis?
10	2 & 3 & 4	T5: What do you mean by enabled in the statement "enabled scenario analysis"? How/ Why/ When/ Where enabled scenario analysis?
11	3 & 4 & 5 & 6 & 7 & 8	T5: What do you mean by analysis in the statement "scenario analysis for aerospace product development"? How/ Why/ When/ Where is scenario analysis for aerospace product development?

Table 12: Generic question representation for environment of the main and constrain words of the statement

Following this, for the question definition step, the answers can be derived by searching for the information to answer the following questions:

1. What is an EBD?

   The EBD leads that design change an existing environment to a desired one by generating a new artefact. (Zeng, 2015).

2. What is a scenario?

   A "scenario" is often used to describe a behaviour. Scenarios and what-if thinking are essential tools for designers who must cope with the uncertainty of the future (Walden et al., 2015).

   For example, with operational scenarios, there would be a step-by-step description of how the system should operate and interact with its users and with the components in the system's environment. Operational scenarios include all operational modes, mission phases (e.g., installation, start-up, nominal, off-nominal, stressful conditions, shutdown, and maintenance), and critical sequences of activities for all classes of users identified. The scenario includes events, actions, stimuli, information, and interactions as appropriate to provide a comprehensive understanding of the operational aspects of the system. Operational scenarios should span all of the specific conditions (e.g. nominal, off-nominal, and stressful) that could occur during the mission. (NASA, 2007).

3. What is an aerospace?

"Aerospace – producing or operating aircraft or spacecraft" (Dictionary, 2020).

Designers, manufacturers, operators, ground support, and maintainers are considered as critical components in the aerospace domain (NASA, 2007).

4. What is a product (in terms of aerospace product, such as an aircraft system)?

Jackson S. explains that the aircraft system is not only the aircraft (the flight vehicle/product) itself. The system includes the aircraft, the training equipment, the support equipment, facilities and personnel. In addition, the aircraft can be broken down into its elements /segments: environmental, avionics, electrical, interiors, mechanical, propulsion, auxiliary and airframe. For example, the avionic segment includes the communications, navigation, indicating and recording, and auto flight equipment. This is an important segment that may be called the aircraft management segment, because it includes the communications subsystem and the aircraft monitoring functions that are found in the indicating and recording subsystem (Jackson, 2015).

5. What is enabled by EBD?

EBD includes three main activities: environment analysis, conflict identification, and solution generation. These activities work together in order to generate the design specifications and solutions. The environment analysis helps to determine the key environment components, in which the product works, and the relationships between those components. From the environment implied in the design problem, a designer, by using EBD, could introduce additional environment components that are relevant to the design problem at hand (Tan et al., 2012).

"The EBD provides designers a sense of direction by guiding them to collect the necessary and sufficient information for a design task, by supporting them to determine the focus at each stage of design, by helping them decompose a complex problem into atomic ones, and by investigating potential solutions for each atomic design problem. This could help designers manage their mental stress in solving a design problem, which increases their chances to take advantage of their creative potentials " (Zeng, 2015).

6. What is analysis? What is scenario analysis?

Analysis is an evaluation based on decomposition into simple elements (SAE Aerospace, 2010).

Scenario analysis is a tool for envisioning alternate futures. This can allow formulation of a strategy that is helpful in responding to future opportunities and challenges (Kahn, 2010).

7. What is development? What is development in aerospace?

Development controls the design process and provides baselines that coordinate design efforts. Development in aerospace includes different stages: concept level, system level, and subsystem/component level. It produces a set of corresponding detailed descriptions of the products' characteristics, essential for their production (Department of Defense, 2001).

8. What is a product? What is development of product (in the context of aerospace product)?

Product describes all goods, services, and knowledge that can be sold. It includes attributes (features, functions, benefits, and uses) and can be tangible (e.g. physical goods), intangible (e.g. service) or can be a combination of the two (Kahn, 2010).

Product development of the systems (e.g. aircraft) is a creative and knowledge-intensive process. This process involves information and knowledge exchange and sharing within the different teams and developers (Wu et al., 2014).

9. What do you mean by enabled in the statement "EBD enabled analysis"? How / Why / When / Where EBD enabled analysis?

EBD provides designers with a systematic approach that helps them to reach a deep understanding of a design problem (Tan et al., 2013). This methodology is based on a recursive logic of the design (Zeng, 2011). This enables a design process to recursively iterate between design requirements and solutions until the final solution is found (Zeng & Cheng, 1991).

10. What do you mean by enabled in the statement “enabled scenario analysis”? How/ Why/ When/ Where enabled scenario analysis?

The EBD methodology proposes a method for environment analysis. The objective of environment analysis is to determine the key environment components in which the product works, and the relationships between these components. By following the rules of asking and answering questions, based on this methodology, the designer is able to introduce more environment components that are relevant to the design problem and scenario analysis.

In addition, Wang and Zeng (M. Wang & Zeng, 2009) introduce EBD rules on the question asking process in order to conduct a comprehensive environment analysis. For verifying the completeness of the extracted environment components and their relations they proposed a roadmap as guidance for requirements modeling (Chen & Zeng, 2006). This roadmap relates to different kinds of environments and life cycle of the product or system (Tan et al., 2013).

11. What do you mean by analysis in the statement “scenario analysis for aerospace product development”? How/ Why/ When/ Where is scenario analysis for aerospace product development?

The analysis process of scenarios in aerospace product development is a critical one. This process enables elicitation of missing requirements that had been discovered within the scenario representation. It analyzes, among others, a wide range of possible future events, the risks, and behaviour of the system.

By analyzing information that is found within the answering question step, the main points summarized in the final information section for the general question asking step are as follows:

1. EBD enabled scenario analysis for aerospace product development. 2. The EBD leads that design change environment to a desired one by generating a new artefact. 3. Scenario analysis is a process of analyzing possible future events by considering alternative possible outcomes. 4. Scenario analysis presents several alternative future developments, sets of behaviour or risk assessment events. 5. Setting scenarios is based on the existing knowledge base and new requirements. 6. Requirements analysis develops: operational (functional and performance), technical and structural requirements at system or aircraft levels. 7. Operational requirements come from operational scenarios. 8. Operational scenarios include operational modes; mission phases (e.g., installation, start-up, nominal, off-nominal, stressful conditions, shutdown, and maintenance); critical sequences of activities for users. 9. Product development process involves extensive information, knowledge exchange and sharing among teams and developers. 10. Public laws, formal directives, technical limitations, regulations (service and component), international agreements, cost, time and human resources effect the development process. 11. This process has 3 main levels: concept, system, subsystems. 12. The product of design or development processes can be tangible or intangible (service/experience/belief). 13. Specialists (designers, manufacturers, operators, ground support, and maintenance) produce essential outcomes that effect and progress the development process. 14. In the aerospace field, development process includes the avionic part. 15. This part develops and produces electronic systems for aircraft. 16. Avionic/ electronic systems support communications, navigation, display and management of

multiple systems and subsystems in the aircraft. 17. The development process of the systems is based on requirements analysis.

This section includes 17 numbered sentences and extraction of general information from the first step of the analysis. Basis on this information, a ROM diagram can be generated as shown in Figure 23.

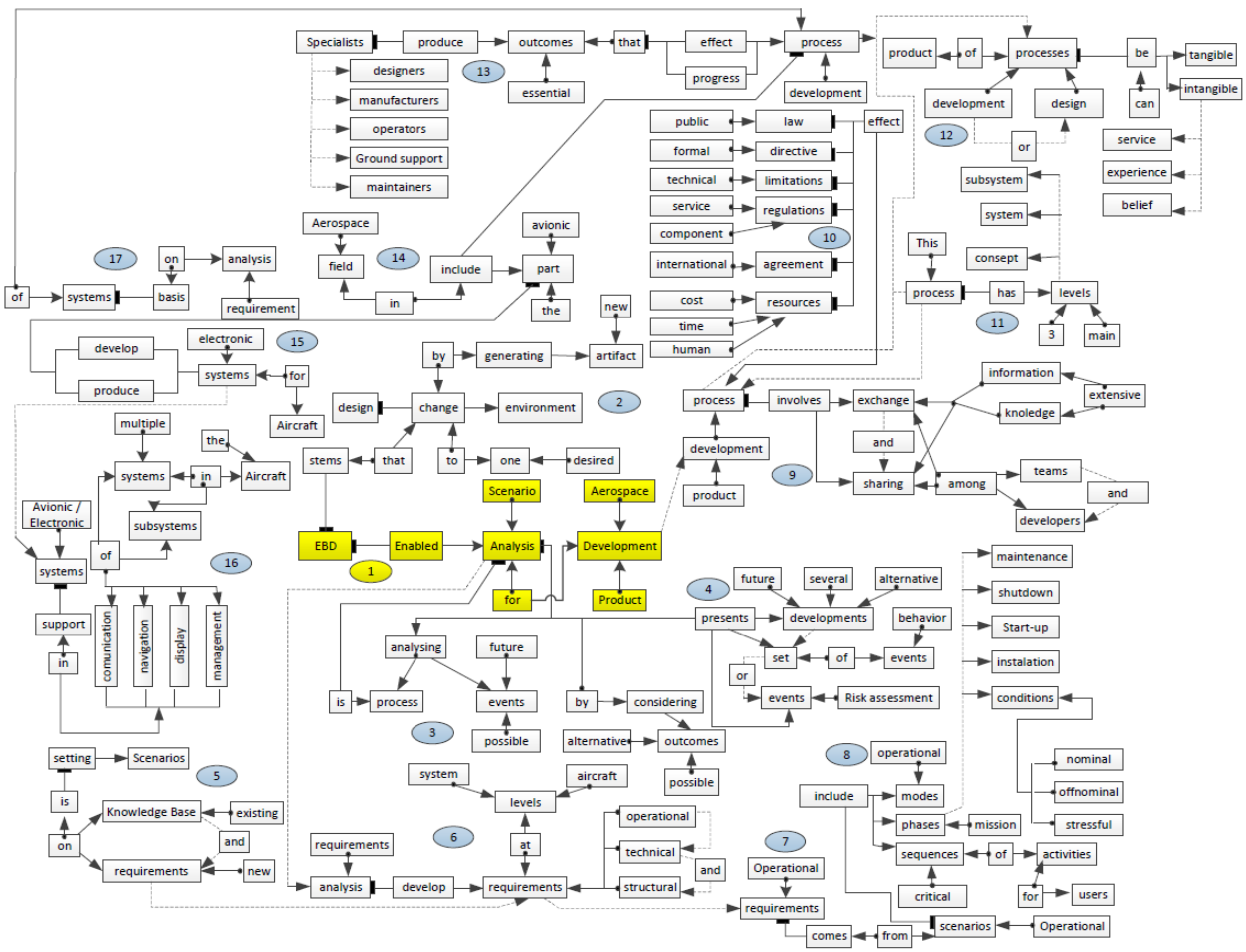

Figure 23: General ROM diagram after the first step of the environment analysis

This ROM diagram represents general knowledge for ADRO with basic and extended information. Information is found within the first step of general question asking by using EBD methodology rules. The next step is domain-specific question asking. This step allows extraction of domain-specific information. Here, the process proceeds with sentences 2, 3, 4, 5, 8 and 9.

These sentences have constrained words that are more specific for the aircraft development process. By analysis of these words and their environment it is possible to elicit more components and relationships as knowledge for ADRO. The other sentences can be analysed in further research as an extension of the knowledge that was found here.

3.1.2 Environment analysis and knowledge representation for ontology

This analysis step focuses on the domain specific environment. Following the EBD methodology rules that are given in Table 5 provides an opportunity to identify domain-specific components and relationships. The following questions and steps will be used to direct this analysis:

Step 1, Question 1: What is the lifecycle of a constrained word in the sentence?
Step 2, Question 2: What are the relevant components for the natural, built and human environments for each event in the lifecycle of the constrained word?
Step 3: Update the ROM diagram with the knowledge components and relationships that were found for the ADRO.

An analysis of the first sentence was given in Section 3.1.1. The analysis for sentence 2 will be performed in Section 3.1.2.1.

3.1.2.1 Sentence 2

Figure 24 represents sentence 2 from the general ROM diagram in order to indicate visually the sentence and its number. To ease the identification of sentences in the ROM diagram, each one was numbered. Sentence 3 will be represented in same way as the second, while the others will be represented alone but in meaning are connected to the general ROM diagram. Each sentence will be updated further with new components and relationships after further steps of the analysis.

Sentence 2 is: "The EBD leads that design change environment to a desired one by generating a new artefact". Here the main constrained words are "**Change**" and "**Artifact**". The artifact is in context of aircraft.

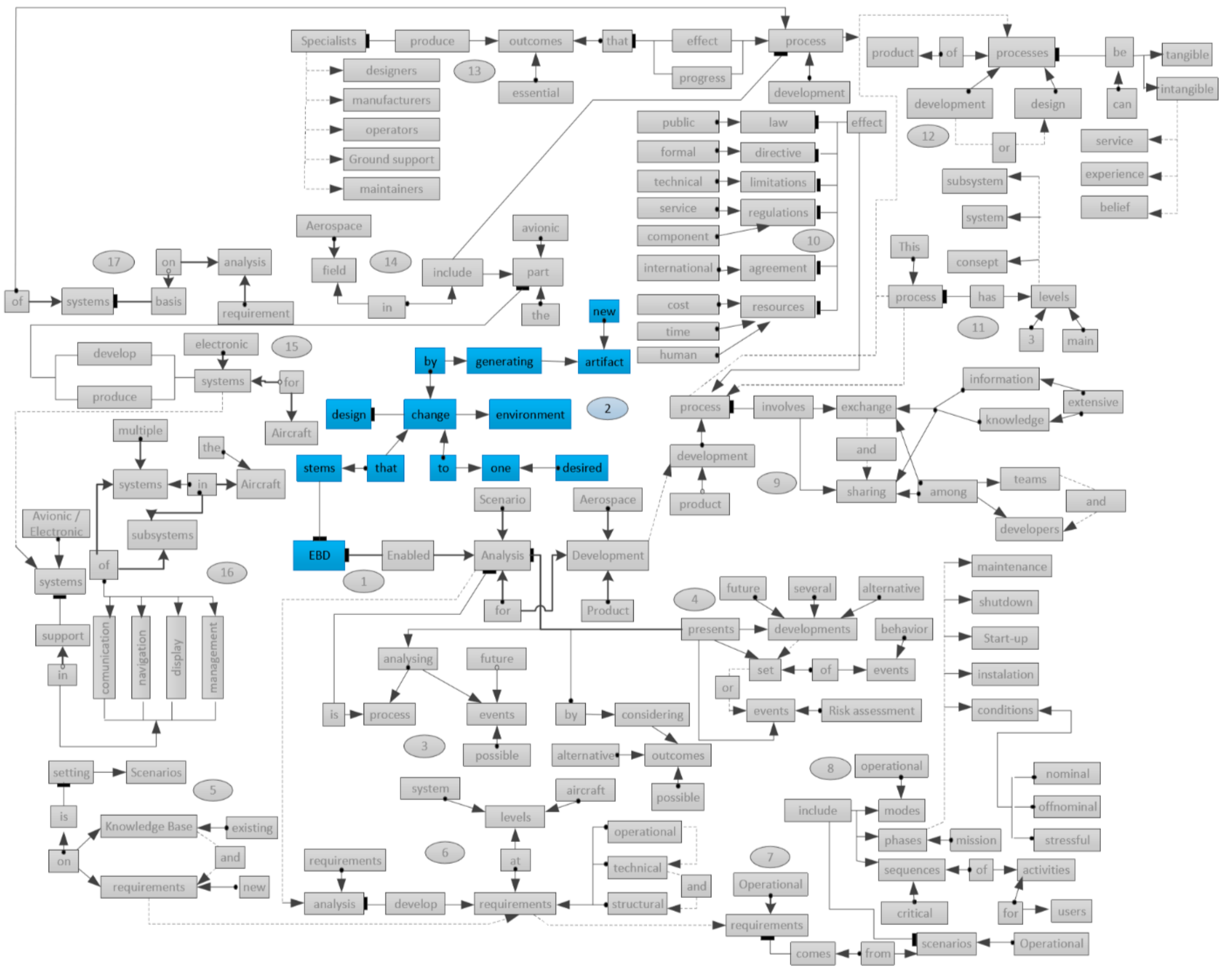

Figure 24: Sentence 2 in the general ROM diagram

Step 1: The lifecycles for the words "**Change**" and "**Artifact**" are defined as follows:

"Change": Before, during, after.

"Artifact" (Aircraft): Design (concept and development), manufacture, sales, transportation, use/utilization, maintenance, recycling/end of life (retirement/disposal).

Step 2: Summary of the relevant components for the natural, built and human environments for each event in the lifecycle of the constrained words.

The tables below represent the main components and relationships to the groups that are found in the natural, built and human environments. These environments are designated in Tables 13 and 14 with their first capital letters in each lifecycle stage.

Main constrained word	Life Cycle stages	Environment components Human (H), Built (B), Natural (N)
Change	Before	**H**: Specialists: Engineers, Technologists, Marketing, Change Analysts, Project representatives. Regulation Bodies: (TCCA (Transport Canada), FAA (Federal Aviation Administration), EASA (European Union Aviation Safety Agency). **B**: Requirements: technical, mandatory, and regulation requirements (including safety), service, ergonomic, market, and exceptions. Resources: budget, time. Equipment: computer, simulators, test, and assembly equipment. Facilities: lab, office. **N**: standards, laws, rules, and expectations.
	During	**H**: Specialists: Engineers, Technologists, Marketing, Change Analysts, Change Management Coordinator, Project representatives. **B**: Documents: specifications, drawings, technical instruction documents, validation and verification plans, tests, and protocols. Communication: software, interaction channels, and interfaces. Facilities: production facilities and infrastructures **N**: standards and laws, certification agencies.
	After	**H**: Specialists: Engineers and Project representatives. Customers. **B**: Documents: Approved SOPs and specifications for mass production, user manuals. Equipment: training and support equipment. Facilities: storage. **N**: Law and certification agencies.

Table 13: The main components and relationships for constraint word "Change"

Main constrained word	Life Cycle Stages	Environment: Human (H), Built (B), Natural (N)
Artifact in the context of **Aircraft**.	**Design (Concept & Development)**	**H**: Customers: aircraft market and specific customers; specialists: designers, engineers, technologists, marketing, coordinator, project representatives.

		Regulation bodies: (TCCA (Transport Canada), FAA (Federal Aviation Administration), EASA (European Union Aviation Safety Agency), and others. **B**: Needs, product requirements: direct operating cost, price, mass properties, dimensions (aircraft, runway, and others), number of seats, route structures, reliability, human factors, environments, maintainability, transportability, flexibility and expansion, permissibility. Mission objectives, development tests, etc. Regulation requirements: CAR (Canadian Aviation Regulations from TCCA), FAR (Federal Aviation Regulations from FAA), CS (Certification Specification from EASA). Standards: ARP 4754A, ARP 4761, AIR 6110, MIL-STD-961D, SAE 4759, AS9100, etc. Technical Requirements: Design, emitted noise level, emitted electromagnetic interference (EMI), safety, architecture (including software and interfaces), basic characteristics, concept (for new, derivative, or change-based aircraft), and engineering sketches. **N**: Technologies (new and existing), SDR (System Design Review), Preliminary Design Review (PDR), Critical Design Review, and others.
	Manufacture	**H**: Project managers, engineers, assemblers, integrators, quality control, and quality assurance representatives, transporters, maintainers, etc. **B**: Facilities: hangar, runway, test tunnel, etc. Equipment: computers, testers, assembly tools, templets, jigs, software (to capture production process, test results, and records), etc. Production Documents: SOW, drawings, specifications, work instruction documents, checklists, reports (test reports, Validation and Verification), etc. Assembly Parts: mechanic, optic, pneumatic, hydraulic, avionic, electronic parts, etc. **N**: Suppliers, subcontractors, manufacturers, customers, distributors, and others.
	Sales	**H**: Customers Specialists: marketing, project representatives Regulation Bodies: (TCCA (Transport Canada), FAA (Federal Aviation Administration), EASA (European Union Aviation Safety Agency), and others. **B**: Cost, spares, marketing strategies, contracts, configuration report, Validation, and Verification reports, test reports, demonstration,

		simulation (FEA, CFD), and production records. System Safety Assessment (SSA), qualification report, certification summary and functional hazard assessment (FHA), user manuals, and training documents and equipment. **N**: Aircraft market, specific customers, or stakeholders.
	Transportation	**H**: Support teams: drivers, pilots, maintenance/technical teams, etc. **B**: Equipment for transportation, limitation requirements: weight, dimensions, shock, vibration, and others. Documents: SOPs, regulation, and customer requirements document for transportation. Facilities and other aircraft. **N**: Weather conditions.
	Use /Utilization	**H**: Customers and Users: airlines, aircraft crew, room control team, airport support teams, passengers/ direct customers. Team: Security and others. **B**: Documents: user manual, SOPs, and check lists. Training: for aircraft crew, airport support teams, simulators. Facilities: terminal, hangars, runways, country, control room, other aircrafts Equipment: satellite navigation system (general). Ground support equipment: Jet way, push and pushbacks tractors, tow tractors, cargo loading equipment. **N**: earth, air, attitude, FOD (Foreign object debris), precipitation, solar radiation, humidity, fungus, volcanic ash, temperature (altitude temperature profile, aerodynamic and induced heating, heat soak (on the ground), pressure (external: atmospheric & aerodynamic, internal: cabin pressure), decompression (pressure and pressure rate), electromagnetic interference (EMI), high-intensity radio fields (HIRF), shock, vibration, load, lightning (spark), sand and dust (during the run-up, taxing or take-off).
	Maintenance	**H**: Specialists: engineers, quality control representatives, maintenance/technical teams. **B**: Facilities: Hangar and others. Equipment: maintenance/technical equipment (ground support equipment). Parts: Spares. Documents: Requirements for maintenance and for equipment, check lists, cost evaluation report: cost per 1,000 flight hours (MN$/1000FH), man-hours (MMH/1000FH), material cost (MT$/1000FH)). Repair process documents: corrosion protection, overnight checks, A checks, B checks, C checks, unscheduled maintenance, and fixed interval checks.

		N: Environment: accessibility environment, corrosive/toxic environment: acids, hydraulic fluids, salt spray, and humidity.
	Recycling / End of Life (Retirement /Disposal):	**H**: Customers and Users: security, customer, supplier, recycling, and maintenance teams. **B**: Documents: BMP (best management practice) manual, Maps for segregation areas, disposal of material procedure, internal audits report, training documents, receiving documents, records, reference manuals, procedures, standards, contracts. Facilities: location for disassembly or recycling, secure and segregation areas. Parts: undesired material, incoming/shipping/packaging materials. Equipment: maintenance/technical equipment (ground support equipment), tooling equipment and/or machinery. **N**: fluids and hazardous materials, inventory, infrastructure, storage, regulations.

Table 14: The main components and relationships for constraint word "Artifact" (In the context of Aircraft)

Here is a list of the sources that helped to find the information and environments components:

(*NASA*, 2007) , (INCOSE, 2006), (Department of Defense, 2001), (SAE Aerospace, 2010), (ISO/IEC/IEEE15288, 2015), (Jackson, 2015), (AFRA, 2018).

After completing a summary of the components, it is possible to continue to Step 3 of updating the ROM diagram with the knowledge components and relationships that were found. Figure 25 below demonstrates the relevant components and their relationships.

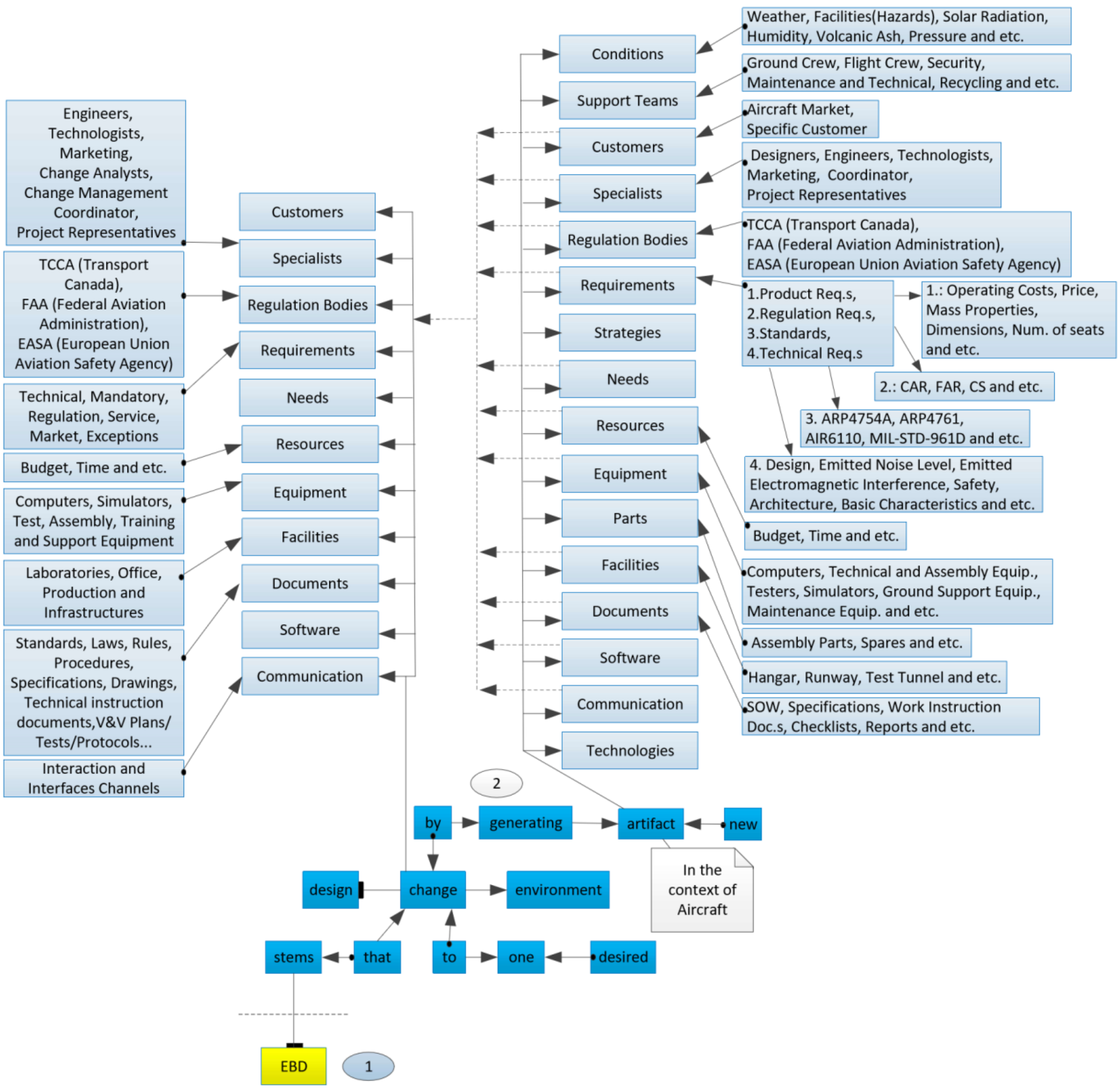

Figure 25: Sentence 2 of the main ROM diagram updated with knowledge from domain-specific environment analysis

3.1.2.2 Sentence 3

The next sentence for the analysis is Sentence 3: "Scenario analysis is a process of analyzing possible future events by considering alternative possible outcomes". This sentence is highlighted from the general ROM diagram in Figure 26.

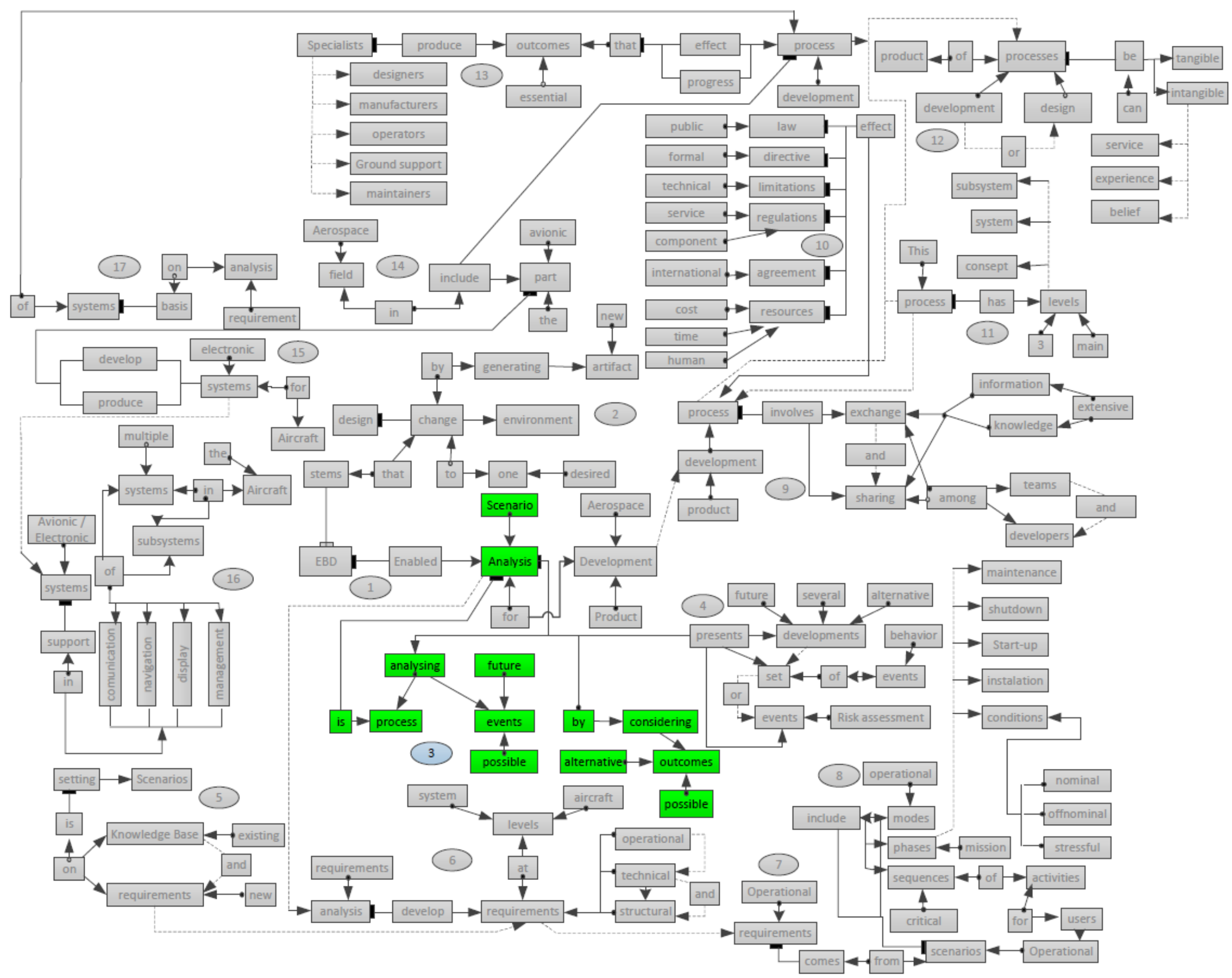

Figure 26: Sentence 3 in the general ROM diagram

Step 1: The main constrained words are: process, events and outcomes. The life cycles for these words defined as follows:

“Process” (In the context of the operational scenario analysis process): before, during, and after.

“Events”: before, during, and after.

“Outcomes” (In the context of requirements): acquisition, verification and validation (V&V), maintenance, utilization.

Step 2: Summary of the relevant components for the natural, built and human environments for each event in the lifecycle of the constrained words.

Tables 15, 16, and 17 represent the main components and relationships to the groups that are found in the natural, built and human environments.

Main constrained word	Life Cycle Stages	Environment: Human (H), Built (B), Natural (N)
Process (In the context of the operational scenario analysis process)	Before	**H**: customers, regulation bodies, specialists: marketing, project representatives, system engineers. **B**: Requirements: regulation bodies, customers, market, standards, needs, Concept (initial concept of the product), equipment (computers, testers, simulators, etc.). Documents: procedures, software (can be for different kind of equipment, systems and subsystems), SOW, etc., Resources: time, budget, etc. **N**: Uncertainty (in functions performance and capabilities), black box model.
	During	**H**: Specialists: marketing, project representatives, system engineers (QA, mechanic, software, etc.), designers and technologists. Customers. **B**: List of requirements, software, trade – off studies, concept, capabilities, functions, initial specifications, events, conditions, and resources: time, budget, software models, etc. **N**: More certainty and understanding (in functions performance and capabilities).
	After	**H**: Specialists: marketing, project representatives, technologists, system engineers (QA, mechanic, software, etc.), customers, regulation bodies. **B**: List of requirements (related to functions, performance and others); Documents: requirements allocation and traceability, validation and verification (V&V) plans, protocols and tests. **N**: Software models for V&V, simulators, achieved new functions and capabilities.

Table 15: The main components and relationships for constrain word "Process"
(In the context of the operational scenario analysis process)

Main constrained word	Life Cycle Stages	Environment: Human (H), Built (B), Natural (N)
Analysis of possible / future **Events**	Before	**H**: customers; regulation bodies; specialists: marketing, project representatives, system engineers (QA, mechanic, software, etc.). **B**: Documents: standards, airspace policy, flight laws & rules, procedures; Design Concept: system, subsystem, component; **N**: Contradictory hazards; Conditions: weather, environment, traffic and safety.
	During	**H**: customers; specialists: marketing, project representatives, system engineers (QA, mechanic, software, etc.), technologists; Actors: operator, crew, passengers. **B**: Trade-off studies; Requirements; Documents: risk assessment, FMEA (Failure Mode and Effects Analysis), specifications, manuals, procedures; Models: mathematical and physical; scenarios; capabilities; concept: architecture (logical and physical), system, subsystem, components, functions, and processes of function exchanges. Software; Equipment: simulators, computers, maintenance tools, etc. **N**: Environment (natural environment of product/system), conditions (pre and post), and interfaces (functional, physical, external, internal, and operational), infrastructures (airport, runway, test tunnel and others), human factors.
	After	**H**: Customers, regulation bodies, specialists: marketing, project and product managers, technologists, engineers (QA, mechanic, software, etc.). **B**: List of requirements (function, performance, regulation, certification, qualification, etc.); documents: requirements allocation and traceability, FMEA, risk mitigation documents, Validation and Verification plans, protocols and tests, specifications, manuals, etc; Software, software models for V&V; Equipment: simulators and others. **N**: ---

Table 16: The main components and relationships for constraint word "Events"

Main constrained word	Life Cycle Stages	Environment: Human (H), Built (B), Natural (N)
Outcomes (In the context of Requirements).	Acquisition	**H**: Customers, Specialists: marketing, project and product managers, technologists, engineers (QA, mechanic, software, and others). **B**: Customer needs, requirements: capabilities ,marketing/ technical, mission, aircraft operational, functional and performance, safety, maintainability, interface; regulation and certification requirements: RTCA DO-254 or DO-178B, ARP

		4761, ARP4754A, MIL-STD-961D, SAE 4759, AS9100, DOD, NASA, INCOSE, AIR 6110, etc. Methods: mathematical models, analytical techniques, calculated or derived data. **N**: Software, software models.
	Verification and Validation (V&V)	**H**: Specialists: engineers (QA, mechanic, software, system and others), project and product managers, technologists; suppliers, manufacturers, regulation bodies: TCCA (Transport Canada Civil Aviation), FAA (Federal Aviation Administration - US), EASA (European Union Aviation Safety Agency), CAST (Commercial Aviation Safety team), etc. **B**: Requirements (marketing/technical, aircraft operational, functional and performance, safety, maintainability, interface, regulation and certification requirements: CAR (Canadian Aviation Regulations) - TCCA, FAR (Federal Aviation Regulations) - FAA, CS (Certification Specification) - EASA). Documents: FMEA, QPL, Validation and Verification plans and program protocols and tests (verification matrix: (requirements, associated functions, development assurance level, verification methods, verification conclusion, verification coverage summary)), certification plan, specifications, requirements allocation and traceability). Methods: mathematical models, analytical techniques, simulations, etc. Inspection (the visual examination): system, component, or subsystem for physical design features or specific manufacturer identification, ground tests, flight test, in – service data or similarity, system safety assessment (SSA), buyer-furnished equipment (BFE), potential risks, technical performance measurements (TPM). Demonstration: system, subsystem, or component operation. **N**: software, software models for V&V, simulators, agencies (regulation, certification and qualification), infrastructures (airport, runway, etc.).
	Maintenance	**H**: Specialists: knowledge owner, knowledge focal, and system engineers. **B**: Documentation Methods: archive, knowledge base, servers, and software. **N**: ---
	Utilization (in new projects)	**H**: Specialists: engineers (QA, mechanic, software, system engineers and others), managers (project and product), technologists, marketing team. **B**: Documentation methods: archive, knowledge base, servers; searching methods, software, data base, model based system. **N**: New or existing programs or projects.

Table 17: The main components and relationships for constraint word "Outcomes" (in the context of Requirements)

Sources:

(EIA, 1999), (RTCA SC- 167 / EUROCAE WG- 12, 1992), (Federal Aviation Administration, 2016), (*NASA*, 2007), (INCOSE, 2006), (Department of Defense, 2001), (SAE Aerospace, 2010), (ISO/IEC/IEEE15288, 2015), (Jackson, 2015), (Crawley et al., 2016), (Raymer, 2012).

The ROM diagram for Sentence 3 updated with this knowledge is represented in Figure 27.

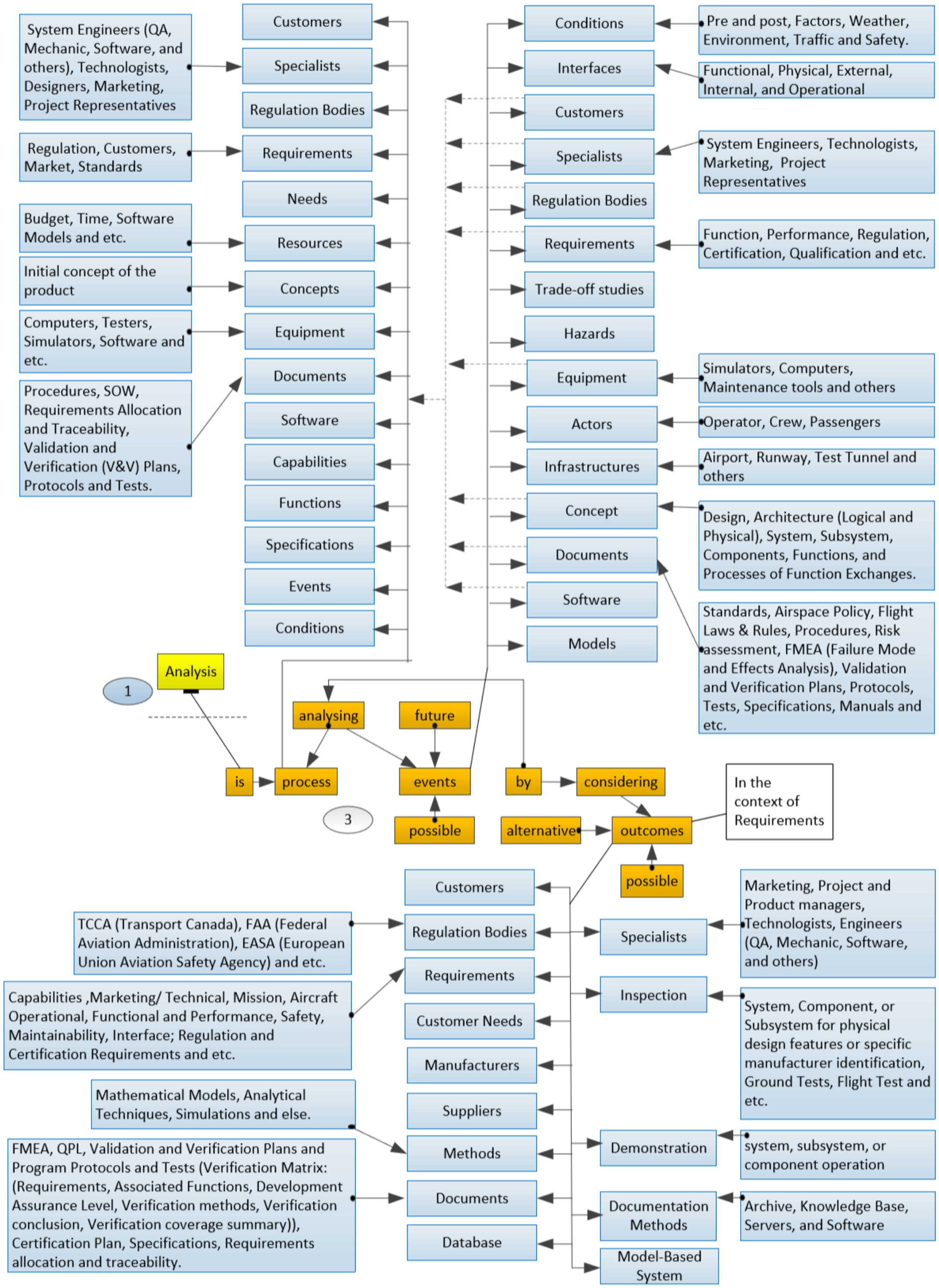

Figure 27: Sentence 3 of the main ROM diagram updated with knowledge from the domain-specific environment analysis

3.1.2.3 Sentence 4

Sentence 4 is: "Scenario analysis presents several alternative future developments, sets of behaviour or risk assessment events". Figure 28 demonstrates this sentence as follows:

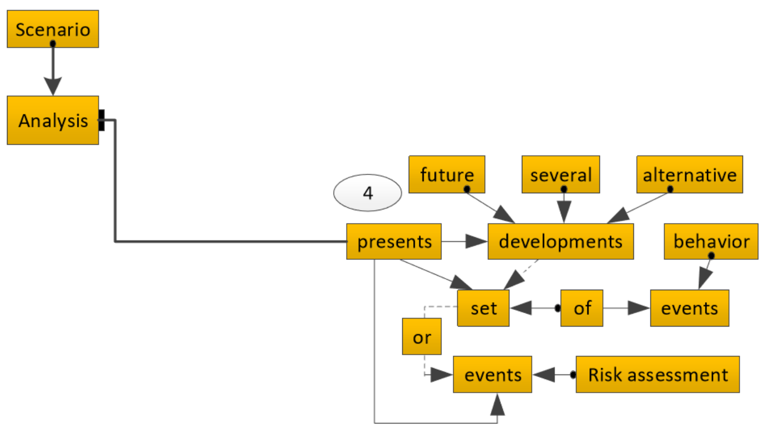

Figure 28: Sentence 4 from the general ROM diagram

Step1: The main constrained word is "**Developments**". Here, the analysis includes two meanings of development: software and technologies. Their lifecycles are defined as follows:

"Developments" (meaning "software"): concept, development, synthesis, validation and verification (V&V), approval, maintenance and modification.

"Developments" (meaning "technologies"): problem definition, knowledge discovering, trade-offs generation, validation and verification (V&V), approval.

An additional constrained word is "events"; the environment analysis for this word was performed in Sentence 3.

Step 2: Summary of the relevant components for the natural, built and human environments for each event in the lifecycle of the constrained words.

Tables 18 and 19 represent the main components and relationships to the groups that are found in the natural, built and human environments.

Main constrained word	Life Cycle Stages	Environment: Human (H), Built (B), Natural (N)
Developments (meaning "**software**")	Concept (through Functional Analysis)	**H**: Specialists: CSE (Chief System Engineer), CDE (Chief Design Engineer), system engineers, software engineers, etc. **B**: Scenarios, Requirements: system, software protection and computer anomalies; Software constraints: software level (A, B, C, D and E), memory and timing consumption, memory protection, program storage and integrity, computer anomalies, fail-safe design, robustness, programming language and compiler; Documents: software standards, software plans: the plan for software aspects of certification, the software development plan, the software verification plan, the software configuration management plan, the software quality assurance plan **N**: Design constraints
	Development (Design and Coding)	**H**: Specialists: system engineers, software developers, commercial aviation safety team- CAST and others. **B**: Design constraints: hardware definition to software, Requirements: software and redundancy requirements, protection requirements, programming language; Documents: software architecture, PSSA – preliminary system safety assessment, control structures, software structure, fault containment boundaries, failure modes; standards: ARP4754A, RTCA/DO-178B, etc. Hardware resources: processors, memory devices, I/O devices, interrupts, and timers. Coupling: Control and data. **N**: ---
	Synthesis (Integration)	**H**: Specialists: system engineers, software developers, etc. **B**: Target hardware/computer; documents: Software architecture, FHA – Functional Hazard Assessment, integrity (correctness of behaviour) of the functions; Codes: source, object, executable object code; Data: Source, linking and loading data. **N**: Computers, system, subsystems and others.
	Verification and Validation (V&V)	**H**: Specialists: system engineers, software developers and SQA - software quality assurance; Regulation Bodies: TCCA (Transport Canada Civil Aviation), FAA (Federal Aviation Administration - US), EASA (European Union Aviation Safety Agency), CAST (Commercial Aviation Safety team), etc.

		B: Documents: software architecture, system and software requirements, test cases, test procedures, test results, coverage, code structure, traceability data, hardware addresses, software planning process, software verification plan, software verification results, overlapping software verification process activities, robustness tests, software configuration management (SCM), software configuration index (SCI), verification tests, software accomplishment survey. **N**: Equipment: target computer and simulator or emulator (test equipment).
	Approval (including Certification)	**H**: Certification authorities: TCCA, FAA, EASA, CAST, JAA and others; customer, supplier, specialists: system engineers, software developers and SQA -software quality assurance; air carrier associations. **B**: Standards: ARP 4754A, AIR 6110, RTCA/DO-178B etc.; Documents: certification, software aspects of certification, plan for software aspects of certification, software accomplishment survey, software quality assurance records, software lifecycle data, software plans and standards, software requirement deviations, software accomplishment summary, software configuration index, software requirements data, design description, source code, executable object code, compiler assumptions, reverification guidelines, software verification results,configuration, etc. **N**: System, computers, simulator or emulator (test equipment).
	Maintenance and Modification	**H**: Specialists: system engineers, software developers and SQA - software quality assurance; air carrier associations; certification authorities; customer. **B**: Documents: configuration management plan (includes: configuration identification, baselines and traceability, problem reports, change control, change review, etc.), SCM (software configuration management) records, software configuration index (SCI), software lifecycle environment configuration index (SECI), software requirements standards, software requirements data, software code standards, design description(definition of the software architecture and the low-level requirements), source code, executable object code, software verification cases and procedures, certification data, etc. **N**: ---

Table 18: The main components and relationships for constraint word "Developments" in meaning of software.

Main constrained word	Life Cycle Stages	Environment: Human (H), Built (B), Natural (N)
Developments (meaning "**technologies**")	Problem definition	**H**: Customers; regulation bodies; specialists: marketing, project representatives, system engineers (QA, mechanic, software, etc.).

		B: Project boundaries: strategy, cost, mission, scenario, budget/ funding, scope and purpose, and resources; Requirements (operational) and organizational policies. **N**: ---
	Knowledge discovering	**H**: Customers; regulation bodies; specialists: marketing, project representatives, system engineers (QA, mechanic, software, etc.), and institutions (who are developing the new technologies). **B**: Research, data, documents: project plan, functional descriptions, mission needs, traceability, design boundaries, technological features, risks, and preliminary design criteria; project financial assumptions: return on investments, benefits, assessment of potential payoffs, etc. **N**: Step-specific tools and equipment.
	Trade-offs generation	**H**: Specialists: marketing, project representatives, system engineers (QA, mechanic, software, etc.), and institutions (who are developing the new technologies). **B**: Concept, capability, prototype, models: (physical, simulation, demonstration, assumption), Documents: specifications, engineering plan, technical event plan, design traceability, analysis, configuration management plan, design plans (e.g., electrical, mechanical, structural, etc.), production / manufacturing plan, quality management plan, risk management plan, technology assessment, development plan (includes: schedules, funding and conclusions), regulatory compliance data, internal compliance data, maintenance and other operational information; Requirements: performance, cost, weight, risk, safety, efficiency; errors: human factors etc.; constraints, conflicts, architectures: functional, and physical; limitations. **N**: Laboratory, simulators and others.
	Verification and Validation (V&V)	**H**: Regulation bodies; specialists: marketing, project representatives, system engineers (QA, mechanic, software, and others), and institutions (who are developing the new technologies). **B**: Project boundaries: mission, mission needs, scenarios; Requirements: cost, weight, risk; test methods: comparison with proven concepts, operational test and evaluation; Documents: lessons learned, procedures, validation plan, validation protocol, and test report. **N**: ---
	Technology Approval (Qualification)	**H**: Regulation bodies; specialists: marketing, project representatives, system engineers (QA, mechanic, software, and others), and institutions (who are developing the new technologies). **B**: Requirements, documents: in-service data and similarity, system safety assessment (SSA), Certification plan, System Integration Plan, etc.; Test methods: simulations, evaluation, demonstration: operational and flight test, ground test , etc; **N**: Training opportunities.

Table 19: The main components and relationships for constraint word "Developments" in meaning of technologies.

Sources:

(EIA, 1999) (Deutsch & Pew, 2005), (RTCA SC- 167 / EUROCAE WG- 12, 1992), (NASA, 2007) , (INCOSE, 2006), (Department of Defense, 2001), (SAE Aerospace, 2010), (ISO/IEC/IEEE15288, 2015), (Jackson, 2015).

Step 3 is represented by the updated ROM diagram in Figure 29 with the knowledge components and relationships that were determined.

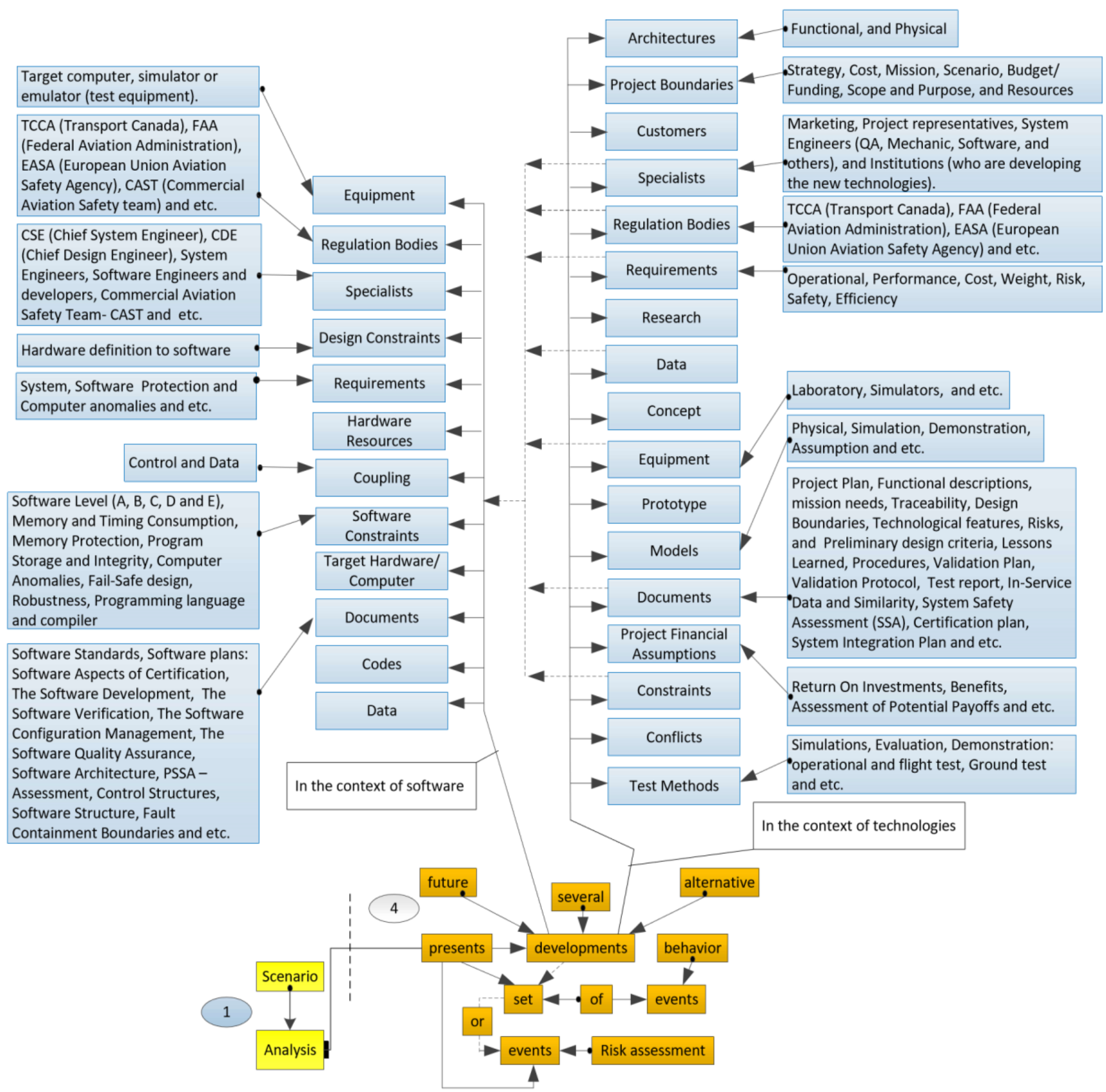

Figure 29: Sentence 4 of the main ROM diagram updated with knowledge from the domain-specific environment analysis

3.1.2.4 Sentence 5

Sentence 5 is: “Setting scenarios is based on the existing knowledge base and new requirements”. Figure 30 demonstrates this sentence in the form as it appears in the general ROM diagram.

Step1: The main constrained words are “**Knowledge Base**” and “**Requirements**”. The analysis for the word “**Requirements**” was performed under the analysis of Sentence 3. Therefore, “**Knowledge Base**” is the analysis to be carried over.

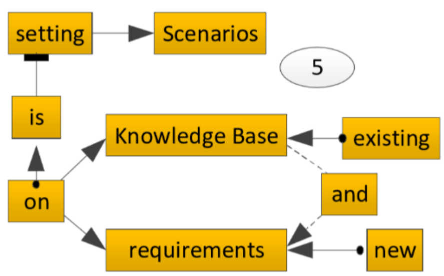

Figure 30: Sentence 5 from the general ROM diagram

Step 2: Summary of the relevant components for natural, built and human environments for each event in the lifecycle of the constrained words.

Table 20 represents the main components and relationships to the groups that are found in the natural, built and human environments.

Main constrained word	Life Cycle Stages	Environment: Human (H), Built (B), Natural (N)
Knowledge Base	Definition	**H**: <u>Customer</u>, <u>user</u>, <u>specialists</u>: software system engineers, knowledge owner, knowledge focal, system engineers, software expert engineers, software developers.

		B: Problem description, requirements (e.g. user, objects, activities, and constraints), human expertise, domain's semantic structure, sequences (e.g. functional), specifications (e.g. user requirement, functions, software system, tools, parameters, mathematical equations, truth tables, rules, empiric relations, sequence of actions, software engineering tasks, numerical computation, graphics, general categories, software system specifications), entities, documents: procedures, information flow structure, structure and behaviour of the desired software application; artificial intelligence, databases, expert systems, inference engine , man-machine interface, queries(e.g. sophisticated deductive query), rules, strategies (e.g. optimization tools), expert systems, spatial objects, concepts, relationships, functions, activities (e.g. queries), constraints (e.g. restrictions on the set of possibilities), entity, models, knowledge, archive, servers, storage, software. **N**: Computer systems, networks, data.
	Creation	**H**: Specialists: software system engineers, software expert engineers, and software developers. **B**: Requirements (e.g. user, objects, activities, and constraints), Domain's Semantic Structure, Sequences (e.g. functional), Specifications (e.g. user requirement, functions, software system, tools, parameters, mathematical equations, truth tables, rules, empiric relations, sequence of actions, software engineering tasks, numerical computation, graphics, general categories, analysis, software system specifications), entities, documents: procedures, software engineering tasks, information flow structure, structure and behaviour of the desired software application, lessons learned; artificial intelligence, databases, expert systems, inference engine, man-machine interface, queries (e.g. sophisticated deductive query), rules, strategies (e.g. optimization tools), expert systems, spatial objects, concepts, relationships, functions, activities (e.g. queries), constraints (e.g. restrictions on the set of possibilities), entity, models, notations and guidelines, interactions, knowledge, archive, servers, storage, software. **N**: Computer systems, networks, data.
	Maintenance	**H**: User, specialists: maintenance programmers, software system engineers, knowledge owner, knowledge focal, system engineers, software expert engineers, software developers. **B**: Knowledge, documents and specifications, structure and behaviour of the desired software application, artificial intelligence, databases , databases-knowledge-directed technologies, inference engine, man-machine interface, queries, rules, expert system shells, expert systems, graphics, general categories. **N**: Computer systems, networks, data, servers, software.
	Searching and Leveraging	**H**: User, specialists: software system engineers, knowledge owner, knowledge focal, system engineers, software expert engineers, software developers. **B**: Knowledge, documents and specifications, software, search engines, rules, artificial intelligence, databases, databases-knowledge-directed technologies, inference engine, man-machine

		interface, queries, rules, expert systems, graphics, general categories. **N**: Computer systems, networks, data, servers, software.

Table 20: The main components and relationships for the constraint word "Knowledge Base"

Sources:

(Jarke et al., 1989), (Zeroual & Robillard, 1992), (Institute of Electrical and Electronics Engineers, 1998), (Burgun et al., 2001), (Anjum et al., 2013), (NASA, 2007) ,(Noy & McGuinness, 2001) (INCOSE, 2006), (Department of Defense, 2001).

Step 3: Representation of updated ROM diagram in Figure 31 with the knowledge components and relationships that were found within the analysis of Sentence 5.

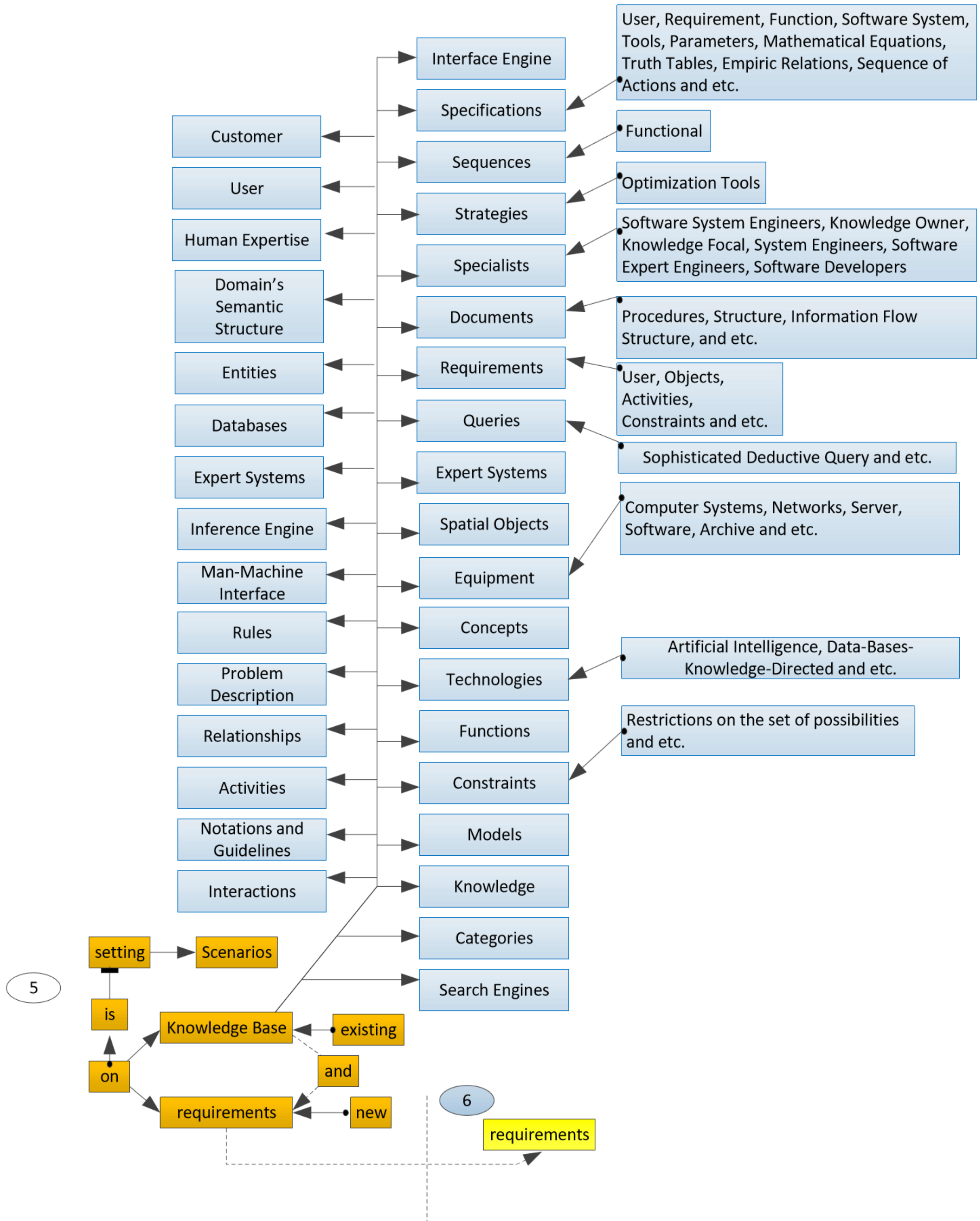

Figure 31: Sentence 5 of the main ROM diagram updated with knowledge from domain-specific environment analysis

3.1.2.5 Sentence 8

Sentence 8 states: “Operational scenarios include operational modes; mission phases (e.g., installation, start-up, nominal, off-nominal, stressful conditions, shutdown, and maintenance); critical sequences of activities for users”. Figure 32 demonstrates this sentence in the form as it appears in the general ROM diagram.

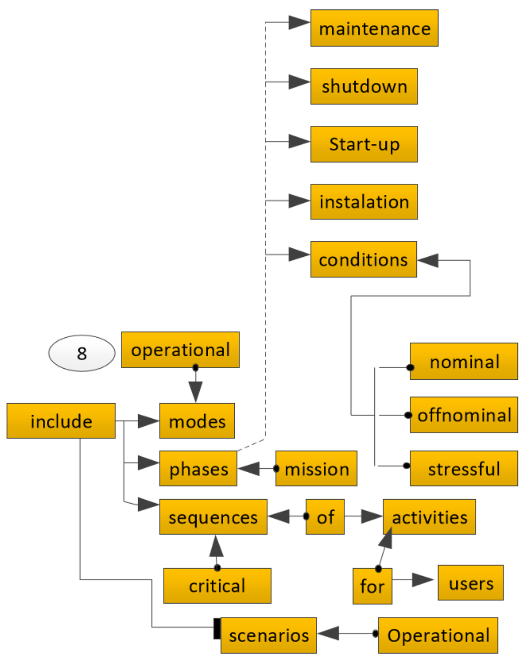

Figure 32: Sentence 8 from the general ROM diagram

Step1: The lifecycle for the main constrained words “**Modes**”, “**Phases**”, and “**Sequences**” defined as follows:

“Modes”: planning action, preparation to action, action, and end of action.

“Phases”: pre-mission, mission, after mission.

"Sequences": analysis, triggers (for the sequences, hierarchy and order), creation of sequences, documentation of sequences, training, validation and verification (V&V).

Step 2: Summary of the relevant components for the natural, built and human environments for each event in the lifecycle of the constrained words.

The tables below represent the main components and relationships to the groups that are found in the natural, built and human environments. These environments are designated in Tables 21, 22 and 23 with their initial capital letters in each lifecycle stage.

Main constrained word	Life Cycle Stages	Environment: Human (H), Built (B), Natural (N)
Modes (in the context of operational modes of Aircraft)	Planning Action	**H**: Actors: aircraft crew, air traffic controller **B**: Mission, methods, documents: plans, protocols, checklists, procedures; sequences (of functions), data: technical data, anomalies, variances and others; software, strategy, conditions: normal, emergency, weather, surface and others; capabilities, resources: time, technical limitations; scenarios. **N**: Place: ground, air, and water; factors: weather, mechanic and human.
	Preparation to Action	**H**: Actors: aircraft crew, air traffic controller **B**: Methods, documents: plans, protocols, checklists, procedures; sequences (of functions), data: technical data, anomalies, variances and others; software, strategy, conditions: normal, emergency, weather, surface and others; capabilities, resources: time, technical limitations; facilities, systems: navigation system, transponder, radio, braking system, etc. **N**: Place: ground, air, and water; factors: weather, mechanic and human.
	Action	**H**: Actors: aircraft crew, air traffic controller **B**: Methods, documents: plans, protocols, checklists, procedures; sequences (of functions), data: technical data, anomalies, variances, etc.; software, strategy, conditions: normal, emergency, weather, surface and others; capabilities, resources: time, technical limitations, etc.; facilities, systems: navigation system, transponder, radio, braking system and others; subsystems, and components. **N**: Place: ground, air, and water; factors: weather, mechanic and human.
	End of Action	**H**: Actors: aircraft crew, air traffic controller, and ground support team. **B**: Methods, documents: plans, protocols, checklists, procedures; sequences (of functions), data: technical data, anomalies, variances and

		others; software, strategy, conditions: normal, emergency, weather, surface and others; capabilities, resources: time, technical limitations and others; facilities, systems: navigation system, transponder, radio, braking system, etc.; subsystems, and components. **N**: Place: ground, air, and water; factors: weather, mechanic and human.

Table 21: The main components and relationships for constraint word "Modes"

Main constrained word	Life Cycle Stages	Environment: Human (H), Built (B), Natural (N)
Phases (in the context of mission phases)	Pre – mission	**H**: Actors: aircraft crew, air traffic controller, and ground support team. **B**: Mission requirements, methods, documents: plans, protocols, checklists, procedures and manuals; sequences (of functions), data: technical data, anomalies, variances, etc.; variables: flight time, fuel amount, weight, load (e.g. cargo, passengers, crew, baggage), balance, altitude, pressure, AOA - angle of attack, temperature, sea level, turbulence index and others; software, strategy, conditions: normal, emergency, weather, surface and others; capabilities, resources: time, technical limitations; facilities, systems: navigation system, transponder, radio, braking system and others; constraints: short runways, obstacles, etc. **N**: Weather conditions: wind, rain, icing, thunderstorms and others; natural horizon, aircraft traffic, terrain influences, hazards, cross wind component, daylight /night conditions.
	Mission	**H**: Actors: aircraft crew, air traffic controller, and ground support team. **B**: Capabilities (e.g. pilot, aircraft), aircraft parts forms: wing, tail, fuselage, and empennage; mission requirements, methods, documents: plans, protocols, checklists, procedures and manuals; sequences (of functions), data: technical data, anomalies, variances and others; variables: speed, flight time, fuel amount, weight, load (e.g. cargo, passengers, crew, baggage), balance, altitude, pressure, AOA - angle of attack, temperature, sea level, turbulence index, etc.; software, strategy, conditions: normal, emergency, weather, surface, etc.; resources: time, technical limitations; facilities, systems: navigation system, transponder, radio, flight instruments, flight control, landing gear, spoilers, flaps, speed brake, engines, thrust and thrust reverser, auxiliary power unit, fuel, hydraulic, pneumatic, electrical, oxygen, environmental control, anti-icing, navigation, fire control; constraints: short runways, obstacles, etc. **N**: Weather conditions: wind, rain, icing, thunderstorms and others; natural horizon, aircraft traffic, terrain influences, hazards, cross wind component, daylight /night conditions.
	After mission	**H**: Actors: aircraft crew, air traffic controller, and ground support team.

		B: Aircraft parts forms: wing, tail, fuselage, and empennage; mission requirements, documents: checklists, procedures and manuals; data: maintenance costs, technical data, anomalies, variances and others; software, facilities. **N**: Weather conditions: wind, rain, icing, thunderstorms and others; natural horizon, aircraft traffic, terrain influences, hazards, cross wind component, daylight /night conditions.

Table 22: The main components and relationships for constraint word "Phases"

Main constrained word	Life Cycle Stages	Environment: Human (H), Built (B), Natural (N)
Sequences	Analysis	**H**: Specialists: system engineers (QA, mechanic, software, etc.), system architect, quality assurance representatives. **B**: Events/scenarios, missions, requirements, modes, structure, data: process parameters (e.g. time, dependency rules, resources, semantics, interactions, language); boundaries (system, operations), system (the form), system components (objects), level of abstraction, functions, interfaces (functional, physical and data interfaces between system elements), capabilities, risks, conditions: weather, environment, etc.; tools/models: functional flow diagram (FFD), event sequence diagrams, event trees; criteria (e.g. task criticality, environmental constraints, workload, feasibility), human and machine adaptability. **N**: ---
	Triggers (For the sequences, hierarchy and order)	**H**: Customers; actors: aircraft crew, air traffic controller, and ground support team and others, regulation bodies; specialists: marketing, project representatives, system engineers (QA, mechanic, software, etc.), quality assurance representatives, and system architect. **B**: Events/scenarios, requirements: standards and authority; FMEA (e.g. risk mitigation and others); functional interactions, capabilities, functions, interfaces, system emergence, structure (of the operational functions), etc. **N**: ---
	Creation of sequences	**H**: Specialists: marketing, project representatives, human engineering engineers, system engineers (QA, mechanic, software, and others), quality assurance representatives, and system architect. **B**: Documents: SOPs, procedures, instructions, checklists, task descriptions, and others; interfaces (e.g. human-machine), instructional drawings that include ideas, thoughts, arguments, and conditions; characteristics or attributes that describes the system and its components. **N**: Network, computers, software, simulators, etc.

	Documentation of sequences	**H**: Specialists: human engineering engineers, system engineers (QA, mechanic, software, etc.), quality assurance representatives, and system architect. **B**: Documents: SOPs, procedures, instructions, checklists, task descriptions, manuals, etc.; instructional drawings. **N**: Project/program environment, software, servers, knowledge bases, archives, etc.
	Training	**H**: Users: aircraft and ground crew, passengers; instructors: flight and training; specialists: quality representative team, system engineers. **B**: Documents: SOPs, procedures (e.g. operational, etc.), checklists, training manuals, training records; equipment: simulators and computers; software, training events (theoretical and practical), skills, requirements. **N**: Network, facilities.
	Validation and Verification (V&V)	**H**: Customers; actors: aircraft crew, air traffic controller, and ground support team and others, regulation bodies; specialists: marketing, project representatives, system engineers (QA, mechanic, software, and others), quality assurance representatives, and system architect. **B**: Documents: SOPs, procedures, checklists, training documents, evaluation tests and exams (theoretical and practical), V&V: plans, protocols, test reports and records; requirements; equipment: simulators and computers; **N**: Network, facilities.

Table 23: The main components and relationships for constraint word "Sequences"

Sources:

(Federal Aviation Administration, 2016), (NASA, 2007) , (INCOSE, 2006), (Department of Defense, 2001), (SAE Aerospace, 2010), (Jackson, 2015), (Dufresne, 2008), (Gutierrez, 2018), (Crawley et al., 2016), (Swaminathan & Smidts, 1999), (Deutsch & Pew, 2005), (Harris et al., 2015), (Micskei & Waeselynck, 2011), (Piperni et al., 2013), (Piperni et al., 2007), (SAE Aerospace, 2011).

Figure 33 represents the updated ROM diagram Step 3, with the knowledge components and relationships that found in the analysis of Sentence 8.

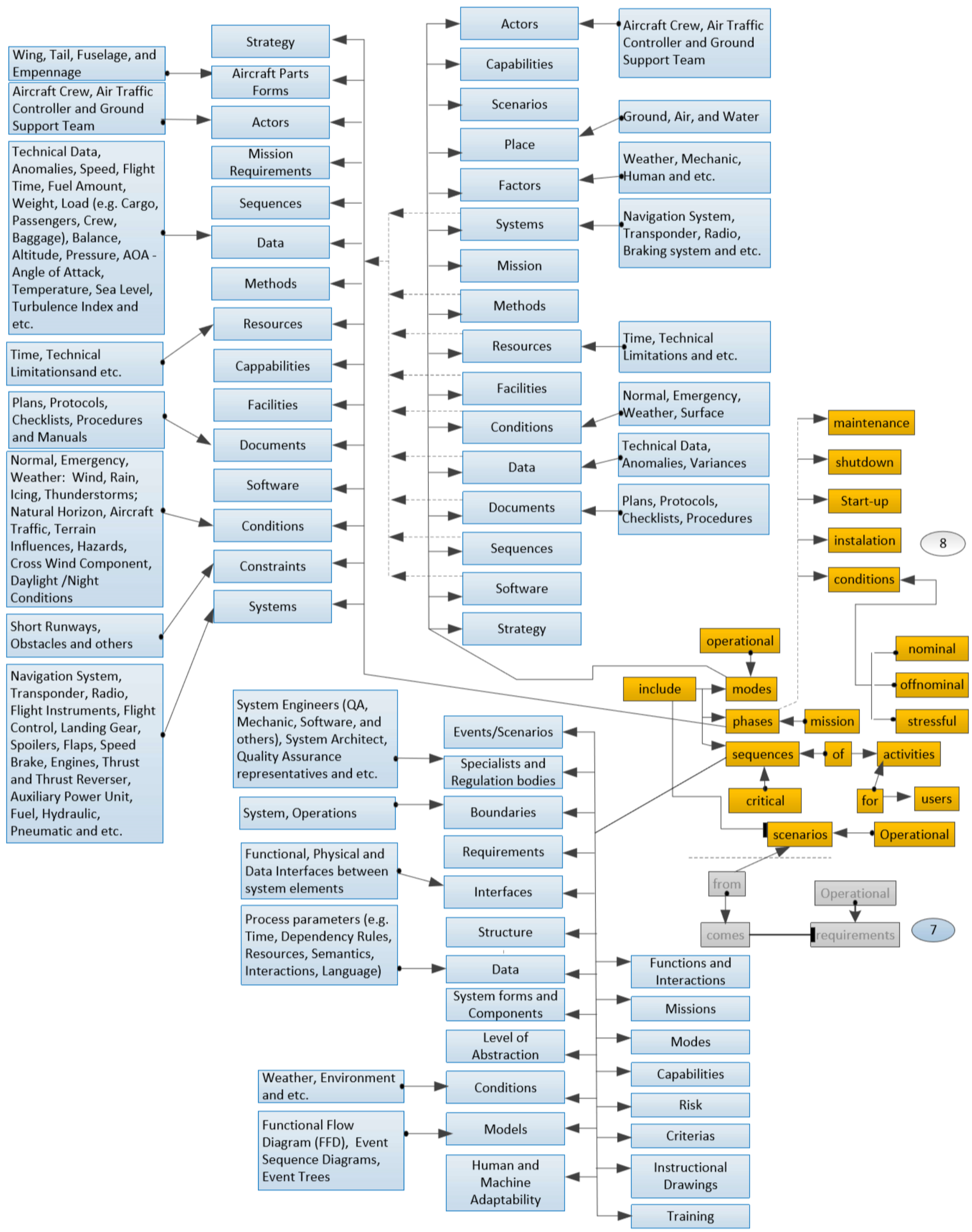

Figure 33: Sentence 8 of the main ROM diagram updated with knowledge from domain-specific environment analysis

3.1.2.6 Sentence 9

Sentence 9 states: "The product development process involves extensive information, knowledge exchange and sharing among teams and developers". Figure 34 demonstrates this sentence in the form as it appears in the general ROM diagram.

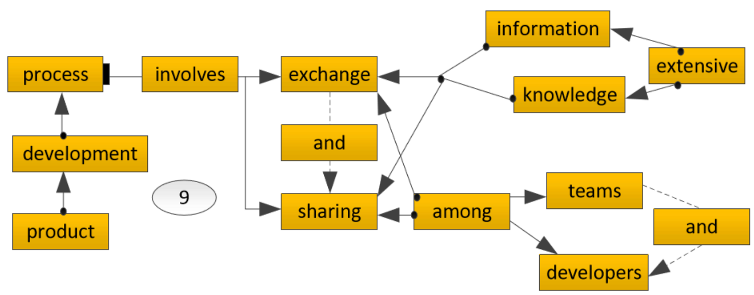

Figure 34: Sentence 9 from the general ROM diagram

Step 1: The life cycle for the main constrained word "**Exchange and Sharing**" is defined as follows:

"Exchange and Sharing" (In the context of knowledge or information): record, maintenance, searching, analysis, and transfer.

Step 2: Summary of the relevant components for natural, built and human environments for each event in the lifecycle of the constrained words.

The table below represents the main components and relationships to the groups that are found in the natural, built and human environments. These environments are designated in Table 24 with their first capital letters in each lifecycle stage.

Main constrained word	Life Cycle Stages	Environment: Human (H), Built (B), Natural (N)
Exchange and Sharing (knowledge or information)	Record	**H**: Users, specialists: marketing, project representatives, system engineers (QA, mechanic, software, etc.), knowledge owner, and knowledge focal. **B**: Data (e.g. parameters, values, etc.), knowledge base architecture, knowledge (e.g. statements, problem description), language format, requirements (e.g. user, objects, activities, and constraints), specifications, rules (e.g. mathematical equations, numerical computation), sequence of actions, graphics, entities, documents(e.g. procedures, strategies), databases, man-machine interface, concepts, views, relationships, constraints (e.g. restrictions on the set of possibilities), models, archive, servers, storage, software. **N**: network.
	Maintenance	**H**: Specialists: knowledge owner, knowledge focal, software expert engineers, software developers. **B**: Functional interoperability, semantic interoperability, data (e.g. parameters, values, etc.), knowledge base architecture, knowledge base, databases, language format, specifications, rules (e.g. mathematical equations, numerical computation), sequence of actions, documents (e.g. procedures, strategies, agent communication protocol, specification of the content of shared knowledge), man-machine interface, concepts, views, relationships, constraints (e.g. restrictions on the set of possibilities), models, archive, servers, storage, software. **N**: network.
	Searching	**H**: Users, specialists: marketing, project representatives, system engineers (QA, mechanic, software, etc.), knowledge owner, knowledge focal, software expert engineers, and software developers. **B**: Activities (e.g. queries), consistent and coherent view, functional interoperability, semantic interoperability, agreement (on the meaning of the terms), ontologies, knowledge-based systems, data (e.g. parameters, values, etc.), knowledge base architecture, knowledge base, databases, language format, specifications, rules (e.g. mathematical equations, numerical computation), sequence of actions, documents (e.g. procedures, strategies, agent communication protocol, specification of the content of shared knowledge), man-machine interface, concepts, views, relationships, constraints (e.g. restrictions on the set of possibilities), models, archive, servers, storage, software. **N**: Network.
	Analysis	**H**: Specialists: knowledge owner, knowledge focal, software expert engineers, and software developers. **B**: Consistent and coherent view, functional interoperability, semantic interoperability, agreement (on the meaning of the terms), semantic and syntactic differences, language format, rules (e.g. mathematical equations, numerical computation), sequence

		of actions, relationships, constrains (e.g. restrictions on the set of possibilities), models, etc. **N**: Network.
	Transfer	**H**: Users (e.g. workgroups, teams), specialists: marketing, project representatives, system engineers (QA, mechanic, software, etc.), knowledge owner, knowledge focal, and software expert engineers. **B**: Functional interoperability, semantic interoperability, agreement (on the meaning of the terms), ontologies, architecture of ontologies, knowledge-based systems, data (e.g. parameters, values, etc.), databases, language format, specifications, rules (e.g. mathematical equations, numerical computation), sequence of actions, documents (e.g. procedures, strategies, agent communication protocol, specification of the content of shared knowledge), man-machine Interface, concepts, constraints (e.g. restrictions on the set of possibilities), models, archive, servers, storage, software. **N**: Network.

Table 24: The main components and relationships for the constraint word "Exchange and Sharing" in the meaning of knowledge or information

Sources:

(Wu et al., 2014) (NASA, 2007) , (Noy & McGuinness, 2001) (Department of Defense, 2001), (Jackson, 2015), (Anjum et al., 2013), (Burgun et al., 2001), (Jarke et al., 1989).

Figure 35 represents Step 3 in the updated ROM diagram with the knowledge components and relationships that were found.

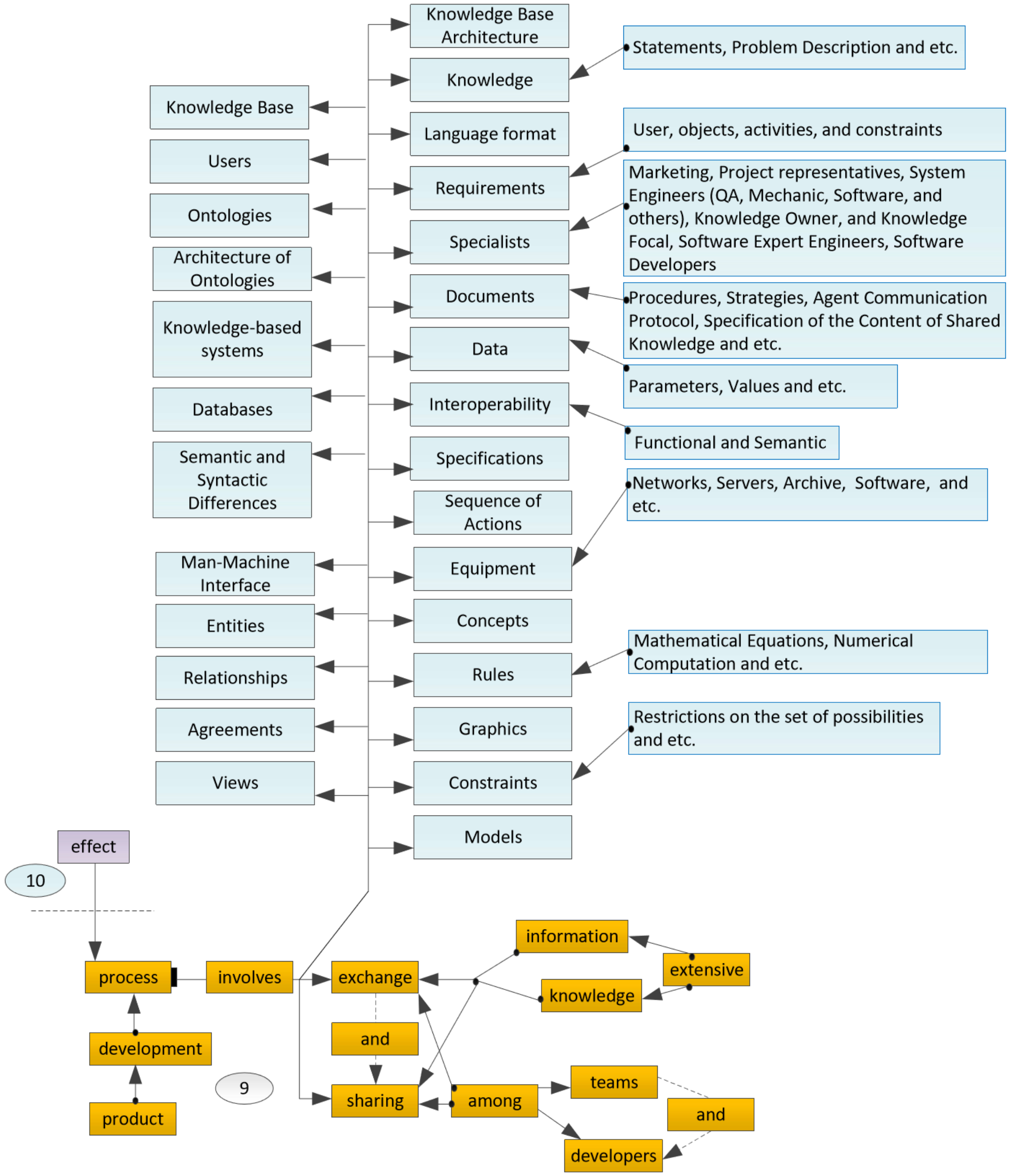

Figure 35: Sentence 9 of the main ROM diagram updated with knowledge from the domain-specific environment analysis

3.2 Summary of the knowledge development part for ADRO

By using the Environment-Based Design (EBD) methodology, a wide range of knowledge for ADRO was provided with less effort than conventional methods. This knowledge includes taxonomies, classes, components and relationships (represented in Sections 3.1.2, and 3.1.2.1 to 3.1.2.6), and is useful in the creation of ADRO and structured organizational knowledge (e.g. requirements and others). In addition, this knowledge can help in transitioning from a document-centric to a model-centric knowledge base that organizations are generally striving for to improve their performance. This transition is essential and has the advantage of using one core information within the organization. This method also provide a uniform language or key words and relationships for utilization in the knowledge searching process. It can also provide an opportunity to obtain a wide spectrum of specific knowledge (e.g. requirements) for implementation in new projects. The EBD methodology has proved to be an effective approach that can be utilized by less experienced persons and can provide essential results in a short period of time.

3.3 Top level requirement analysis to enable knowledge utilization

One of the main general capabilities of an aircraft system is landing. Landing is considered to be a high work load for the pilot phase of flight that requires the simultaneous existence of communication, integration and other activities or specific sequences in time and function. For this reason, the general top level requirement for a system such as an aircraft is further decomposed to a list of requirements for conceptual design, which is the basis for the next steps in system design.

For ease of discussion, Sections 3.3.1 to 3.3.3 describe an analysis of the top level requirement (e.g. example of the procedure-based EBD methodology) to characterize a list of key components as well as their relationships that can be used as tagging components within a searching process of existing relevant decomposed requirements. Section 3.3.4 presents the analysis results versus the model approach, which is based on the so-called Arcadia method (Bonnet, 2015).

3.3.1 Top level requirement analysis by using the EBD methodology

Figure 36 presents graphically the top level aircraft requirement using a ROM diagram an EBD linguistic tool. The requirement is “The Aircraft shall have the braking capability to stop the aircraft during all foreseeable landing conditions within the aircraft defined landing distance and runways”. Based on the ROM diagram, the environment analysis can be initialized by asking generic questions. The ROM diagram emphasizes the main constraining components, which are aircraft, capability, landing, conditions, stop, distance, and runway. The next step is to create a definition of object list for the questioning. Table 25 represents the relations between the components of the ROM diagram following the rules represented in Figure 21.

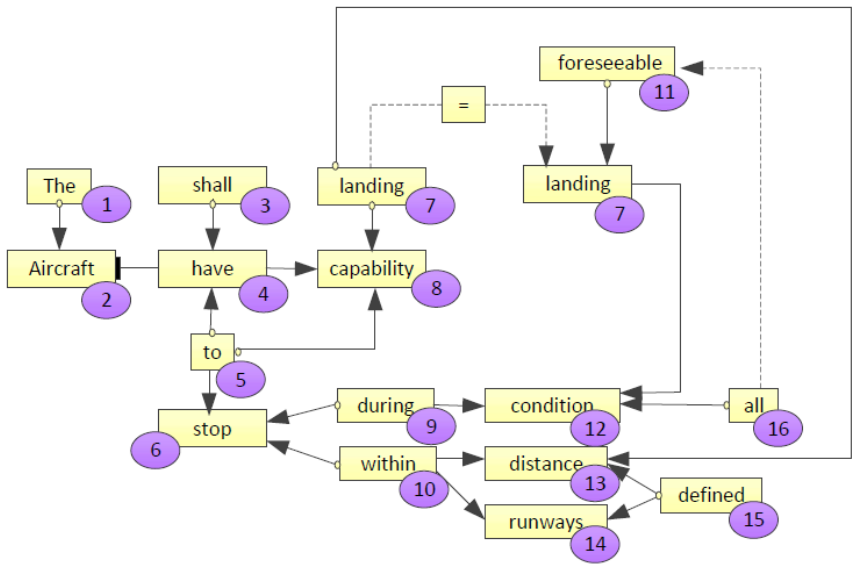

Figure 36: Representation of general top requirement by utilization of ROM diagram

Object	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1		3														
2				1												
3				3												
4		1						2								
5				3				3								
6																
7								3				3	3			
8				2												
9						3						2				
10						3							2	2		
11							3									
12									2							
13										2						
14										2						
15													3	3		
16												3				

Table 25: Representation of the relations between the components of the ROM diagram

Table 26 summarizes the number of relations between the components of the ROM diagram. It can be seen in the table that the central object is 4 since it has higher number of constraints and predicate relations. Figure 37 illustrates the structure of the objects around central object 4 in terms of constraint and predicate relations derived from the ROM diagram.

Number of relations	4	3	2	1
Object	4	8, 12, 13	2, 6, 10, 14	7, 9

Table 26: Number of constraint and predicate relations on an object

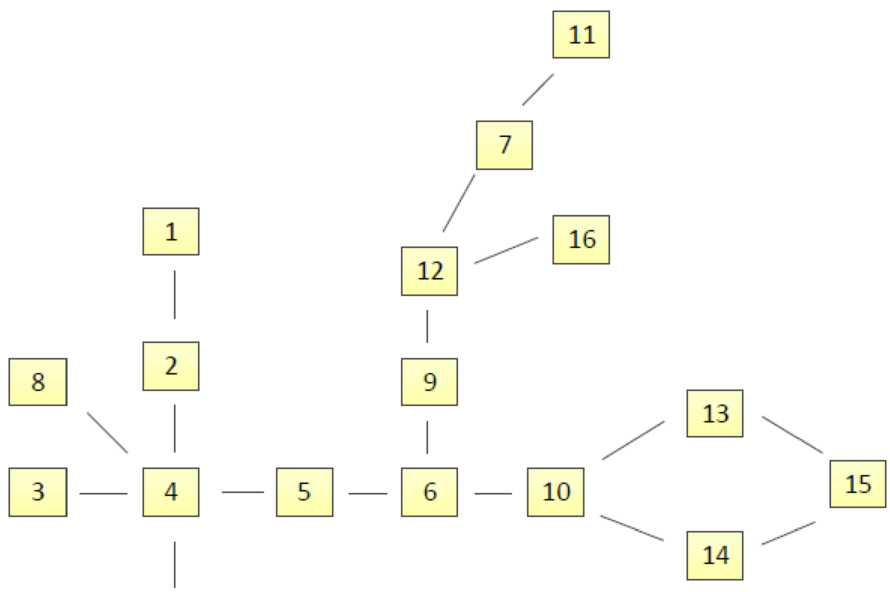

Figure 37: Structure tree of constraint and predicate relations of objects

Figure 37 and the rules of EBD methodology defined an object list for questioning as provided in Table 27. Table 28 summarizes the generic questions that were defined by the rules and templates that followed the EBD methodology and environment analysis presented in Section 2.3. These questions define the boundaries but do not filter out important information for the analysis.

Questioning object list	2, 8, 2&8&4, 11&7, 16&12, 2&4&5&6&9&12&7&11, 2&4&5&6&9&16&12, 2&4&5&6&10&15&13, 2&4&5&6&10&15&14

Table 27: Object list for questions generation

#	Component/s	Question
1	2	T1: What is aircraft?
2	8	T1: What is capability?
3	2&8&4	T4: What do you mean by aircraft has capability? How/why/when/where does aircraft have capability?
4	11&7	T2: What is foreseeable landing?
5	16&12	T6: What are all conditions? In the context of landing conditions...
6	2&4&5&6&9&12&7&11	T4: How/When/Why/Where does aircraft have to stop during foreseeable landing conditions?
7	2&4&5&6&9&16&12	T4: How/When/Why/Where does aircraft have to stop during all conditions?
8	2&4&5&6&10&15&13	T4: How/When/Why/Where does aircraft have to stop within defined distance?
9	2&4&5&6&10&15&14	T4: How/When/Why/Where does aircraft have to stop within defined runway?

Table 28: Generic question representation for environment of the main and constrain words of the requirement

1. What is aircraft?

   An aircraft is a machine that is able to fly through the air by creating a pressure difference using a specific shape (Dictionary, 2015).

2. What is capability?

   "An expression of a system, product, function or process' ability to achieve a specific objective under stated conditions" (INCOSE, 2006).

3. What do you mean by aircraft has capability? How/why/when/where does aircraft have capability?

   Aircraft has capability of Load, Taxi, Take Off, Climb/Abort, Cruise, Descent, Approach, Divert, Landing, Decelerating, Unload, Communicate and Maneuver. These capabilities are achieved by aircraft functions. The design process defines the functions for the aircraft by analyzing requirements, constraints, entities, attributes, features, and interactions. The functions allow the aircraft to perform its missions (task, action, or activity).

4. What is foreseeable landing?

   "Landing is the horizontal distance necessary to land and to come to a complete stop from a point 50 feet above the landing surface" (U.S. Code of Federal Regulations, 2020). It is accomplished by slowing down and descending to the runway (CICTT, 2013). This phase includes the following sub-phases:

   • Flare is transition from nose-low to nose-up attitude until touchdown position.

   • Landing roll is the step when aircraft exits the landing runway or comes to a stop.

   The speed reduction is accomplished by reducing thrust and/or inducing a greater amount of drag using flaps, landing gear or speed brakes (Gage Educational Pub, 1994), (Crane & Crane, 1997).

5. What are all conditions? In the context of landing conditions...

   The conditions include: the plane size and weight, the runway length and runway conditions (e.g. smooth, dry, hard-surfaced), obstacles, ground effects, certain weather conditions (e.g. wind, crosswind, precipitation (e.g. icing, snow, water and fog)), runway altitude, air temperature, air pressure, air traffic control, visibility, avionics and the overall situation (U.S. Code of Federal Regulations, 2020), (Federal Aviation Administration, 2016), (Flight Operations Support & Line Assistance, 2002).

6. How/When/Why/Where does the aircraft have to stop during foreseeable landing conditions?

   The aircraft must stop at the Actual Landing Distance (ALD). ALD is a measure between a point 50 feet above the runway threshold, and the point where the aircraft comes to a complete stop. Standard temperature, landing configuration, stabilized approach at VLS or VMCL, non-excessive vertical acceleration, runway conditions, acceptable pressures on the wheel braking systems, spoilers and reversers all effect ALD (Flight Operations Support & Line Assistance, 2002).

7. How/When/Why/Where does aircraft have to stop during all conditions?

   "The aircraft must stop at the runway within all weather and runway surface conditions (dry, wet and contaminated runways (smooth, grooved, iced and non-iced))" (U.S. Code of Federal Regulations, 2020).

8. How/When/Why/Where does the aircraft have to stop within a defined distance?
   and
9. How/When/Why/Where does the aircraft have to stop within a defined runway?

   Engine power, brakes and mechanical spoilers are the methods for deceleration of the aircraft. The deceleration function has two sub-functions: primary stopping force (wheel brake system) and secondary stopping force (thrust reverser). Entities for decelerating the aircraft on the ground include the: wheel brake system, thrust reverser, spoilers, engine controls and structural integrity (landing gear, fuselage, etc.). In addition, the aircraft shall have auto brake, anti-skid and hydraulically-driven brake systems and the pilot shall be allowed to override the autobrake function. Quantified requirements for deceleration include landing weight, approach speed, deceleration method used, pilot technique and environmental conditions (SAE Aerospace, 2011), (SkyBrary, 2019), (Flight Operations Support & Line Assistance, 2002).

By analyzing information that was found, the main points summarized in the final informational section for the general question asking step are as follows:

1. An aircraft flies through the air by creating a pressure difference using a specific shape. 2. It uses static lift or dynamic lift of an airfoil, or the downward thrust from jet engines. 3. Aircraft capabilities are Load, Taxi, Take Off, Climb/Abort, Cruise, Descent, Approach, Divert, Landing, Decelerating, Unload, Communicate and Maneuver. 4. The aircraft has functions (task, action, or activity) to complete a defined mission under stated conditions. 5. The landing mission is accomplished by slowing down and descending to the runway. 6. This is accomplished by reducing thrust and/or increasing a drag using flaps, landing gear or speed brakes. 7. Important entities for landing are plane size, weight, runway characteristics (altitude, length) and conditions (dry, wet, hard-surfaced, contaminated: smooth, grooved, iced), obstacles, ground effects, weather (wind, crosswind, precipitation (icing, snow, water/rain), fog…), runway, air temperature, air pressure, air traffic control, visibility, avionics. 8. Technical entities for the landing process include stabilized approach at VLS or VMCL (approach speed), deceleration method used, pilot technique, wheel braking systems and its' pressure, brake temperatures, spoilers and reversers, engine controls and structural integrity (landing gear, fuselage…). 9. ALD (Actual Landing Distance) is the horizontal distance necessary to land and to come to a complete stop from a point 50 feet above the landing surface. 10. This includes flare (transition from nose-low to nose-up attitude until touchdown position) and landing roll (when the aircraft exits the landing runway or comes to a stop). 11. In addition, the aircraft can have auto brake, anti-skid

and hydraulically-driven brake systems. 12. The aircraft pilot can override the autobrake function.

The section introduces information from the general environment of approach and landing processes. This step makes it easy to recognize the active verbs or nouns. These words are highlighted in green for emphasis. The specific nouns by rephrasing to the verbs have the meaning of active words and can be useful in the definition of interactions and performances following the EBD methodology. Interactions here are simple sentences that include active verbs from the general section and relative nouns and/or extension for some descriptive words in a specific environment. Performances are the relationships within interactions. Table 29 lists definitions of 19 interactions:

Interaction Number	Interaction description
In#1	Aircraft flies
In#2	An aircraft is creating pressure difference
In#3	An aircraft using a specific shape
In#4	An aircraft using lift
In#5	An aircraft using the downward thrust
In#6	An aircraft completes defined mission
In#7	An aircraft accomplishes landing mission
In#8	An aircraft is slowing down
In#9	An aircraft is descending to the runway
In#10	Reducing thrust or increasing a drag accomplishes slowing down and descending
In#11	An aircraft is using flaps, landing gear or speed brakes
In#12	Landing process include stabilized approach
In#13	An aircraft stabilizes speed
In#14	Pilot uses deceleration method and technique
In#15	An aircraft lands within a given distance
In#16	An aircraft stops within a given distance
In#17	An aircraft performs Flare
In#18	Aircraft exits the landing runway
In#19	Pilot override the autobrake

Table 29: List of defined interactions

The general idea of the approach and landing processes can be understood by following the information provided in the general section and questions answering steps. It facilitates recognizing the relationships within interactions. In this process, each interaction with another from the 19 is analyzed. This analysis created 19 tables. The presence of a relationship is indicated by the number "4" for indication purposes only. In the same way the absence of any relationship is indicated by the number "0" in the "Causal" column. It should be noted that only the first three tables, namely, Tables 30, 31 and 32 are presented for discussion purposes; other tables are included in the Appendix.

Interaction	Interaction	Causal
In#1: Aircraft flies	In#2: An aircraft is creating pressure difference.	0
	In#3: An aircraft using a specific shape.	0
	In#4: An aircraft using static and dynamic lift	4
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	0
	In#7: An aircraft accomplishes landing mission.	0
	In#8: An aircraft is slowing down	0
	In#9: An aircraft is descending to the runway	0
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	4
	In#14: Pilot uses deceleration method and technique	0
	In#15: An aircraft lands within a given distance	0
	In#16: An aircraft stops within a given distance	0
	In#17: An aircraft performs Flare	0
	In#18: Aircraft exits the landing runway	0
	In#19: Pilot override the autobrake	0

Table 30: Representation of relationships of interaction#1 with other interactions

Interaction	Interaction	Causal
In#2: An aircraft is creating pressure difference.	In#1: Aircraft flies	4
	In#3: An aircraft using a specific shape.	0
	In#4: An aircraft using static and dynamic lift	4
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	4
	In#7: An aircraft accomplishes landing mission.	0

	In#8: An aircraft is slowing down	0
	In#9: An aircraft is descending to the runway	0
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	0
	In#14: Pilot uses deceleration method and technique	0
	In#15: An aircraft lands within a given distance	0
	In#16: An aircraft stops within a given distance	0
	In#17: An aircraft performs Flare	0
	In#18: Aircraft exits the landing runway	0
	In#19: Pilot override the autobrake	0

Table 31: Representation of relationships of interaction #2 with other interactions

Interaction	Interaction	Causal
In#3: An aircraft using a specific shape.	In#1: Aircraft flies	4
	In#2: An aircraft is creating pressure difference.	4
	In#4: An aircraft using static and dynamic lift	4
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	4
	In#7: An aircraft accomplishes landing mission.	4
	In#8: An aircraft is slowing down	0
	In#9: An aircraft is descending to the runway	0
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	4
	In#14: Pilot uses deceleration method and technique	0
	In#15: An aircraft lands within a given distance	0
	In#16: An aircraft stops within a given distance	0
	In#17: An aircraft performs Flare	4
	In#18: Aircraft exits the landing runway	0
	In#19: Pilot override the autobrake	0

Table 32: Representation of relationships of interaction #3 with other interactions

Table 33 represents all 19 defined interactions and existing relationships. For a better visualization, the defined relationships using number “4” are highlighted in green.

	In #1	In #2	In #3	In #4	In #5	In #6	In #7	In #8	In #9	In# 10	In# 11	In# 12	In# 13	In# 14	In# 15	In# 16	In# 17	In# 18	In# 19
In# 1	0	0	0	4	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
In# 2	4	0	0	4	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
In# 3	4	4	0	4	0	4	4	0	0	0	0	0	4	0	0	0	4	0	0
In# 4	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
In# 5	4	4	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0
In# 6	4	0	0	0	0	0	4	4	4	0	0	0	4	0	4	4	4	0	0
In# 7	0	0	0	0	0	0	0	0	4	0	0	0	0	4	0	0	0	0	0
In# 8	0	0	4	0	0	0	0	0	4	0	0	0	0	4	0	4	0	4	0
In# 9	0	0	4	0	0	0	4	0	0	0	0	0	0	4	0	0	0	0	0
In# 10	0	0	0	0	0	4	0	4	4	0	0	0	0	0	0	0	0	0	0
In# 11	0	0	4	0	0	0	4	4	0	0	0	0	0	0	0	4	0	0	4
In# 12	0	0	4	0	0	0	4	0	4	0	0	0	4	4	4	0	4	0	0
In# 13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0
In# 14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
In# 15	0	0	0	0	0	0	4	0	0	0	0	0	0	4	0	4	0	0	0
In# 16	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	4	4
In# 17	0	0	0	0	0	0	0	0	4	0	0	0	0	0	4	0	0	0	0
In# 18	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0
In# 19	0	0	0	0	0	0	0	4	0	0	0	0	0	4	0	0	0	0	0

Table 33: Summary of relationships between all interactions

3.3.2 Identification of dependent interactions

The purpose of this step of analysis that is summarized in the Sections 3.3.3 and 3.3.4, is to focus the tagging by key words that will lead to the domain specific requirements that correlate with the subsystems of the aircraft in order to produce the result of "approach and landing". First, this step represents all interactions that are listed in Table 33 in a general performance network (Figure 38). Second, the process identifies independent conflicts in order to filter unnecessary interactions. This is a repetitive process toward representation only of dependent interactions in the performance network. Those interactions suggests an environment analysis for the components and propose the key words for tagging.

The independent conflicts occur in the interactions that feed others but are not dependent on any other interactions. For example, 4 independent conflicts φ5, φ10, φ11, φ12 are shown in Figure 38. These independent conflicts are then removed to present the performance network for the next iteration.

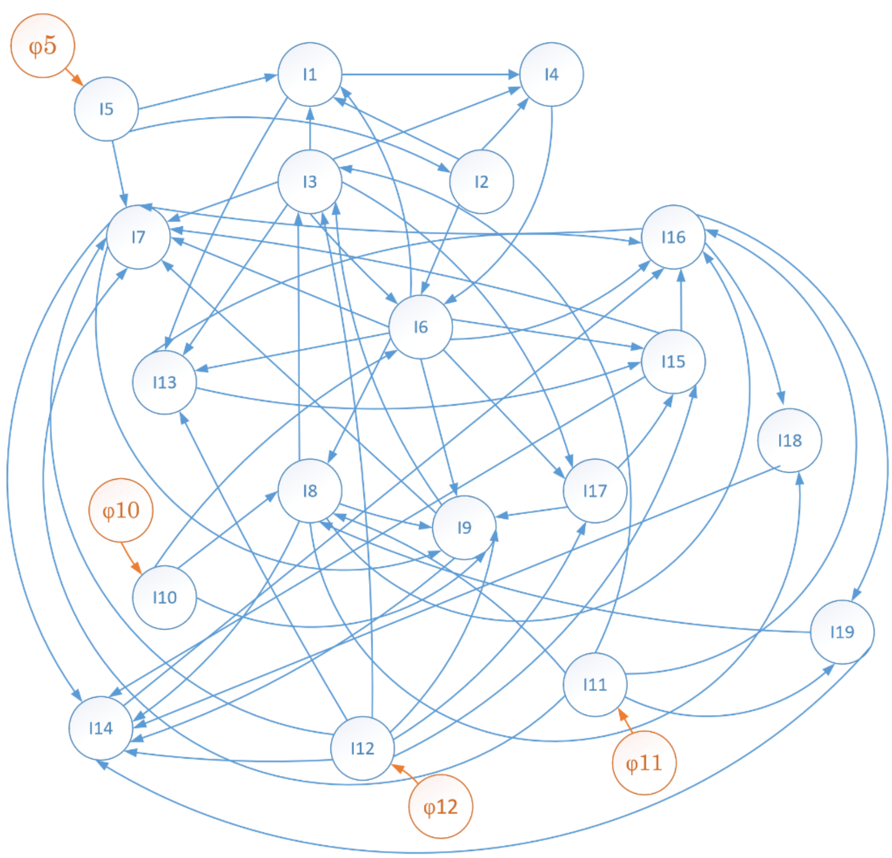

Figure 38: General performance network representation

Figure 39 represents the updated performance network with a new independent conflict φ2 that has appeared. In addition, the relevance of In#16 on In#19 was checked. In#16 is "An aircraft stops within a given distance". It is not necessary that In#16 will lead to In#19 "Pilot overrides the autobrake". In this case, the connection between these interactions can be canceled because the issue is irrelevant. Cancelation of the connection causes new independence conflict φ19 to appear.

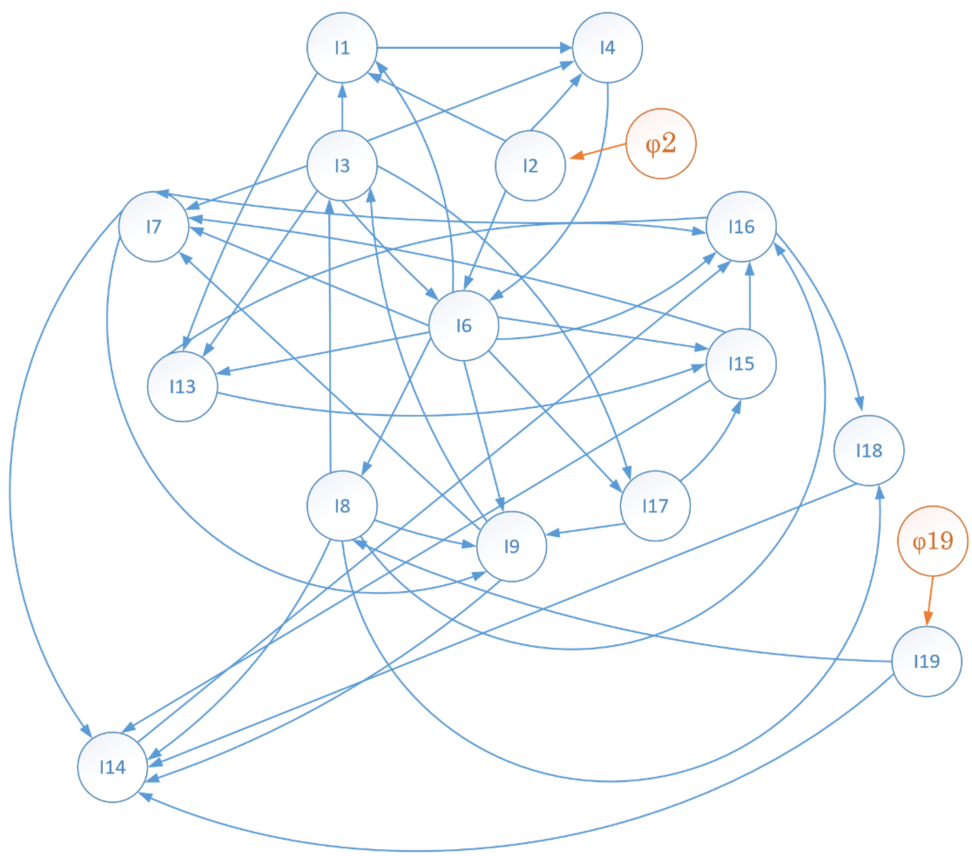

Figure 39: Performance network after the second iteration

Conflicts φ2 and φ19 have been removed. Figure 40 represents the updated performance network.

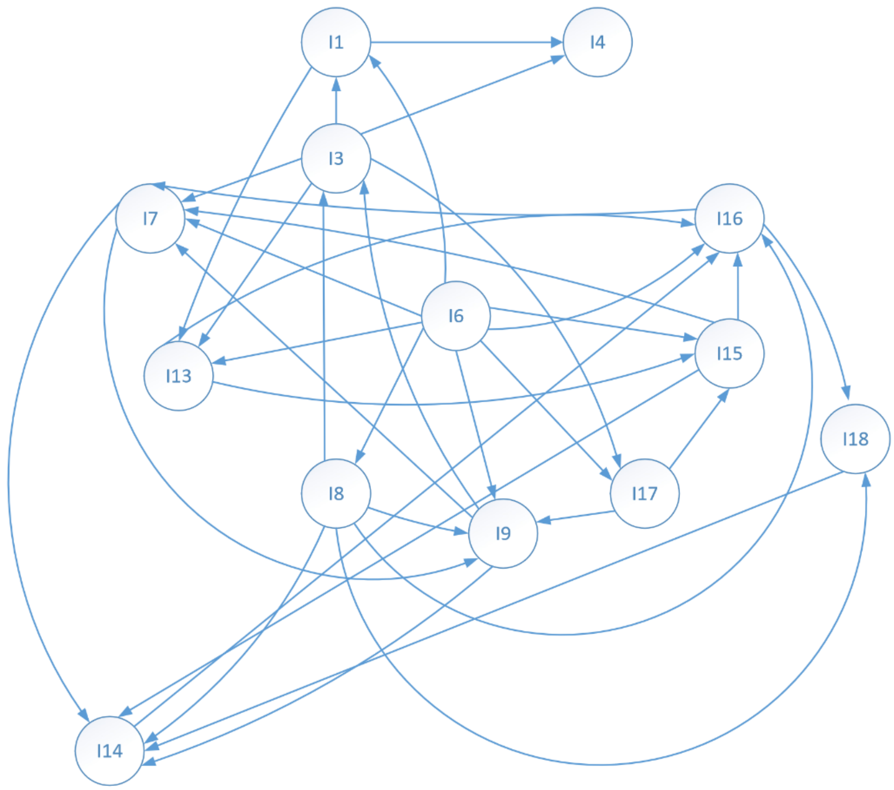

Figure 40: Updated performance network after the second iteration

The next step examines the connections of In#3 "An aircraft using a specific shape" and In#4 "An aircraft using static and dynamic lift" with In#6 "An aircraft completes defined mission". It is found that In#3 and In#4 do not necessarily lead to In#6. The connections In#3 and In#4 with In#6 are found to be irrelevant and are canceled. A new conflict, φ6, is found. In addition, following this, a conflict named φ8 appears; see Figure 41.

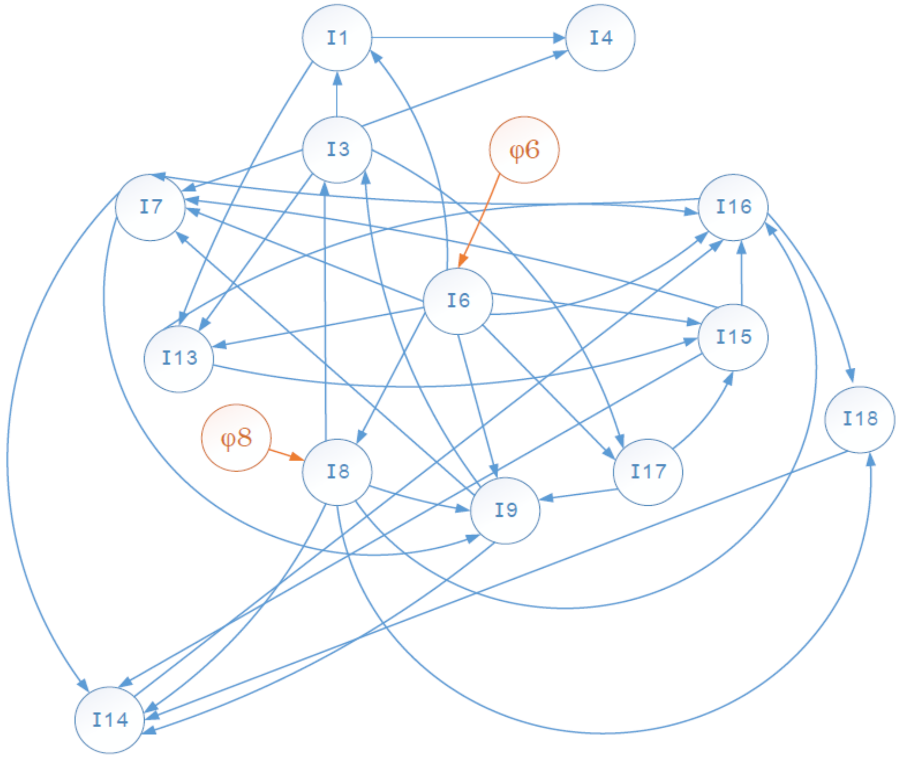

Figure 41: Updated performance network after the third iteration

Conflict φ6 was independent and then removed. Following this, conflict φ8 was removed for the same reason. The updated performance network is given in Figure 42.

This is the final performance network. It summarizes only dependent interactions, corresponding to those that work together and simultaneously.

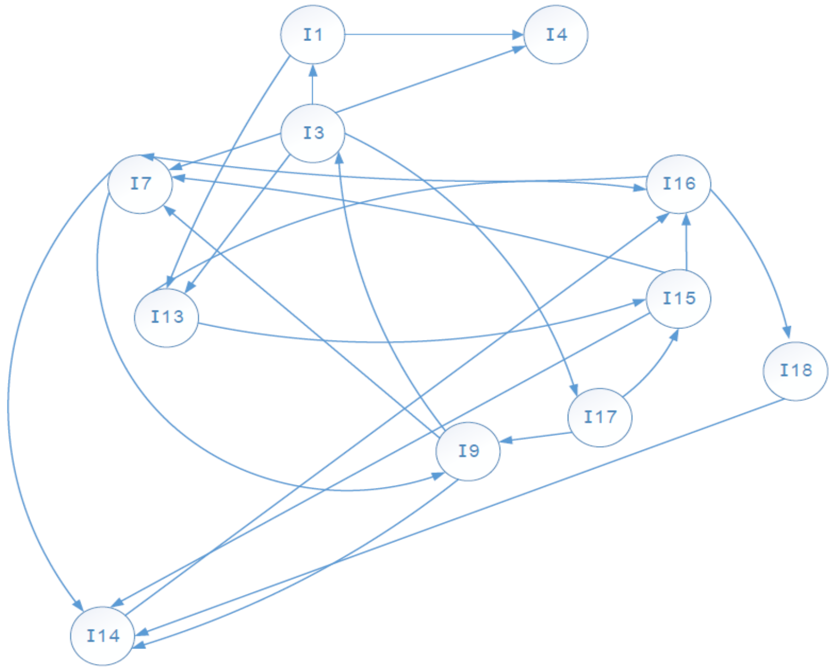

Figure 42: Final performance network

3.3.3 Environment analysis and key words representation for the searching process

Table 34 summarizes the list of dependent interactions. In the interaction description column are highlighted components that were noted in the updated ROM diagram. This ROM diagram includes only relevant original sentences from the section and decomposition of some components to the level of domain-specific components in their environment; see Figure 43.

The highlighted components, the main constrained ones and the components that are found within the decomposition step (e.g. systems) represent key words for tagging that can be used in the searching process for requirements that are relevant for approach and landing.

Interaction Number	Interaction description
In#1	Aircraft flies
In#3	An aircraft using a specific shape.
In#4	An aircraft using lift
In#7	An aircraft accomplishes landing mission.
In#9	An aircraft is descending to the runway
In#13	An aircraft stabilizes speed
In#14	Pilot uses deceleration method and technique
In#15	An aircraft lands within a given distance
In#16	An aircraft stops within a given distance
In#17	An aircraft performs Flare
In#18	Aircraft exits the landing runway

Table 34: Final list of dependent interactions

The key words are: lift, air, shapes, land, runway, landing roll, attitude, stop, speed, method (in meaning of deceleration method), technique, systems.

The decomposition process of component “systems” to domain-specific components represents the following additional components/key words: spoilers, reversers, engine controls, wheel braking, auto-brake, brake pedal, hydraulic, and antiskid.

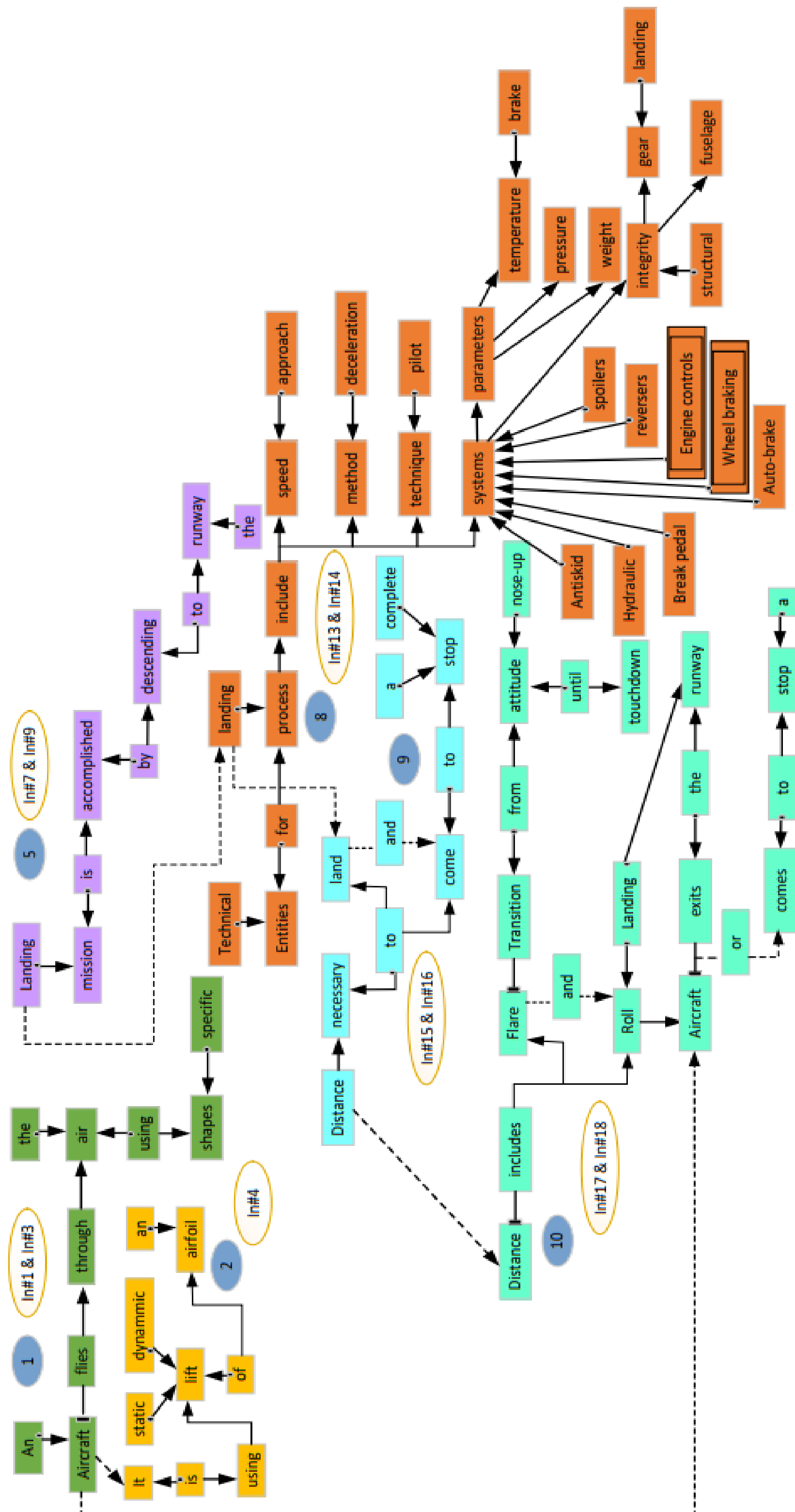

Figure 43: ROM diagram with key words for tagging purposes in the searching process

3.3.4 Evaluation versus model (Arcadia method) representation

For the purposes of this study, the Arcadia method is used to identify the main components/entities that are present within the landing and deceleration processes. The modeling is performed by following the steps suggested by: (Arcadia, 2019), (Bonnet, 2015), (Roques, 2018), (Voirin, 2018), (Raymer, 2012) (Flight Operations Support & Line Assistance, 2002), (SAE Aerospace, 2011), and (Federal Aviation Administration, 2016). The analysis procedure is outlined below,

Step 1: Definition of the general operational entities, actors and capabilities.

a. Operational entities and actors are defined as follows:
    i. Actors: pilot and air traffic controller
    ii. Aircraft: technical entities (transponder, radio, braking systems, landing flaps, landing gear, engine controls, thrust reversers, spoilers, etc.).
    iii. Weather
    iv. Components in environment: runway characteristics and conditions, obstacles, air temperature, air pressure, visibility, radar.

b. Operational capabilities and relationships with actors are defined as follows in Figure 44:

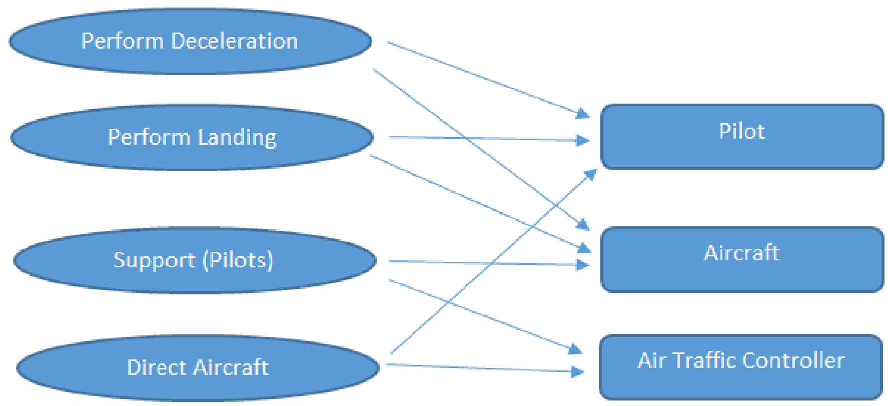

Figure 44: General Operational Capabilities within landing and deceleration processes and relationships with actors

Step 2: Definition of high-level expected activities for each of the actors. Figure 45 indicates those activities as follows:

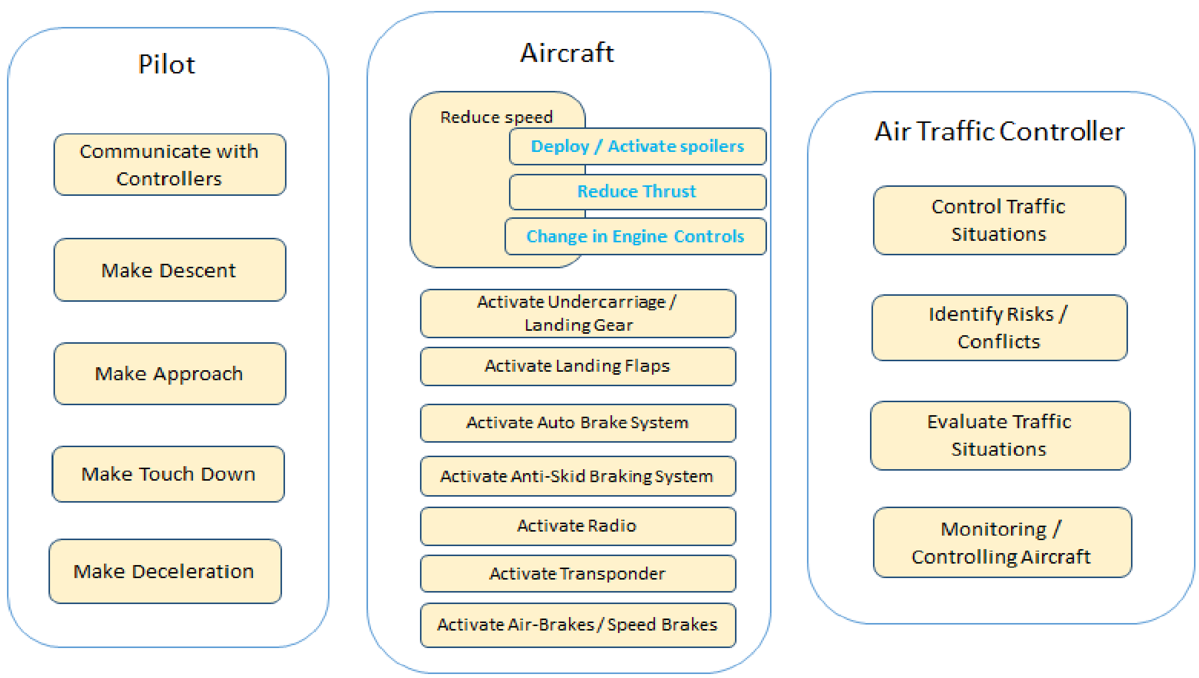

Figure 45: High-level expected activities by actors within the landing and deceleration processes

Step 3: Representation of combined information in the overview of operational activities and entities. Figure 46 represents the model with all relevant information including actors, activities and entities.

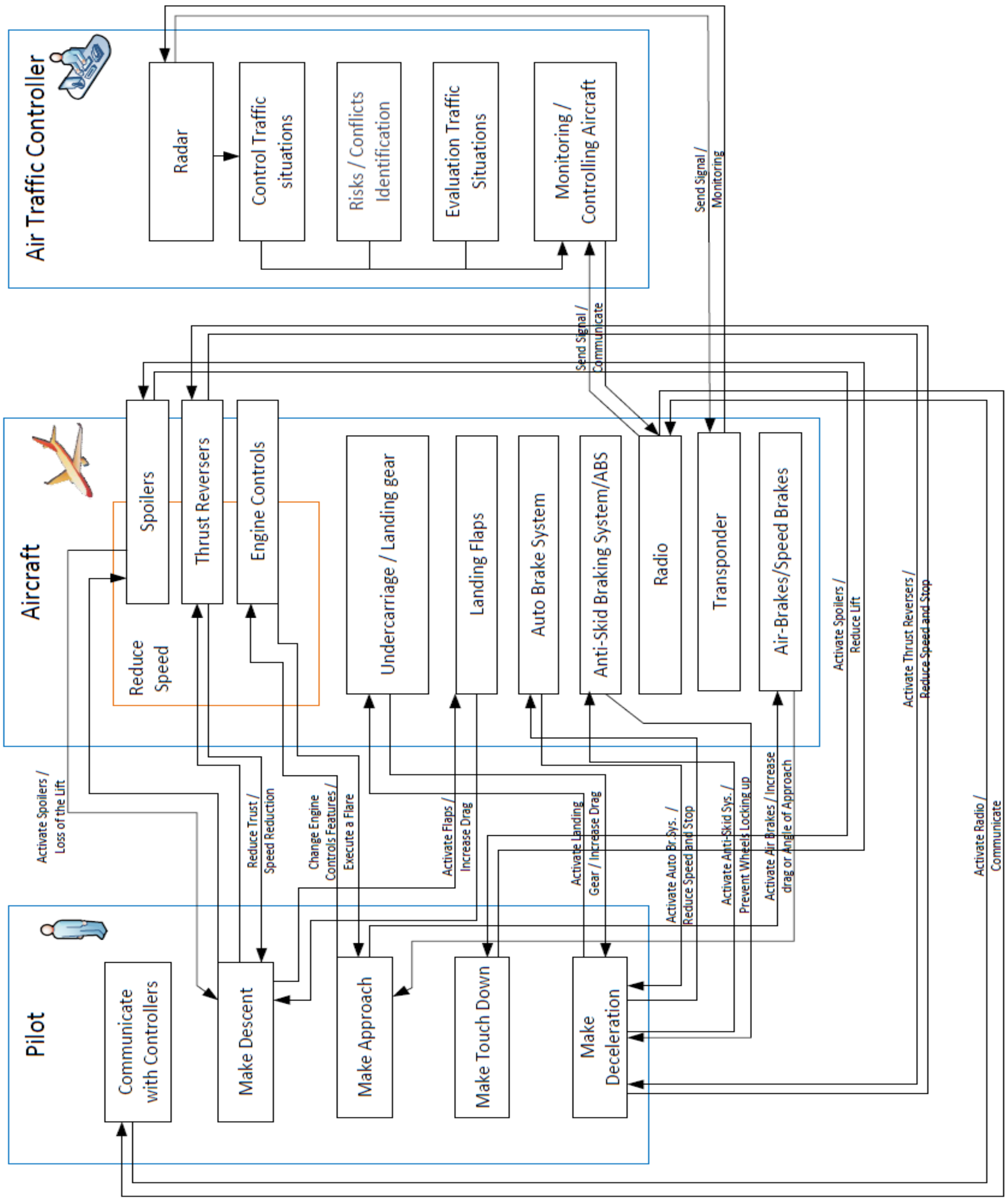

Figure 46: Overview of Operational Activities and Entities within the landing and deceleration processes

The evaluation compares the results of the main key words for the “Systems” found within the EBD analysis and represented by the ROM diagram in Figure 43, with the components of the “System” (e.g. aircraft) that are represented by the modeling process in Figure 46. The comparison shows that the EBD results match approximately 86% of the model analysis results, i.e., 6 out of the 7 components.

3.4 Summary for top level requirement analysis by the EBD methodology

The EBD methodology is an effective and efficient method for the analysis of top level requirements, and is able to represent a wide range of information in the relevant information stream of any domain.

CHAPTER 4 Conclusions

4.1 Summary of research results

Developing an Aircraft Design Requirements Ontology (ADRO) is a complex and evolving process that never stops. This process has many iteration steps and can be improved continually over time. The ADRO can be modified and adapted to the existing reality and to organizational needs.

Implementation of the EBD methodology in this research has provided an opportunity to define the knowledge components and relationships for ADRO. These standardize the method for structuring the organizational knowledge (e.g. requirements). In addition, an example of the analysis for the top level requirement, based on the EBD methodology, has been given. The analysis summarized the key words for the tagging process that can be used in the requirements searching process. Results from this analysis were compared with those from the Arcadia modeling process, and the matching was close.

Hence, we can suggest that implementing the EBD methodology for creating solutions orients users in the right direction to extract information. EBD is sufficient and necessary in development processes. This methodology guides users step-by-step and can be effectively implemented in any area.

4.2 Recommendations for future research

The results of this study can be extended and used in future studies to define a framework for creating a Model-Based Environment for Systems Engineering. It can also offer a framework for software development for knowledge searching processes. In addition, the knowledge identified in this research can suggest directions for creating model's traceability.

Appendix

This Appendix presents additional tables of relationships between interactions to supplement those that were introduced in Section 3.3.2.

Interaction	Interaction	Causal
In#4: An aircraft using static and dynamic lift	In#1: Aircraft flies	0
	In#2: An aircraft is creating pressure difference.	0
	In#3: An aircraft using a specific shape.	0
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	4
	In#7: An aircraft accomplishes landing mission.	0
	In#8: An aircraft is slowing down	0
	In#9: An aircraft is descending to the runway	0
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	0
	In#14: Pilot uses deceleration method and technique	0
	In#15: An aircraft lands within a given distance	0
	In#16: An aircraft stops within a given distance	0
	In#17: An aircraft performs Flare	0
	In#18: Aircraft exits the landing runway	0
	In#19: Pilot overrides the autobrake	0

Table 35: Representation of relationships of interaction#4 with other interactions

Interaction	Interaction	Causal
In#5: An aircraft using the downward thrust	In#1: Aircraft flies	4
	In#2: An aircraft is creating pressure difference.	4
	In#3: An aircraft using a specific shape.	0
	In#4: An aircraft using static and dynamic lift	0
	In#6: An aircraft completes defined mission	0
	In#7: An aircraft accomplishes landing mission.	4
	In#8: An aircraft is slowing down	0
	In#9: An aircraft is descending to the runway	0
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	0
	In#14: Pilot uses deceleration method and technique	0
	In#15: An aircraft lands within a given distance	0
	In#16: An aircraft stops within a given distance	0

	In#17: An aircraft performs Flare	0
	In#18: Aircraft exits the landing runway	0
	In#19: Pilot overrides the autobrake	0

Table 36: Representation of relationships of interaction#5 with other interactions

Interaction	Interaction	Causal
In#6: An aircraft completes defined mission	In#1: Aircraft flies	4
	In#2: An aircraft is creating pressure difference.	0
	In#3: An aircraft using a specific shape.	0
	In#4: An aircraft using static and dynamic lift	0
	In#5: An aircraft using the downward thrust	0
	In#7: An aircraft accomplishes landing mission.	4
	In#8: An aircraft is slowing down	4
	In#9: An aircraft is descending to the runway	4
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	4
	In#14: Pilot uses deceleration method and technique	0
	In#15: An aircraft lands within a given distance	4
	In#16: An aircraft stops within a given distance	4
	In#17: An aircraft performs Flare	4
	In#18: Aircraft exits the landing runway	0
	In#19: Pilot overrides the autobrake	0

Table 37: Representation of relationships of interaction#6 with other interactions

Interaction	Interaction	Causal
In#7: An aircraft accomplishes landing mission.	In#1: Aircraft flies	0
	In#2: An aircraft is creating pressure difference.	0
	In#3: An aircraft using a specific shape.	0
	In#4: An aircraft using static and dynamic lift	0
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	0
	In#8: An aircraft is slowing down	0
	In#9: An aircraft is descending to the runway	4
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	0
	In#14: Pilot uses deceleration method and technique	4
	In#15: An aircraft lands within a given distance	0
	In#16: An aircraft stops within a given distance	0

	In#17: An aircraft performs Flare	0
	In#18: Aircraft exits the landing runway	0
	In#19: Pilot overrides the autobrake	0

Table 38: Representation of relationships of interaction#7 with other interactions

Interaction	Interaction	Causal
In#8: An aircraft is slowing down	In#1: Aircraft flies	0
	In#2: An aircraft is creating pressure difference.	0
	In#3: An aircraft using a specific shape.	4
	In#4: An aircraft using static and dynamic lift	0
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	0
	In#7: An aircraft accomplishes landing mission.	0
	In#9: An aircraft is descending to the runway	4
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	0
	In#14: Pilot uses deceleration method and technique	4
	In#15: An aircraft lands within a given distance	0
	In#16: An aircraft stops within a given distance	4
	In#17: An aircraft performs Flare	0
	In#18: Aircraft exits the landing runway	4
	In#19: Pilot overrides the autobrake	0

Table 39: Representation of relationships of interaction#8 with other interactions

Interaction	Interaction	Causal
In#9: An aircraft is descending to the runway	In#1: Aircraft flies	0
	In#2: An aircraft is creating pressure difference.	0
	In#3: An aircraft using a specific shape.	4
	In#4: An aircraft using static and dynamic lift	0
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	0
	In#7: An aircraft accomplishes landing mission.	4
	In#8: An aircraft is slowing down	0
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	0
	In#14: Pilot uses deceleration method and technique	4
	In#15: An aircraft lands within a given distance	0
	In#16: An aircraft stops within a given distance	0

	In#17: An aircraft performs Flare	0
	In#18: Aircraft exits the landing runway	0
	In#19: Pilot overrides the autobrake	0

Table 40: Representation of relationships of interaction#9 with other interactions

Interaction	Interaction	Causal
In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	In#1: Aircraft flies	0
	In#2: An aircraft is creating pressure difference.	0
	In#3: An aircraft using a specific shape.	0
	In#4: An aircraft using static and dynamic lift	0
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	4
	In#7: An aircraft accomplishes landing mission.	0
	In#8: An aircraft is slowing down	4
	In#9: An aircraft is descending to the runway	4
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	0
	In#14: Pilot uses deceleration method and technique	0
	In#15: An aircraft lands within a given distance	0
	In#16: An aircraft stops within a given distance	0
	In#17: An aircraft performs Flare	0
	In#18: Aircraft exits the landing runway	0
	In#19: Pilot overrides the autobrake	0

Table 41: Representation of relationships of interaction#10 with other interactions

Interaction	Interaction	Causal
In#11: An aircraft is using flaps, landing gear or speed brakes	In#1: Aircraft flies	0
	In#2: An aircraft is creating pressure difference.	0
	In#3: An aircraft using a specific shape.	4
	In#4: An aircraft using static and dynamic lift	0
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	0
	In#7: An aircraft accomplishes landing mission.	4
	In#8: An aircraft is slowing down	4
	In#9: An aircraft is descending to the runway	0
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	0
	In#14: Pilot uses deceleration method and technique	0
	In#15: An aircraft lands within a given distance	0
	In#16: An aircraft stops within a given distance	4
	In#17: An aircraft performs Flare	0
	In#18: Aircraft exits the landing runway	0

	In#19: Pilot overrides the autobrake	4

Table 42: Representation of relationships of interaction#11 with other interactions

Interaction	Interaction	Causal
In#12: Landing process include stabilized approach	In#1: Aircraft flies	0
	In#2: An aircraft is creating pressure difference.	0
	In#3: An aircraft using a specific shape.	4
	In#4: An aircraft using static and dynamic lift	0
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	0
	In#7: An aircraft accomplishes landing mission.	4
	In#8: An aircraft is slowing down	0
	In#9: An aircraft is descending to the runway	4
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#13: An aircraft stabilizes speed	4
	In#14: Pilot uses deceleration method and technique	4
	In#15: An aircraft lands within a given distance	4
	In#16: An aircraft stops within a given distance	0
	In#17: An aircraft performs Flare	4
	In#18: Aircraft exits the landing runway	0
	In#19: Pilot overrides the autobrake	0

Table 43: Representation of relationships of interaction#12 with other interactions

Interaction	Interaction	Causal
In#13: An aircraft stabilizes speed	In#1: Aircraft flies	0
	In#2: An aircraft is creating pressure difference.	0
	In#3: An aircraft using a specific shape.	0
	In#4: An aircraft using static and dynamic lift	0
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	0
	In#7: An aircraft accomplishes landing mission.	0
	In#8: An aircraft is slowing down	0
	In#9: An aircraft is descending to the runway	0
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#14: Pilot uses deceleration method and technique	0
	In#15: An aircraft lands within a given distance	4
	In#16: An aircraft stops within a given distance	4
	In#17: An aircraft performs Flare	0
	In#18: Aircraft exits the landing runway	0

	In#19: Pilot overrides the autobrake	0

Table 44: Representation of relationships of interaction#13with other interactions

Interaction	Interaction	Causal
In#14: Pilot uses deceleration method and technique	In#1: Aircraft flies	0
	In#2: An aircraft is creating pressure difference.	0
	In#3: An aircraft using a specific shape.	0
	In#4: An aircraft using static and dynamic lift	0
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	0
	In#7: An aircraft accomplishes landing mission.	4
	In#8: An aircraft is slowing down	0
	In#9: An aircraft is descending to the runway	0
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	0
	In#15: An aircraft lands within a given distance	0
	In#16: An aircraft stops within a given distance	4
	In#17: An aircraft performs Flare	0
	In#18: Aircraft exits the landing runway	0
	In#19: Pilot overrides the autobrake	0

Table 45: Representation of relationships of interaction#14 with other interactions

Interaction	Interaction	Causal
In#15: An aircraft lands within a given distance	In#1: Aircraft flies	0
	In#2: An aircraft is creating pressure difference.	0
	In#3: An aircraft using a specific shape.	0
	In#4: An aircraft using static and dynamic lift	0
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	0
	In#7: An aircraft accomplishes landing mission.	4
	In#8: An aircraft is slowing down	0
	In#9: An aircraft is descending to the runway	0
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	0
	In#14: Pilot uses deceleration method and technique	4
	In#16: An aircraft stops within a given distance	4
	In#17: An aircraft performs Flare	0
	In#18: Aircraft exits the landing runway	0

	In#19: Pilot overrides the autobrake	0

Table 46: Representation of relationships of interaction#15 with other interactions

Interaction	Interaction	Causal
In#16: An aircraft stops within a given distance	In#1: Aircraft flies	0
	In#2: An aircraft is creating pressure difference.	0
	In#3: An aircraft using a specific shape.	0
	In#4: An aircraft using static and dynamic lift	0
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	0
	In#7: An aircraft accomplishes landing mission.	4
	In#8: An aircraft is slowing down	0
	In#9: An aircraft is descending to the runway	0
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	0
	In#14: Pilot uses deceleration method and technique	0
	In#15: An aircraft lands within a given distance	0
	In#17: An aircraft performs Flare	0
	In#18: Aircraft exits the landing runway	4
	In#19: Pilot overrides the autobrake	4

Table 47: Representation of relationships of interaction#16 with other interactions

Interaction	Interaction	Causal
In#17: An aircraft performs Flare	In#1: Aircraft flies	0
	In#2: An aircraft is creating pressure difference.	0
	In#3: An aircraft using a specific shape.	0
	In#4: An aircraft using static and dynamic lift	0
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	0
	In#7: An aircraft accomplishes landing mission.	0
	In#8: An aircraft is slowing down	0
	In#9: An aircraft is descending to the runway	4
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	0
	In#14: Pilot uses deceleration method and technique	0
	In#15: An aircraft lands within a given distance	4
	In#16: An aircraft stops within a given distance	0
	In#18: Aircraft exits the landing runway	0

	In#19: Pilot overrides the autobrake	0

Table 48: Representation of relationships of interaction#17 with other interactions

Interaction	Interaction	Causal
In#18: Aircraft exits the landing runway	In#1: Aircraft flies	0
	In#2: An aircraft is creating pressure difference.	0
	In#3: An aircraft using a specific shape.	0
	In#4: An aircraft using static and dynamic lift	0
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	0
	In#7: An aircraft accomplishes landing mission.	0
	In#8: An aircraft is slowing down	0
	In#9: An aircraft is descending to the runway	0
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	0
	In#14: Pilot uses deceleration method and technique	4
	In#15: An aircraft lands within a given distance	0
	In#16: An aircraft stops within a given distance	0
	In#17: An aircraft performs Flare	0
	In#19: Pilot overrides the autobrake	0

Table 49: Representation of relationships of interaction#18 with other interactions

Interaction	Interaction	Causal
In#19: Pilot overrides the autobrake	In#1: Aircraft flies	0
	In#2: An aircraft is creating pressure difference.	0
	In#3: An aircraft using a specific shape.	0
	In#4: An aircraft using static and dynamic lift	0
	In#5: An aircraft using the downward thrust	0
	In#6: An aircraft completes defined mission	0
	In#7: An aircraft accomplishes landing mission.	0
	In#8: An aircraft is slowing down	4
	In#9: An aircraft is descending to the runway	0
	In#10: Reducing thrust or increasing a drag accomplishes slowing down and descending	0
	In#11: An aircraft is using flaps, landing gear or speed brakes	0
	In#12: Landing process include stabilized approach	0
	In#13: An aircraft stabilizes speed	0
	In#14: Pilot uses deceleration method and technique	4
	In#15: An aircraft lands within a given distance	0
	In#16: An aircraft stops within a given distance	0
	In#17: An aircraft performs Flare	0

	In#18: Aircraft exits the landing runway	0

Table 50: Representation of relationships of interaction#19 with other interactions

References

Aaker, D. A. (2008). *Strategic market management* (8th ed). John Wiley & Sons,Inc.

AFRA. (2018). *Aircraft Fleet Recycling Association*. https://afraassociation.org/accreditation/the-afra-bmp/

Anjum, N., Harding, J., Young, R., Case, K., Usman, Z., & Changoora, T. (2013). Verification of knowledge shared across design and manufacture using a foundation ontology. *International Journal of Production Research*, *51*(22), 6534–6552. https://doi.org/10.1080/00207543.2013.798051

Arcadia. (2019). *Sample Model Quick Reading Guide In Flight Entertainment*. Eclipse. https://download.eclipse.org/capella/samples/IFE/Capella_IFE_Sample_Model_1.0_Quick_Reading_Guide.pdf

Arnold, S., & Lawson, H. W. (2004). Viewing systems from a business management perspective: The ISO/IEC 15288 standard. *Systems Engineering*, *7*(3), 229–242. https://doi.org/10.1002/sys.20006

Ast, M., Glas, M., & Roehm, T. (2013). *Creating an Ontology for Aircraft Design* (301356). 11.

Bonnet, S. (2015). *Sample Model Highlights In-Flight Entertainment*. http://www.polarsys.org/capella

Burgun, A., Botti, G., Le Beux, P., & Fieschi, M. (2001). Issues in the Design of Medical Ontologies Used for Knowledge Sharing. *Journal of Medical Systems*, *25*(2), 95–108. https://doi.org/10.1023/A:1005668530110

Cantot, P., & Luzeaux, D. (Eds.). (2011). *Simulation and modeling of systems of systems*. ISTE ; Wiley.

Chen, Z. Y., & Zeng, Y. (2006). Classification of Product Requirements Based on Product Environment. *Concurrent Engineering*, *14*(3), 219–230. https://doi.org/10.1177/1063293X06068389

CICTT. (2013). *Phase of Flight: Definitions and Usage Notes*. CAST/ICAO. http://www.intlaviationstandards.org/Documents/PhaseofFlightDefinitions.pdf

Crane, D., & Crane, D. (1997). *Dictionary of aeronautical terms* (3rd ed). Aviation Supplies & Academics.

Crawley, E., Cameron, B., & Selva, D. (2016). *System architecture: Strategy and product development for complex systems* (Global Edition). Pearson.

Department of Defense. (2001). *Systems Engineering Fundamentals* (Defense Acquisition University Press, Ed.). Fort Belvoir, Virginia 22060-5565.

Deutsch, S., & Pew, R. W. (2005). Single Pilot Commercial Aircraft Operation. *BBN Technologies*, 27.

Dictionary. (2015). *Aircraft—Definition at Dictionary*. Dictionary.com. https://www.dictionary.com/browse/aircraft

Dictionary. (2020). *Aerospace—Meaning in the Cambridge English Dictionary*. https://dictionary.cambridge.org/dictionary/english/aerospace

Dufresne, S. (2008). *A Hierarchical Modeling Methodology for the Definition and Selection of Requirements* [Georgia Institute of Technology]. https://scholar.google.ca/scholar?hl=en&as_sdt=0%2C5&q=A+Hierarchical+Modeling+Methodology+for+the+Definition+and+Selection+of+Requirements&btnG=

EIA. (1999). *ANSI/EIA-632, EIA STANDARD, Processes for Engineering a System*. Government Electronics and Information Technology Association Engineering Department.

Eppinger, S. D., & Browning, T. R. (2012). *Design structure matrix methods and applications*. MIT Press.

Federal Aviation Administration, F. (2016). *Airplane Flying Handbook* (B). U.S. Department of Transportation.

Flight Operations Support & Line Assistance. (2002). *Getting to Grips with Aircraft Performance*. Airbus. https://www.skybrary.aero/bookshelf/books/2263.pdf

Fosse, E. (2012). *Model-based Systems Engineering (MBSE) 101* [MBSE Workshop]. Presented at Board Meeting, MBSE Workshop.

Friedenthal, S., & Wolfrom, J. (2010). *Modeling with SysML*. INCOSE 2010 Symposium, Chicago, IL.

Gage Educational Pub (Ed.). (1994). *Flight training manual* (4th ed).

Gonzalez, A. M. (2008). *Implementation of ISO standards through formalization of requirements for process management.* Library and Archives Canada = Bibliothèque et Archives Canada.

Guarino, N. (1997). Understanding, building and using ontologies. *International Journal of Human-Computer Studies*, *46*(2–3), 293–310. https://doi.org/10.1006/ijhc.1996.0091

Gutierrez, R. (2018). *A Requirement Ontology To Guide The Analysis Of System Life Cycle Processes*. Concordia University.

Harris, D., Stanton, N. A., & Starr, A. (2015). Spot the difference: Operational event sequence diagrams as a formal method for work allocation in the development of single-pilot operations for commercial aircraft. *Ergonomics*, *58*(11), 1773–1791. https://doi.org/10.1080/00140139.2015.1044574

Hart, L. E. (2015). *Introduction To Model-Based System Engineering (MBSE) and SysML*. Delaware Valley INCOSE Chapter Meeting.

INCOSE. (2006). *Systems Engineering Handbook a Guide for System Life Cycle Processes and Activities* (C. Haskins, Ed.; version 3).

Institute of Electrical and Electronics Engineers (Ed.). (1998). *IEEE recommended practice for software requirements specifications*. IEEE.

ISO/IEC 15288: Systems engineering—System life cycle processes. (2002). *ISO/IEC 15288:2002(E)*, 1–70. https://doi.org/10.1109/IEEESTD.2001.8684399

ISO/IEC/IEEE15288: Systems and software engineering—System life cycle processes. (2015). *ISO/IEC/IEEE15288:2015(E)*, 118.

Jackson, S. (2015). *Systems Engineering for Commercial Aircraft: A domain-specific adaptation* (Second Edition). Ashgate.

Jarke, M., Neumann, B., Vassiliou, Y., & Wahlster, W. (1989). KBMS Requirements of Knowledge-Based Systems. *Springer*, 10.

Kahn, K. B. (2010). *The PDMA handbook of new product development*. Wiley.

Li, X., Ni, Y., Ming, X. G., Song, W., & Cai, W. (2015). Module-based similarity measurement for commercial aircraft tooling design. *International Journal of Production Research*, *53*(17), 5382–5397. https://doi.org/10.1080/00207543.2015.1047973

Lim, S. C. J., Liu, Y., & Lee, W. B. (2009). Multi-facet product information search and retrieval using semantically annotated product family ontology. *Elsevier Ltd.*, 479–493. https://doi.org/10.1016/j.ipm.2009.09.001

Long, D. A., & Scott, Z. B. (2011). *A primer for model-based systems engineering* (2nd ed.). Vitech.

McBurney, P. (2012). *Multi-agent systems: What Are Models for?* (M. Cossentino, M. Kaisers, K. Tuyls, & G. Weiss, Eds.; 9th European workshop, EUMAS, Maastricht, the Netherlands 2011). Springer.

Micskei, Z., & Waeselynck, H. (2011). The many meanings of UML 2 Sequence Diagrams: A survey. *Software & Systems Modeling*, *10*(4), 489–514. https://doi.org/10.1007/s10270-010-0157-9

NASA Systems Engineering Handbook NASA/SP-2007-6105 (Rev1 ed.). (2007). Washington, DC, USA, National Aeronautics and Space Administration. www.nasa.gov

NASA Systems Engineering Handbook NASA/SP-2016-6105 (Rev2 ed.). (2016). Washington, DC, USA, National Aeronautics and Space Administration. www.nasa.gov

Nilsson, P., & Fagerström, B. (2006). Managing stakeholder requirements in a product modelling system. *Computers in Industry*, *57*(2), 167–177. https://doi.org/10.1016/j.compind.2005.06.003

Noy, N. F., & McGuinness, D. L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. *Stanford University*, 25.

Piperni, P., Abdo, M., Kafyeke, F., & Isikveren, A. T. (2007). Preliminary Aerostructural Optimization of a Large Business Jet. *Journal of Aircraft*, *44*(5), 1422–1438. https://doi.org/10.2514/1.26989

Piperni, P., DeBlois, A., & Henderson, R. (2013). Development of a Multilevel Multidisciplinary-Optimization Capability for an Industrial Environment. *AIAA Journal*, *51*(10), 2335–2352. https://doi.org/10.2514/1.J052180

Pop, R. V. (2011). Understanding of ISO 9000 Standards through Recursive Object Modeling (ROM). *Concordia University, Montreal*, 113.

Raymer, D. P. (2012). *Aircraft design: A conceptual approach* (5th ed). American Institute of Aeronautics and Astronautics.

Rodriguez, S. (2014). *TOP 10 FAILED DEFENSE PROGRAMS OF THE RMA ERA*. War on the rocks. https://warontherocks.com/2014/12/top-10-failed-defense-programs-of-the-rma-era/

Roques, P. (2018). *Systems architecture modeling with the Arcadia method: A practical guide to Capella*. ISTE Press Ltd.

RTCA SC- 167 / EUROCAE WG- 12. (1992). *RTCA/DO-178B: Software Considerations in Airborne Systems and Equipment Certification*. RTCA, Inc.

SAE Aerospace. (2010). *Aerospace Recommended Practice ARP4754A, Guidelines for Development of Civil Aircraft and Systems*.

SAE Aerospace. (2011). *Contiguous Aircraft/System Development Process Example. Aerospace Information Report, AIR 6110*.

Sanya, I. O., & Shehab, E. M. (2015). A framework for developing engineering design ontologies within the aerospace industry. *International Journal of Production Research*, *53*(8), 2383–2409. https://doi.org/10.1080/00207543.2014.965352

SkyBrary. (2019). *Deceleration on the Runway*. https://www.skybrary.aero/index.php/Deceleration_on_the_Runway

Staab, S., Oberle, D., & Guarino, N. (2009). *Handbook on Ontologies: What Is an Ontology?* (2nd edition). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-92673-3

Sun, X., Zeng, Y., & Zhou, F. (2011). Environment-Based Design (EBD) Approach to Developing Quality Management Systems: A Case Study. *Journal of Integrated Design and Process Science*, 19.

Swaminathan, S., & Smidts, C. (1999). The Event Sequence Diagram framework for dynamic Probabilistic Risk Assessment. *Reliability Engineering & System Safety*, *63*(1), 73–90. https://doi.org/10.1016/S0951-8320(98)00027-1

Tan, S., Zeng, Y., Chen, B., Bani Milhim, H. K., & Schiffauerova, A. (2012). Environment Based Design Approach to Integrating Enterprise Applications. *Journal of Computing and Information Science in Engineering*, *12*(3), 031003. https://doi.org/10.1115/1.4007171

Tan, S., Zeng, Y., Huet, G., & Fortin, C. (2013). Effective Reverse Engineering of Qualitative Design Knowledge: A Case Study of Aerospace Pylon Design. *Volume 4: 18th Design for Manufacturing and the Life Cycle Conference; 2013 ASME/IEEE International Conference on Mechatronic and Embedded Systems and Applications*, V004T05A026. https://doi.org/10.1115/DETC2013-13006

Technical Operations - INCOSE. (2007). *Systems Engineering Vision 2020*. International Council on Systems Engineering (INCOSE).

U.S. Code of Federal Regulations. (2020). *14CFR § 25.125—Landing*. GOVREGS. https://www.govregs.com/regulations/title14_chapterI_part25_subpartB_subjgrp75_section25.125

Voirin, J.-L. (2018). *Model-based system and architecture engineering with the arcadia method*. ISTE Press ; Elsevier.

Walden, D. D., Roedler, G. J., Forsberg, K., Hamelin, R. D., Shortell, T. M., & International Council on Systems Engineering (Eds.). (2015). *Systems engineering handbook: A guide for system life cycle processes and activities* (4th edition). Wiley.

Wang, H., & Wang, S. (2016). Application of ontology modularization to human-web interface design for knowledge sharing. *ELSEVIER*, 122–128.

Wang, M., & Zeng, Y. (2009). Asking the right questions to elicit product requirements. *International Journal of Computer Integrated Manufacturing*, *22*(4), 283–298.

Weissman, A., Petrov, M., & Gupta, S. K. (2011). A computational framework for authoring and searching product design specifications. *Advanced Engineering Informatics*, *25*(3), 516–534. https://doi.org/10.1016/j.aei.2011.02.001

Wu, Z. Y., Ming, X. G., He, L. N., Li, M., & Li, X. Z. (2014). Knowledge integration and sharing for complex product development. *International Journal of Production Research*, *52*(21), 6296–6313. https://doi.org/10.1080/00207543.2014.923121

Zeng, Y. (2008). Recursive object model (ROM)—Modelling of linguistic information in engineering design. *Computers in Industry*, *59*(6), 612–625. https://doi.org/10.1016/j.compind.2008.03.002

Zeng, Y. (2014). *EBD Book draft*. Concordia University.

Zeng, Y. (2015). Environment-Based Design (EBD): A Methodology for Transdisciplinary Design+. *Journal of Integrated Design and Process Science*, *19*(1), 5–24. https://doi.org/10.3233/jid-2015-0004

Zeng, Y. (2011). Environment-Based Design (EBD). *Volume 9: 23rd International Conference on Design Theory and Methodology; 16th Design for Manufacturing and the Life Cycle Conference*, 237–250. https://doi.org/10.1115/DETC2011-48263

Zeng, Y., & Cheng, G. D. (1991). On the logic of design. *Design Studies*, *12*(3), 137–141. https://doi.org/10.1016/0142-694X(91)90022-O

Zeroual, K., & Robillard, P.-N. (1992). KBMS: A Knowledge-Based System for Modeling Software System Specifications. *IEEE*, *4*, 15.